

# Incentive-Aware Machine Learning for Decision Making

A DISSERTATION PRESENTED

BY

CHARIKLEIA PODIMATA

TO

THE DEPARTMENT OF COMPUTER SCIENCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN THE SUBJECT OF

COMPUTER SCIENCE

HARVARD UNIVERSITY

CAMBRIDGE, MASSACHUSETTS

JUNE 2022

©2022 – CHARIKLEIA PODIMATA  
ALL RIGHTS RESERVED.

# Incentive-Aware Machine Learning for Decision Making

## ABSTRACT

Machine Learning algorithms are increasingly being deployed in consequential decision making for people's lives. These decisions affect widely different aspects of our lives; e.g., Machine Learning algorithms decide what content to serve us online (thus guiding our purchasing behavior and overall beliefs) or whether we are creditworthy enough for a loan. In light of how consequential these algorithmic decisions are, people have been documented to "strategize" with the data that they feed to the Machine Learning algorithms hoping to obtain better outcomes or decisions. This dissertation focuses on such decision-making settings where people are incentivized to react to the algorithmic decisions made.

Incentive-aware Machine Learning for decision-making creates a new ecosystem with three key stakeholders: the *institutions* deploying the algorithms, the *individuals* who are being impacted by the algorithms, and *society* as a whole. It is natural to expect that oftentimes there is tension between the goals of these stakeholders. In this dissertation we study the tension that arises from each of the stakeholders' perspectives. We start by thinking about the problem of algorithmic decision making from the institution's perspective. Drawing intuition from the literature on Algorithmic Game Theory, we build new machine learning algorithms that align the incentives of the individuals with those of the institution (incentive compatibility), while not sacrificing too much accuracy. Incentive-compatibility is the holy grail of incentive alignment and is sometimes unattainable. To alleviate this, we propose Machine Learning algorithms that learn to adapt to the incentives of the agents that they face, hence also ensuring a form of robustness. Shifting gears, we focus on the societal implications of information discrepancy regarding the deployed machine learning algorithms to different subpopulations. Finally, we study incentive-aware learning in settings where there are misspecifications regarding the behavioral model of the agents and incentive-aware learning in settings where the agents are non-myopic (and sometimes, they are even learners themselves).

This dissertation presents foundational theoretical advancements that range from algorithms for ubiquitous tasks such as linear regression and classification, prediction with expert advice, and Lipschitz bandits to theoretical results tailored to specific application domains such as credit scoring, forecasting, pricing, and auctions.

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>INTRODUCTION</b>                                       | <b>1</b>  |
| 1.1      | Application Domains . . . . .                             | 4         |
| 1.2      | Dissertation Roadmap . . . . .                            | 7         |
| 1.3      | Bibliographic Remarks . . . . .                           | 17        |
| 1.4      | Funding . . . . .   | 18        |
| <b>2</b> | <b>BACKGROUND</b>   | <b>19</b> |
| 2.1      | Offline Learning . . . . .                                | 19        |
| 2.2      | Online Learning . . . . .                                 | 20        |
| 2.3      | Multi-Armed Bandits . . . . .                             | 22        |
| 2.4      | Algorithmic Game Theory Background . . . . .              | 23        |
| 2.5      | Regret Notions against Strategic Agents . . . . .         | 23        |
| <b>I</b> | <b>Robustness to Strategic Individuals</b>                | <b>25</b> |
| <b>3</b> | <b>STRATEGYPROOF LINEAR REGRESSION IN HIGH DIMENSIONS</b> | <b>26</b> |
| 3.1      | Chapter Overview . . . . .                                | 26        |
| 3.2      | Model . . . . .   | 29        |
| 3.3      | Families of Strategyproof Mechanisms . . . . .            | 31        |
| 3.4      | Characterizing Strategyproof Mechanisms . . . . .         | 56        |

|           |  |            |
|-----------|--|------------|
| 3.5       | Efficiency of Strategyproof Mechanisms . . . . .                 | 61         |
| 3.6       | Discussion and Open Questions . . . . .                          | 63         |
| <b>4</b>  | <b>NO-REGRET AND INCENTIVE-COMPATIBLE ONLINE LEARNING</b>        | <b>65</b>  |
| 4.1       | Chapter Overview . . . . .                                       | 65         |
| 4.2       | Model and Preliminaries . . . . .                                | 67         |
| 4.3       | The Full Information Setting . . . . .                           | 71         |
| 4.4       | The Partial Information Setting . . . . .                        | 74         |
| 4.5       | Forward-Looking Experts . . . . .                                | 80         |
| 4.6       | Experiments . . . . .  | 81         |
| 4.7       | Discussion and Open Questions . . . . .                          | 84         |
| <b>II</b> | <b>ADAPTATION TO STRATEGIC INDIVIDUALS</b>                       | <b>85</b>  |
| <b>5</b>  | <b>LEARNING STRATEGY-AWARE LINEAR CLASSIFIERS</b>                | <b>86</b>  |
| 5.1       | Chapter Overview . . . . .                                       | 86         |
| 5.2       | Model and Preliminaries . . . . .                                | 90         |
| 5.3       | Stackelberg versus External Regret . . . . .                     | 91         |
| 5.4       | The GRINDER Algorithm . . . . .                                  | 95         |
| 5.5       | Simulations . . . . .  | 107        |
| 5.6       | Lower Bound . . . . .  | 109        |
| 5.7       | Discussion and Open Questions . . . . .                          | 112        |
| <b>6</b>  | <b>ADAPTIVE DISCRETIZATION FOR ADVERSARIAL LIPSCHITZ BANDITS</b> | <b>113</b> |
| 6.1       | Chapter Overview . . . . .                                       | 115        |
| 6.2       | Our Algorithm: Adversarial Zooming . . . . .                     | 123        |
| 6.3       | Algorithm's Guarantees . . . . .                                 | 127        |
| 6.4       | Regret Analysis Outline . . . . .                                | 130        |
| 6.5       | Regret Analysis (Details) . . . . .                              | 134        |
| 6.6       | AdvZoomDim under Stochastic Rewards . . . . .                    | 163        |

|     |                               |     |
|-----|-------------------------------|-----|
| 6.7 | AdvZoomDim Examples . . . . . | 166 |
|-----|-------------------------------|-----|

### III Behavioral Model Misspecifications in Incentive-Aware ML 175

#### 7 BEHAVIORAL MODEL MISSPECIFICATIONS AS ADVERSARIAL CORRUPTIONS: CONTEXTUAL SEARCH AND EXTENSIONS TO PRICING 176

|     |  |     |
|-----|--|-----|
| 7.1 | Chapter Overview . . . . .   | 176 |
| 7.2 | Model . . . . .  | 183 |
| 7.3 | Corrupted Projected Volume: Algorithm and Main Guarantee . . . . . | 186 |
| 7.4 | Analysis . . . . .   | 196 |
| 7.5 | Extension to Bounded Rationality . . . . .                         | 211 |
| 7.6 | Gradient Descent Algorithm . . . . .                               | 212 |
| 7.7 | Discussion and Open Questions . . . . .                            | 214 |

#### 8 NEARLY TIGHT BOUNDS FOR CORRUPTION-ROBUST CONTEXTUAL SEARCH 215

|     |   |     |
|-----|---|-----|
| 8.1 | Chapter Overview . . . . .  | 215 |
| 8.2 | Model and Preliminaries . . . . .   | 218 |
| 8.3 | A $O(C_0 + d \log(1/\varepsilon))$ Algorithm for the $\varepsilon$ -Ball Loss . . . . . | 220 |
| 8.4 | An Efficient $O(C_1 + d \log T)$ Algorithm for the Absolute Loss . . . . .              | 226 |
| 8.5 | Discussion and Open Questions . . . . .   | 232 |

### IV Fairness Considerations in Incentive-Aware ML 233

#### 9 INFORMATION DISCREPANCY IN INCENTIVE-AWARE LEARNING 234

|     |   |     |
|-----|---|-----|
| 9.1 | Chapter Overview . . . . .              | 234 |
| 9.2 | Model and Preliminaries . . . . .       | 237 |
| 9.3 | Equilibrium Computation . . . . .       | 240 |
| 9.4 | Equilibrium Analysis . . . . .          | 244 |
| 9.5 | Experiments . . . . .                   | 254 |
| 9.6 | Discussion and Open Questions . . . . . | 257 |

|  |            |
|--|------------|
| <b>V Incentive-Aware Machine Learning Beyond Myopic Agents</b>             | <b>258</b> |
| <b>10 THE USER PERSPECTIVE: LEARNING TO BID WITHOUT KNOWING YOUR VALUE</b> | <b>259</b> |
| 10.1 Chapter Overview . . . . .  | 259        |
| 10.2 Model and Preliminaries . . . . .                                     | 266        |
| 10.3 Abstraction: Learning with Win-Only Feedback . . . . .                | 267        |
| 10.4 Beyond Binary Outcomes: Outcome-Based Feedback . . . . .              | 269        |
| 10.5 Continuous Actions with Piecewise-Lipschitz Rewards . . . . .         | 276        |
| 10.6 Further Extensions . . . . .  | 283        |
| 10.7 Experimental Results . . . . .  | 290        |
| 10.8 Discussion and Open Questions . . . . .                               | 293        |
| <b>11 THE PLATFORM PERSPECTIVE: BANDITS WITH LONG-TERM EFFECTS</b>         | <b>296</b> |
| 11.1 Chapter Overview . . . . .  | 296        |
| 11.2 Model . . . . .   | 299        |
| 11.3 Relaxation: Dynamic Programming with Approximate Rewards . . . . .    | 301        |
| 11.4 Algorithm for General Speed Parameters . . . . .                      | 303        |
| 11.5 “Sticky” Arms . . . . .   | 311        |
| 11.6 Discussion and Open Questions . . . . .                               | 316        |
| <b>VI Conclusion and Open Questions</b>                                    | <b>318</b> |
| <b>12 CLOSING THOUGHTS</b>   | <b>319</b> |
| <b>APPENDIX A APPENDIX FOR CHAPTER 4</b>                                   | <b>324</b> |
| A.1 Gradient Descent Violates Incentive Compatibility . . . . .            | 324        |
| A.2 Supplementary Material for Sections 4.3– 4.4 . . . . .                 | 325        |
| A.3 Supplementary Material Forward-Looking Experts (Section 4.5) . . . . . | 327        |
| A.4 Additional Experiments (Supplementary for Section 4.6). . . . .        | 330        |
| <b>APPENDIX B APPENDIX FOR CHAPTER 5</b>                                   | <b>332</b> |

|   |   |            |
|---|---|------------|
| B.1                                       | Appendix for Section 5.3 . . . . .                                    | 332        |
| B.2                                       | Appendix for Section 5.4 . . . . .                                    | 337        |
| B.3                                       | Appendix for Section 5.5 . . . . .                                    | 340        |
| <b>APPENDIX C APPENDIX FOR CHAPTER 6</b>  |   | <b>344</b> |
| C.1                                       | Probability Tools . . . . .   | 344        |
| C.2                                       | Extension to Arbitrary Metric Spaces . . . . .                        | 345        |
| <b>APPENDIX D APPENDIX FOR CHAPTER 7</b>  |   | <b>348</b> |
| D.1                                       | Uncorrupted Contextual Search for $\varepsilon$ -ball loss . . . . .  | 348        |
| D.2                                       | Extension to Unknown Corruption (Proof of Theorem 7.1) . . . . .      | 350        |
| D.3                                       | Auxiliary Lemmas . . . . .  | 353        |
| <b>APPENDIX E APPENDIX FOR CHAPTER 9</b>  |   | <b>362</b> |
| E.1                                       | The Principal’s Learning Problem . . . . .                            | 362        |
| E.2                                       | Supplementary Material for Section 9.3 . . . . .                      | 366        |
| E.3                                       | Supplementary Material for Section 9.5 . . . . .                      | 367        |
| E.4                                       | Generalizing to Multiple Subgroups . . . . .                          | 368        |
| <b>APPENDIX F APPENDIX FOR CHAPTER 10</b> |   | <b>372</b> |
| F.1                                       | Notes on Chapter 10.4.1 . . . . .                                     | 372        |
| F.2                                       | Standard Proof for the Regret of Exponential Weights Update . . . . . | 373        |
| <b>APPENDIX G APPENDIX FOR CHAPTER 11</b> |   | <b>375</b> |
| G.1                                       | Generalization for Unknown Replenishing Arm . . . . .                 | 375        |
| <b>REFERENCES</b>                         |   | <b>401</b> |

# Listing of figures

|     |   |     |
|-----|---|-----|
| 1.1 | The Incentive-Aware ML Ecosystem . . . . .  | 4   |
| 1.2 | Dissertation Roadmap . . . . .  | 8   |
| 3.1 | Classification of agents into $N_0, \dots, N_4$ used in the Proof of Theorem 1.1. . . . .   | 34  |
| 3.2 | Counterexamples of the strategyproofness of $(S, S')$ -CRM. . . . .                         | 42  |
| 3.3 | Verification of various claims through Mathematica . . . . .                                | 62  |
| 4.1 | Experiments on the 2018–2019 FiveThirtyEight NFL dataset. . . . .                           | 82  |
| 5.1 | Sketch of strong incompatibility example . . . . .  | 92  |
| 5.2 | Agent’s action space in 2d. . . . .   | 96  |
| 5.3 | Polytope partitioning in 2d . . . . .   | 97  |
| 5.4 | Performance of GRINDER vs. EXP3 . . . . .   | 107 |
| 5.5 | Performance of GRINDER vs. BGD. . . . .   | 109 |
| 7.1 | Sketch of Carathéodory’s theorem and the computation of a separating cut. . . . .           | 191 |
| 7.2 | Sketch on why proper cuts do not suffice. . . . .   | 207 |
| 7.3 | Undesirability of proper cuts in 2 dimensions. . . . .                                      | 210 |
| 9.1 | Evaluation on the TAIWAN-CREDIT and ADULT dataset for $\mathbf{w}^*$ being the ERM. . . . . | 256 |

|      |  |     |
|------|--|-----|
| 10.1 | Example interfaces of bid simulators of two major search engines, Google Adwords (left) and BingAds (right), that enables learning the allocation and the payment function. (sources Standard (2014), Land (2014)) . . . . . | 263 |
| 10.2 | Regret of WIN-EXP vs EXP3 for different discretizations $\varepsilon$ ( $\text{CTR} \sim U[0.5, 1]$ ). . . . .   | 292 |
| 10.3 | Regret of WIN-EXP vs EXP3 for different CTR distributions and stochastic adversaries, $\varepsilon = 0.01$ . . . . .   | 292 |
| 10.4 | Regret of WIN-EXP vs EXP3 for different CTR distributions and adaptive EXP3 adversaries, $\varepsilon = 0.01$ . . . . .  | 293 |
| 10.5 | Regret of WIN-EXP vs EXP3 for different CTR distributions and adaptive WINEXP adversaries, $\varepsilon = 0.01$ . . . . .  | 293 |
| 10.6 | Regret of WIN-EXP vs EXP3 with noise $\sim \mathcal{N}(0, \frac{1}{m})$ for stochastic adversaries, $\varepsilon = 0.01$ . . . . .   | 294 |
| 10.7 | Regret of WIN-EXP vs EXP3 with noise $\sim \mathcal{N}(0, \frac{1}{m})$ for adaptive EXP3 adversaries, $\varepsilon = 0.01$ . . . . .  | 294 |
| 10.8 | Regret of WIN-EXP vs EXP3 with noise $\sim \mathcal{N}(0, \frac{1}{m})$ for adaptive WINEXP adversaries, $\varepsilon = 0.01$ . . . . .  | 294 |
| 11.1 | State evolution for different parameters for MAB with long-term effects. . . . .   | 300 |
| A.1  | Experiments on the 2019–2020 FiveThirtyEight NFL dataset. . . . .  | 329 |
| A.2  | Simulation results for $K = 50$ experts. . . . .   | 330 |
| B.1  | GRINDER vs. EXP3 for utility function from Eq. (B.3.1) . . . . .   | 342 |
| B.2  | GRINDER vs. EXP3 for utility function from Eq. (B.3.1) . . . . .   | 343 |
| D.1  | Single Dimensional Binary Search. . . . .  | 350 |
| E.1  | Evaluation on ADULT dataset when $w^*$ is drawn uniformly at random (left) and when $A_g$ 's are drawn uniformly at random (right). . . . .  | 367 |
| E.2  | Evaluation on TAIWAN-CREDIT dataset when $w^*$ is drawn uniformly at random (left) and when $A_g$ 's are drawn uniformly at random right). . . . .   | 369 |

TO MY GIRLS, LYDIA AND TERRA.

# Acknowledgments

This dissertation (and getting through gradschool and the academic job market) would not have been possible without the help and support from so many folks. I want to take the time to acknowledge them in this part. This is a long acknowledgments section, but I somehow felt that I have way more to say as I finished writing it.

First and foremost, I want to thank my advisor, Yiling Chen. Some 7 years ago, when I was submitting my applications for gradschool, I remember my undergrad advisor Dimitris Fotakis telling me that I should definitely apply to Harvard and try to be advised by Yiling. His argument was that she is exactly the advisor I need. What he meant became obvious to me from the very first meeting with Yiling. Yiling shaped me into the researcher I am today. The way that I approach questions, the way that I build models of the world, even the way that I present technical arguments are due to her. Through our discussions, she taught me how to find and calibrate research questions. Yiling was always there any time I asked for her help or input; be it about a career opportunity or just about micro-strategizing during the PhD. Lastly, Yiling has offered to me the epitome of a healthy advisor-advisee relationship as she always made it clear that she was there to help me build and advance my own research agenda. I feel truly indebted to have been her advisee.

I would then like to thank my dissertation committee: David Parkes, Ariel D. Procaccia, Vasilis Syrgkanis, and Jennifer Wortman Vaughan. Throughout the years, David has been hugely helpful intellectually in the way that I talk about my research (both in talks and this very dissertation) and he is responsible for fostering an incredibly welcoming and fun community at Harvard EconCS. My collaboration with Ariel was incredible: not only is he extremely bright research-wise, but he

was also extremely caring in making sure to foster an inclusive environment where I felt supported, heard, and welcome. Vasilis —apart from being incredibly supportive whenever I needed him— was the one who introduced me to multi-armed bandits and taught me the basics. As for Jenn, and I have said this story many times, I have never felt a deeper connection research-wise as I felt with her during our first research meeting while I was interviewing for the MSR PhD fellowship. She has been a fierce advocate for me for years now and I wish we can soon craft some time in our schedules to work on something together again.

I would next like to take the time to thank all the folks at the MSR NYC office that I have spent so much time interacting with and writing papers, particularly Solon Barocas, Rupert Freeman (now at UVA Darden), Hal Daumé III, Miro Dudík, Dan Goldstein, Jake Hofman, Akshay Krishnamurthy, Dipendra Misra, Dave Pennock (now at DIMACS), Sid Sen, Alex Slivkins, and Hanna Wallach. All these folks came into my life after that first, fateful meeting with Jenn during my interview for the PhD fellowship. Jenn and Dave Pennock accepted me as their intern for my first internship and this is how I met everyone. I always marveled at Dave’s shrewdness and calm disposition; outside of a great collaborator and mentor, he was a huge help during my academic job market. The fourth member of the wonderful team that we had with Jenn and Dave was Rupert Freeman; I am incredibly thankful for his friendship which has way trascended my internship time. I subsequently returned for my second internship, this time with the inimitable Alex Slivkins, but this was unfortunately cut short due to the early restrictions of COVID in March 2020. Apart from a wonderful internship mentor, Alex always found time for me and helped tremendously in crafting my job talk slides. During my internships, I also had the opportunity to work with Akshay Krishnamurthy, Rob Schapire, and Thodoris Lykouris. It has been such an invaluable experience working with them: Akshay and Thodoris have become close friends of mine and Rob has taught me a lot about the traditional ML theory lens to incentive-aware problems. All the MSR folks (be it collaborators or not) have created an amazing environment in this office, where you can not only connect with people on a work level but also on a more personal one; and that is what clicked for me. They all hold a special place in my heart and I cannot wait to visit the (new, for me!) office again to see everyone.

During my PhD, I was very privileged to also work with the fantastic folks in the Market Algo-

rithms group in Google Research NYC. Although the fact that each of us was locked in our houses due to COVID made the experience very different from an in-person internship, I got a glimpse of how incredible and close-knit this group is. I want to single-out Renato Paes Leme, who was my mentor during that internship. Working with him has been a blast and he was immensely helpful during my academic job search. I would also like thank Jon Schneider and Khashayar Khosravi for spending many fun hours brainstorming, discussing ideas, and writing papers with me as well as Vahab Mirrokni, Balu Sivan, and Yuan Deng who regularly met with me and chatted research. All of them went above and beyond in making sure that I was working on fun projects that fit my research agenda while not overwhelming me.

Next, I want to thank my many other collaborators not highlighted above: Yahav Bechavod, Zhe Feng, Dimitris Fotakis, Kyriakos Lotidis, Nisarg Shah, Steven Zhiwei Wu, and Juba Ziani. With all of them, we have spent countless hours staring at half-baked proof ideas on whiteboards or weird experimental results. This dissertation would not have been the same without each and every one of them. Among these people, I would like to especially thank Dimitris Fotakis who was also my undergrad advisor as I mentioned earlier. A large part of my academic journey is due to him: he recognized my potential when I was a junior undergrad and helped me achieve my dream of pursuing a PhD in the US. I am also very thankful to my “job market buddies” Katerina Sotiraki, Manolis Zampetakis, Chamsi Hssaine, and Surbhi Goel, although we have not (yet) become collaborators. I am very proud of all of them and hope to co-author some works soon.

I big thank you goes out to the Harvard EconCS community which has always been a very fun and supportive environment. Special shout-out to Dimitris, Hongyao, Eric, Thibault, Jean, Debmalya, and Lily for spending countless hours laughing and nearly-crying about the state of our PhDs or reviewer #2 in some conference. Outside of the EconCS group, I am very thankful to the friends that I made from my cohort: Tanvi, Ilia, Michael, Austin, Yamini, Christina, Sharon, and Brian. Years from now, I will look back fondly to the memories of us eating insane amounts of hotpot together.

I have been extremely fortunate to have enjoyed the unwavering support of a superb group of people. I am thankful to Ioanna, Nikos, and Polina. Although we do not talk all the time (and now that Lydia is here and we have Terratouli, I do not visit Greece so often), every-time we meet is as

if I never left.

Outside of my own group of friends, I have been very lucky that Lydia's friends have embraced me as part of their group. I want to thank Isidoros, Guille, Myria, Miltos, Eirini, and Christina for never taking me very seriously, for deep, insightful discussions, for allowing me to sleep in greek feasts in Ikaria or in various tavernas, and for loving Lydia the way they do. I want to also thank Danai, Adam, and Turing (the dog!) for being extremely caring and supportive (and also not taking me very seriously). Some of the best memories of my life have been just chatting with them at night in the living room in Asklipiou. I am so thrilled to be doing the US cross-country roadtrip with them!

When I left Greece in 2016, I did not know that I would find a second family in Boston. Words cannot express how thankful I am for our close friends here; Lydia Z., Marinos, Manolis, Katherine, Shibani, Dimitris, Sophie, Konstantina M., Konstantina B., Konstantinos, Christina, Artemisa, Stella, Yorgos, Will, Ilias, Thodoris, and Ariadne. From building our wedding cake and organizing our wedding party to endless hours of cooking, baking, moving apartments in Boston, and dancing to Greek 90s songs, there are so many wonderful memories that it would be unfair for me to single out any one. I still dream that in a couple of years from now, we will all find a way to all live (very) close.

It is hard to nurture an extremely anxious child (that was me); my family has done this, however, and has stood by me in every step of the way. My mom, Roza, used to stay up at night with me when I was studying in Volos, so that I would not be alone. My dad, Thomas, loves reminding me (and everyone else) that he was the one who taught me my very first algorithms. My younger sister, Stella, has always been so different from me but at the same time has been one of my fiercest advocates. I am so incredibly proud of the person she has become.

I want to close this (long) acknowledgments section by talking about my wife, Lydia, and our dog, Terra. Lydia and I have been together for nearly a decade now. We have traveled around Greece with my old 2000 Honda CR-V, camped in various places, and now we get to explore the US. I do recognize my immense privilege when I say this, but I enjoyed the quarantine to some extent; Lydia and I had so much time together! It was a dream! We played cards and Nintendo and cooked and did Zoom birthdays with just the two of us and all our friends online. She has

always been a beacon of light and hope in my (constantly busy, anxious, depressed) life and she does so without having any clue what I actually do for a living! And I know that I am biased when I say these things, but it is not a coincidence that anyone who meets her just comments on how positive, funny, calm, and cool she is as a person. Late in 2020, after the first wave of the pandemic, she begrudgingly agreed to us bringing home this black, 10 week old puppy with the big paws that only weighed about 10lbs at the time. And the rest is history. We now spend our days adventuring with Terra, taking her for swimming in the New England lakes, discussing how beautiful and loving she is, and daily falling in love with her. So this dissertation is dedicated to them, my two girls, Lydia and Terra for making me a better person and filling my every day with love and joy. I cannot wait to make more memories with you!

# 1

## Introduction

From applications enhancing our everyday lives and online websites guiding our purchasing behaviors to a slew of consequential decisions made for hiring, loan approvals, college admissions, probation decisions and many more, Machine Learning (ML) algorithms are omnipresent. Because of how important these decisions are for our lives, people are oftentimes incentivized to alter the data that they submit to these algorithms, in an effort to obtain better outcomes for themselves. For example, individuals try to increase the number of credit cards or bank accounts that they have and to improve their overall credit standing before petitioning for a loan. In school admissions, student candidates retake the GRE, they pay for tutoring classes, and sometimes even change schools (to improve their in-class ranking) in an effort to be deemed more qualified for the admissions process. Similar examples of “strategic adaptation” from people have been docu-

mented in a lot of works from a variety of domains that span Economics, Computer Science, and Public Policy (Björkegren et al., 2020, Dee et al., 2019, Dranove et al., 2003, Greenstone et al., 2020, Gonzalez-Lira and Mobarak, 2019). The problem here arises because when the decision maker deploying the ML algorithms does not account for this adaptation, they risk making policy decisions that are incompatible with the original policy’s goal. For example, in the school admissions case, changing schools just to obtain a better in-class ranking does not make the student inherently more qualified to be admitted. Note here that the “strategic adaptation” does not need to have a purely negative sign; some adaptations constitute efforts from the individuals to *game* our algorithms (e.g., changing schools to improve in-class ranking) but some others include efforts for *genuine improvement* (e.g., studying more).

To understand where this incompatibility between the “*a priori target policy*” and the “*a posteriori policy*” arises, we have to look deeper into what motivated us to use ML to make such policy decisions in the first place; we wanted to use powerful ML tools to learn from past human data and thus make better future decisions. The reason why these tools were deemed appropriate lies at the heart of the ML paradigm, which roughly says that patterns in the past/training data should translate to patterns in test/future data. Mathematically, this means the following. Assume that there is a data generating process through which we sample datapoints  $(x, y)$  from a distribution  $\mathcal{D}$ . In our loan approvals example, each datapoint would correspond to an individual applying for a loan with  $x \in \mathbb{R}^d$  corresponding to their *features* (such as income, education, gender, race, number of credit cards etc) and  $y \in \mathbb{R}$  being the individual’s creditworthiness. In standard supervised ML, we assume that the learner gets access to a *training* set of say  $n$  examples:  $\text{Train} = \{(x_i, y_i)\}_{i \in [n]}$ . Given this set, standard algorithms output a hypothesis  $h^*$  from the hypothesis set  $\mathcal{H}$  that minimizes the expected error, i.e.,

$$h^* = \arg \min_{h \in \mathcal{H}} \sum_{i \in [n]} (h(x_i) - y_i)^2 = \arg \min_{h \in \mathcal{H}} L(h, \text{Train})$$

for some convex loss function  $L$ . Then, if  $h^*$  is applied as a hypothesis on a test dataset  $\text{Test} = \{(x'_i, y'_i)\}_{i \in [m]}$  with fresh examples sampled from  $\mathcal{D}$ , then  $L(h^*, \text{Test})$  is also small. Translating to the loan approvals example, this would mean that given past application data alongside the credit

scores of the applicants, a bank can identify the hypothesis  $h^*$  that best explains the correlation between high credit scores and applications and deploy this  $h^*$  to judge future loan applications. But when people start becoming aware of the intricacies of the algorithms (e.g., the fact that they place a lot of value in applicants with many credit cards) they become incentivized to change their data, as we have argued above. This means that the decisions made by the learner at training time will affect the datapoints that we receive at test time. Mathematically, this would mean that this original distribution  $\mathcal{D}$  is different from the distribution  $\mathcal{D}(h^*)$ , which is induced after people have become aware of how the algorithms work. This phenomenon has been also observed for different contexts in traditional ML under the name “distribution shift”, e.g., ([Schlimmer and Granger, 1986](#), [Widmer and Kubat, 1993](#), [Kelly et al., 1999](#), [Shimodaira, 2000](#)).

*So what should the learner actually do in these cases, where individuals are incentivized to alter the data that they feed into the ML algorithms in hopes of obtaining better outcomes? And even if the learner finds a way to robustify or just calibrate their algorithms to the behavior of the individuals faced, what are the societal implications?* These are some of the core questions that we address in this dissertation. Before we delve into the ways with which we tackle these questions, we find it useful to present the three key stakeholders that participate in any incentive-aware ML system (see also Figure 1.1). As we will see in the following chapters, these three key stakeholders have oftentimes conflicting objectives and the tension that arises from this conflict is what drives the majority of the research in this area. To understand these tensions, we define each stakeholder’s goals and their leverage.

**Stakeholder 1: Institution.** The first stakeholder is the *institution* (aka the *decision-maker*), e.g., the bank, company, school that deploys the ML algorithm to make decisions. Their goal is to use ML to predict accurately the future in an effort to maximize revenue or some other goal (e.g., maximize the probability that students will be successful if admitted or maximize fairness in the context of the justice system). The institution’s leverage is that they can change the algorithms that they use however they deem fit in an effort to make them more robust.

**Stakeholder 2: Individual.** The second stakeholder is the *individual*. These are the people who supply their data to the institutions. The individual’s goal is to obtain the best outcomes for themselves, e.g., to be admitted to the school of their choice or to be approved for the loan they are

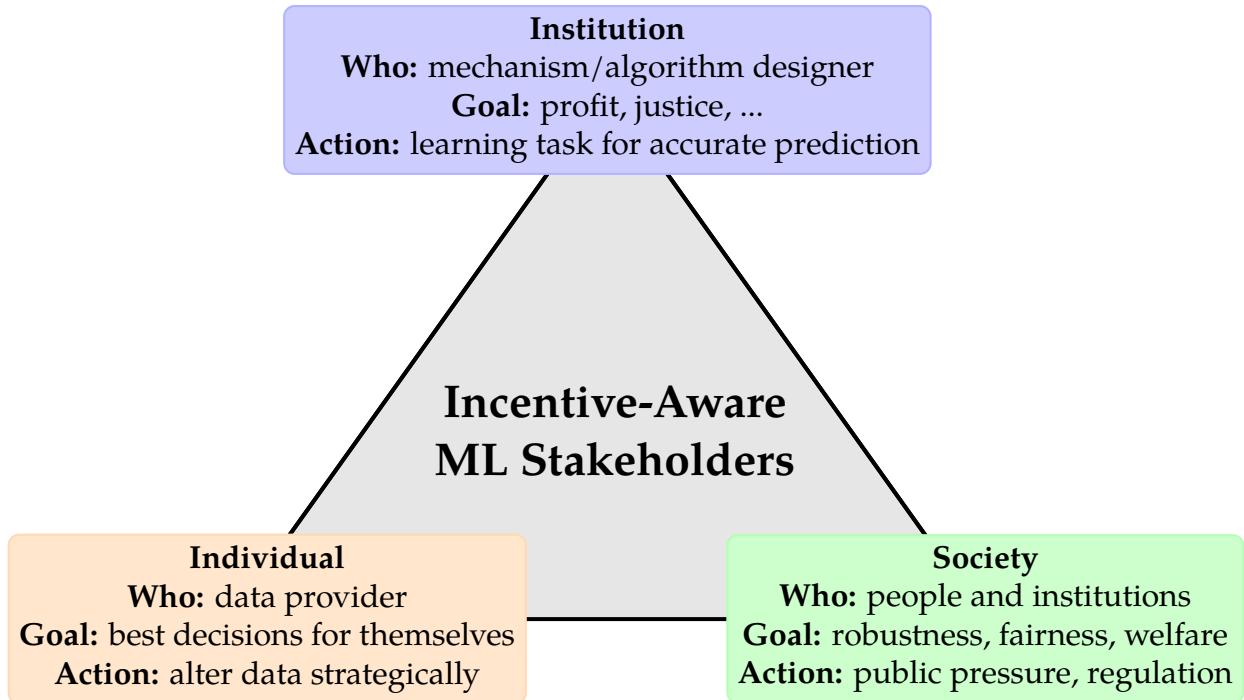


Figure 1.1: The Incentive-Aware ML Ecosystem

applying for. The individual's leverage is that they can strategically alter their data to obtain their desired outcomes. Their strategizing is governed by a specific utility function, which is usually unknown (or only partially known) by the learner.

**Stakeholder 3: Society.** The third and final stakeholder is *society* comprised by all people and institutions as a whole. Society's goals are very broad; we strive for fairness, robustness to bad actors, and improvement of our social welfare. Society's leverage is to regulate, to create public pressure to institutions, and to decide on the norms and expectations from them towards individuals.

## 1.1 APPLICATION DOMAINS

The results of this dissertation present foundational advancements in the area of incentive-aware ML but are heavily motivated from several application domains where the deployment of ML for decision making is prevalent. In this section, we highlight these domains.

## FORECASTING

Forecasting is the practice of soliciting opinions from a set of individuals in order to help inform the prediction that a decision-maker issues for a future event. In the past couple of years, the science behind it has seen extensive scrutiny especially due to its applicability in election polling. However, what all the standard algorithms still miss is the fact that some of the individuals giving us their opinions may be strategic and they may try to influence the prediction outcome in one way or the other so that they are more favorably evaluated by the algorithm. Such strategizing examples have been observed in various real-life scenarios, like The Good Judgment Project<sup>\*</sup> and the website FiveThirtyEight<sup>†</sup>. Motivated by these considerations, in Chapter 4 we design forecasting algorithms that offer simultaneously optimal error guarantees and make sure that no individual has an incentive to misreport.

## SPAM CLASSIFICATION

In spam email classification (see (Dada et al., 2019) for an overview), the agents are spammers sending spam emails that they try to disguise as non-spam so that they fool the classifier used by the learner. Note that the spammers have two characteristics: first, they are not *fully adversarial* in that they do not try to sabotage the classification algorithm altogether but merely “fool” it. Second, they cannot arbitrarily change their original spam emails so as to “pass” the classifier; for example, a very simple way for a spam email to be turned non-spam is to erase any potential link that it includes. But no spammer would actually make this change as it brings them no profit to send non-spam emails. In other words, the spammers are constrained in their effort to fool the learner’s classifier. Motivated by these two characteristics (which can both be addressed through incentive-aware ML), spam classification is the workhorse application of Chapter 5.

## DYNAMIC PRICING

Dynamic pricing (see Chapter 6) is the paradigm of repeatedly posting a specific price for an item and allowing the users to decide to take-it-or-leave-it. This paradigm has been extensively applied

---

<sup>\*</sup><https://goodjudgment.com/>

<sup>†</sup><https://fivethirtyeight.com/>

in a variety of online marketplaces. The contextual version of the problem, where the item is described by a feature vector affecting its price, is a generalization of the original formulation. This generalization can be viewed as an abstraction of the key paradigm behind the SmartPricing<sup>†</sup> algorithm (which AirBnB uses to suggest listing prices to superhosts) and personalized medicine ([Bastani and Bayati, 2020](#)). We focus on the contextual generalization of pricing in Chapters 7 and 8.

## CREDIT SCORING

Credit scoring algorithms are nowadays used extensively in algorithmic decision making, e.g., for making loan decisions. At a high level (see also ([Thomas et al., 2017](#)) for more details), a credit scoring algorithm learns a mapping from a vector of features of an individual to a score. The features correspond to characteristics of the individual (e.g., annual income, number of credit cards, demographic characteristics etc.) and the score is an integer number that encodes how these features are correlated with the individual’s creditworthiness. Motivated by the omnipresence of credit scoring algorithms in algorithmic decision making, in Chapter 9 we study how the information discrepancies in different subpopulations of people are inherently tied with their ability to take actions to improve their credit standing with respect to these algorithms.

## AUCTIONS

Online ad auctions generate billions of dollars in yearly revenue ([pwc report, 2020](#)). In ad auctions, there are three key “players”; the decision-maker (i.e., company serving the ads), the advertisers, and the users. The decision-maker collects the ads and the bids from the advertisers and runs an internal auction to decide how to allocate the ads. The decision-maker also decides on the payment that each advertiser should give for having their ads shown. The bidders/advertisers place their bids for each slot and their ads to the platform. Finally, the users click on the ads that they like (i.e., ads that offer them some value). In the most standard form of online ad auctions, once an ad is clicked, the advertiser pays the fee (decided through the auction) to the platform. Part V studies the different perspectives of the decision-maker and the advertisers in online ad auctions.

---

<sup>†</sup><https://www.airbnb.com/help/article/1168/smart-pricing>

## 1.2 DISSERTATION ROADMAP

This dissertation is comprised by five parts (see also Figure 1.2 for an easy pictorial representation). Each part corresponds to a bigger direction in incentive-aware machine learning and is comprised by chapters which discuss more specific results. In Part I we discuss ways in which an institution can incentivize truthtelling from the agents while not sacrificing too much from the ML algorithm’s accuracy. In Part II we discuss ML algorithms that —although not able to guarantee truthtelling— can guarantee that they appropriately adapt to the strategic agents faced. In Part III we challenge the standard assumption that the institution knows always exactly the utility functions that govern the agents’ behavior. In Part IV we challenge the assumption that the institution always faces a homogeneous population of individuals and discuss fairness in this context. In Part V we go beyond myopic agents in incentive-aware ML settings. We conclude this dissertation by discussing the most important avenues for future work in incentive-aware ML for decision making in Chapter 12. Although there are a lot of common related works across chapters, we have chosen for each chapter to have its own related work section; this serves our purpose of comparing directly with the specific results discussed.

The majority of this dissertation focuses on the institution’s perspective (Chapters 3, 4, 5, 6, 7, 8, 11) for a wide variety of different settings (to name a few, strategic classification/regression, prediction with expert advice, pricing, auctions). We focus on the individual’s perspective in learning in auctions in Chapter 10. The society’s perspective is adopted in Chapter 9. Below we outline in more detail the content of these chapters in Sections 1.2.1—1.2.5.

Before delving into the details for our results, Chapter 2 has some useful background on technical notions that we will keep encountering throughout this dissertation.

### 1.2.1 PART I: ROBUSTNESS TO STRATEGIC INDIVIDUALS

From the perspective of the decision maker, the strategic alteration of the data can be viewed as a form of “strategic” noise, i.e., noise that is added as a result of the underlying utilities of the agents/individuals. Compared to adversarial noise, this model has an idiosyncratic advantage (when its underlying assumptions hold true); if the learner aligned the agents’ incentives cor-

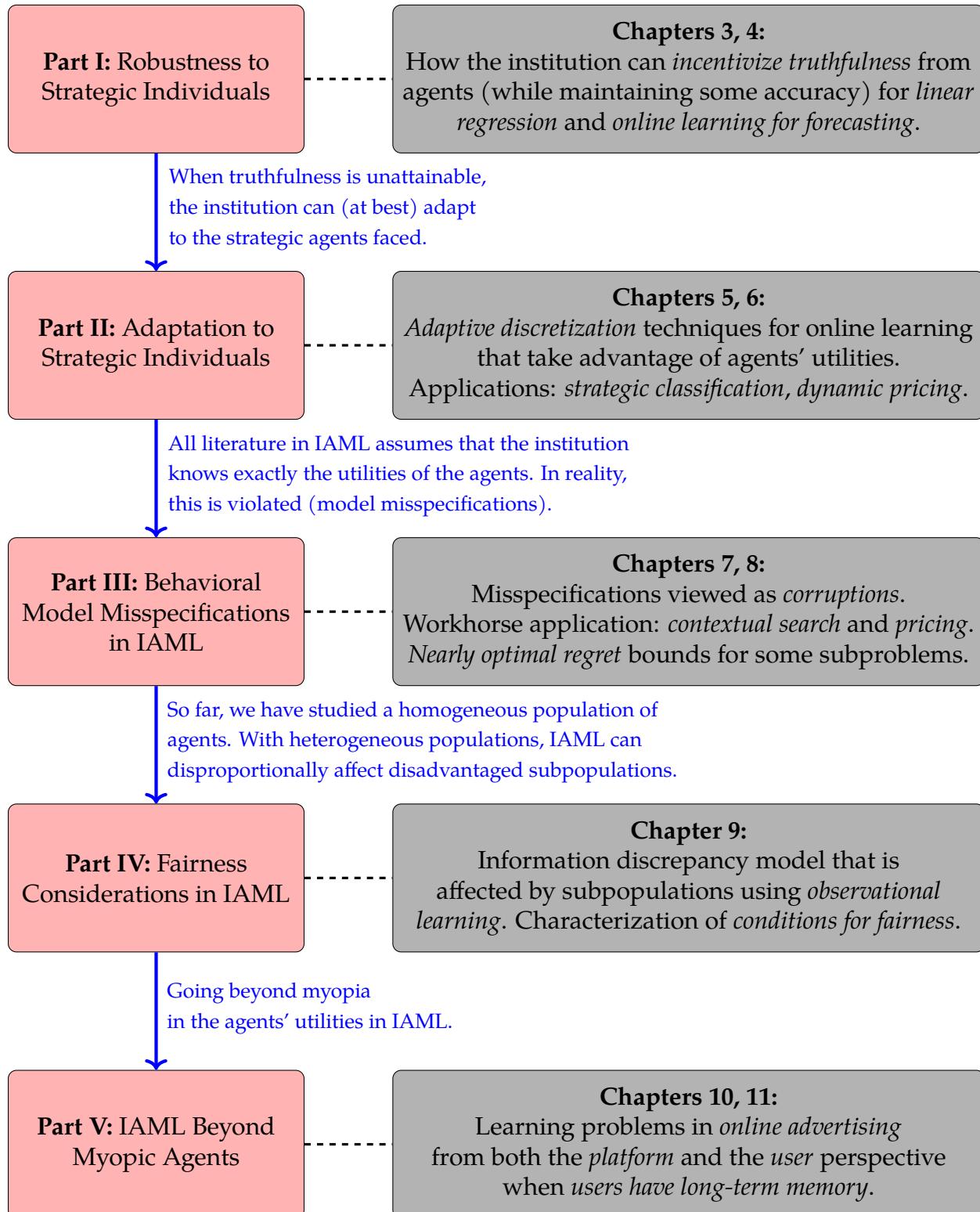


Figure 1.2: Dissertation roadmap. For space reasons, we have used the acronym "IAML" in place for "Incentive-Aware Machine Learning".

rectly, it would be possible to obtain fully uncontaminated data. From this viewpoint, the ideal is the design of learning algorithms that in addition to being statistically efficient, are *strategyproof*, i.e., where supplying pristine data is a dominant strategy for each agent. In Part I we aim to build strategyproof learning algorithms. We start our study of incentive-aware machine learning by presenting algorithms for ubiquitous tasks both in offline and online learning that satisfy the strongest incentive guarantee; *incentive compatibility* (aka *strategyproofness*). Roughly speaking, an algorithm is said to be incentive compatible, if revealing truthfully their data is the best strategy for all agents interacting with it. Note that apart from helping us obtain clean datasets, incentive-compatibility may be highly desirable from the individuals' perspective; indeed, when a mechanism is incentive compatible, it places no cognitive burden on individuals to figure out their optimal behavior, as their optimal behavior is simply truthtelling. In this dissertation, we focus on incentive-compatible/strategyproof algorithms that do not require the use of monetary payments from the decision-maker to the agents/individuals.

In Chapter 3, we study high dimensional linear regression, when the data is given by strategic agents who wish to bring the regression lines as closely as possible to their true datapoint. Such incentives arise when the agents wish to not be perceived as outliers, like for example the case of a company that does not wish to be penalized for a randomly very good (e.g., during Christmas) or very bad (e.g., during lockdown) performance. More concretely, Dekel et al. (2010) give the real-world example of the global fashion chain Zara, whose distribution process relies on regression Caro and Gallien (2010). Specifically, the demand for each product at each store is predicted based on historical data, as well as information provided by store managers. Since the supply of popular items is limited, store managers may strategically manipulate requested quantities so that the output of the regression process would better fit their needs, and, indeed, there is ample evidence that many of them have done so Caro et al. (2010). More generally, as discussed in detail by Perote and Perote-Peña (2004), this type of setting is relevant whenever “data could come from surveys composed by agents interested in not being perceived as real outliers if the estimation results could be used in the future to change the economic situation of the agents that generate the sample”. Our main contribution is the discovery of a family of algorithms for linear regression in high dimensions which are *group-strategyproof*, i.e., if individuals were to create coalitions to

strategize together, there does not exist a group formation such that all agents are at least not worse off and one is strictly better off. We call these algorithms “*generalized resistant hyperplane*” mechanisms. The game-theoretic properties of these mechanisms — and, in fact, their very existence — are established through a connection to a discrete version of the Ham Sandwich Theorem, which is a well-known tool in the field of Computational Geometry.

Moving from offline to online learning, while still requiring incentive compatibility may look tricky at first. Indeed, in a repeated setting agents may have many more opportunities for strategizing compared to the static, offline setting. In Chapter 4, however, we provide positive results on incentive-compatible online learning. Specifically, we provide incentive-compatible ML algorithms for the prediction with expert advice task, where a learner makes predictions about a sequence of  $T$  events. The learner has access to a pool of  $K$  experts, each with beliefs about the likelihood of each event occurring. The standard goal (Vovk, 1990, Littlestone and Warmuth, 1994, Cesa-Bianchi et al., 1997, Freund and Schapire, 1997, Vovk, 1998, Auer et al., 2002b) of the learner is to output a sequence of predictions almost as accurate as those of the best fixed expert in hindsight. Such a learner is said to have no regret. But, as pointed out by Roughgarden and Schrijvers (2017), when the learner is not only making predictions but also (implicitly or explicitly) evaluating the experts, experts might have incentive to misreport. The Good Judgment Project,<sup>§</sup> a competitor in IARPA’s Aggregative Contingent Estimation geopolitical forecasting contest, scored individual forecasters and rewarded the top 2%—dubbed “Superforecasters” Tetlock and Gardner (2015)—with perks such as paid conference travel; some are now employed by a spinoff company. Similarly, the website FiveThirtyEight<sup>¶</sup> not only predicts election results by aggregating different pollsters, but also publicly scores the pollsters, in a way that correlates with the amount of influence that the pollsters have over the FiveThirtyEight aggregate. It is natural to expect that forecasters might respond to the competitive incentive structure in these settings by seeking to maximize the influence that they exert on the learner’s prediction.

Towards achieving simultaneously incentive compatibility and no regret, we show a novel connection between online learning and *wagering mechanisms* (Lambert et al., 2008, 2015), a type of

---

<sup>§</sup><https://goodjudgment.com>

<sup>¶</sup><https://fivethirtyeight.com/>

multi-agent scoring rule that allows a principal to elicit the beliefs of a group of agents without taking on financial risk. Using this connection, we construct online learning algorithms that are incentive-compatible and incur sublinear regret. For the full information setting (i.e., when the prediction of all of the agents is revealed at each round  $t$ ), we introduce Weighted-Score Update (WSU), which yields regret  $\mathcal{O}(\sqrt{T \ln K})$ , matching the optimal regret achievable for general loss functions, even without incentive guarantees. For the partial information setting (i.e., when only the prediction of one chosen agent is revealed at each round  $t$ ), we introduce Weighted-Score Update with Uniform Exploration (WSU-UX), which achieves regret  $\mathcal{O}(T^{2/3}(K \ln K)^{1/3})$ . Although worse than the worst-case optimal regret guarantee achievable when incentives are not an issue, it is unclear where the  $\mathcal{O}(T^{2/3}(K \ln K)^{1/3})$  is optimal when the incentives of the agents are accounted for. We also offer a partial extension to the case of forward-looking experts (i.e., experts that care about maximizing their influence for future rounds). Finally, we complement our theoretical analysis with experiments on data gathered from FiveThirtyEight.

### 1.2.2 PART II: ADAPTATION TO STRATEGIC INDIVIDUALS

In Part I we provided strong incentive-compatibility results for the fundamental tasks of linear regression and prediction with expert advice. Despite these positive results showing that incentive compatibility and convergence to meaningful learned solutions is simultaneously achievable, incentive-compatibility is a very hard desideratum to satisfy. Not only that, but it also requires the decision-maker to know *exactly* the utilities of the agents they face, in an effort to align their incentives with the algorithm's.

In Part II we explore ways for the learning algorithms to “adapt” to the strategizing of the agents. Specifically, in Chapter 5 we study learning for *strategic classification*, which has recently received a lot of interest from the community at the interface of ML and Economics ([Hardt et al., 2016](#), [Dong et al., 2018](#), [Chen et al., 2020b](#), [Ahmadi et al., 2021](#)). Specifically, we are motivated from an email spamming classification example, where spammers try to strategically change their emails in an effort to fool and bypass the classification algorithm. Note that in this case, spammers do not wish to sabotage the classification algorithm only for the sake of harming its performance. Instead, they merely want to game it for their own benefit. And this is precisely what differentiates

them from being fully adversarial. We model the interplay between the learner and the strategic agents as a repeated *Stackelberg* game over  $T$  rounds. In a repeated Stackelberg game, the learner (aka “leader”) commits to an action, and then, the agent (aka “follower”) best responds to it, i.e., reports something that maximizes his underlying utility. The learner’s goal is to minimize her *Stackelberg regret*, which is the difference between her cumulative loss and the cumulative loss of her best-fixed action in hindsight, *had she allowed the agent to best respond to it*. We present a learning algorithm called **GRINDER**, which adaptively discretizes the space of  $d$ -dimensional hyperplanes in order to identify a sequence of them that minimizes Stackelberg regret. **GRINDER** presents an adaptive discretization algorithm that is tailored (and thus, limited) to the strategic setting at hand. That said, to the best of our knowledge, it was the first adaptive discretization algorithm for a non-stochastic setting.

Adaptive discretization algorithms have a long and celebrated history in online learning for Lipschitz functions when the feedback received is bandit/partial (i.e., the algorithm receives feedback only for one queried point at each round and does not get to see the whole loss function) ([Kleinberg et al., 2008b](#), [Bubeck et al., 2008](#)). The core reason for their prominence is their ability to obtain optimal instance-dependent performance guarantees. This means that they can achieve much better guarantees for “nice” instances, while scaling with the worst-case optimal rate in the worst case. However, results have mostly been obtained for the stochastic case, i.e., the case where the sequence of the Lipschitz functions is chosen stochastically. In Chapter 6, we present the first adaptive discretization learning algorithm with bandit feedback for the case that the sequence of Lipschitz functions is chosen adversarially. In fact, our algorithm is slightly more general and can even accommodate online learning of *partially Lipschitz functions*; this is important as it allows our results to seamlessly provide optimal regret bounds for adversarial dynamic pricing.

### 1.2.3 PART III: BEHAVIORAL MODEL MISSPECIFICATIONS IN INCENTIVE-AWARE ML

So far, our study has focused on algorithms that are either fully robust to incentives (Part I) or that they can at least take into account and adapt to the agents’ strategizing (Part II). All of our results, however, made the implicit assumption that all of the agents satisfy *exactly* the behavioral assumption that we have posited for them. Although this assumption is often required to provide mean-

ingful guarantees against strategic agents (and is also satisfied in settings with extensive market research), it fails to capture what happens in cases where there are a few model misspecifications that lead in agent responses that are inconsistent with the assumed behavioral model.

In Part III, we explore ways to construct algorithms in the presence of model misspecifications from the side of the agents. A key contribution here is the presentation of a model for behavior misspecifications via the “Adversarial Corruptions” framework (originally introduced by Lykouris et al. (2018) in the context of multi-armed bandits). The workhorse application here is the problem of *contextual search*, a fundamental generalization of classical binary search to higher dimensions with direct applications to pricing and personalized medicine (Cohen et al., 2020, Bastani and Bayati, 2020, Lobel et al., 2018). In the most standard version of contextual search, at every round  $t \in [T]$ , a *context*  $x_t \in \mathbb{R}^d$  arrives. Associated with this context is an unknown *true value*  $v_t \in \mathbb{R}$ , which we here assume is a linear function of the context so that  $v_t = \langle \theta^*, x_t \rangle$  for some unknown vector  $\theta^* \in \mathbb{R}^d$ , called the *ground truth*. Based on the observed context  $x_t$ , the decision-maker or *learner* selects a *query*  $\omega_t \in \mathbb{R}$  with the goal of minimizing some *loss* that depends on the query as well as the true value; examples include the *absolute/symmetric loss*,  $|v_t - \omega_t|$ , and the  $\varepsilon$ -*ball loss*,  $\mathbb{1}\{|v_t - \omega_t| > \varepsilon\}$ , both of which measure discrepancy between  $\omega_t$  and  $v_t$ . Finally, the learner observes whether or not  $v_t \geq \omega_t$ . Importantly, the true value  $v_t$  is never revealed, nor is the loss that was suffered.

For example, in feature-based dynamic pricing (Cohen et al., 2020, Lobel et al., 2018), say, of Airbnb apartments, each context  $x_t$  describes a particular apartment with components, or features, providing the apartment’s location, cleanliness, and so on. The true value  $v_t$  is the price an incoming customer or *agent* is willing to pay, which is assumed to be a linear function (defined by  $\theta^*$ ) of  $x_t$ . Based on  $x_t$ , the platform decides on a price  $\omega_t$ . If this price is less than the customer’s value  $v_t$ , then the customer makes a reservation, yielding revenue  $\omega_t$ ; otherwise, the customer passes, generating no revenue. The platform observes whether the reservation occurred (that is, if  $v_t \geq \omega_t$ ). The natural loss in this setting is called the *pricing loss*, which captures how much revenue was lost relative to the maximum price that the customer was willing to pay.

A key challenge in contextual search is that the learner only observes *binary feedback*, i.e., whether or not  $v_t \geq \omega_t$ . This contrasts with classical machine learning where the learner observes either

the entire loss function (full feedback) or the loss for just the chosen query (bandit feedback). In Chapter 7, we present the first contextual search algorithms that are robust to a few model misspecifications. We impose no assumptions on the order of corrupted rounds and obtain regret guarantees that gracefully degrade with their number while attaining near-optimality when all agents behave according to the linear model. Formally, if  $C$  is the total number of corrupted rounds, the regret bounds we obtain are of the form:  $R(T) = \mathcal{O}(Cd^3 \log^3 T)$ , when the optimal guarantee for the non-corrupted setting is  $R(T) = \mathcal{O}(d \log T)$ .

The bounds presented in Chapter 7 are not optimal, but they represent the first attempt on robustifying standard contextual search to few model misspecifications. Our algorithms are based on the idea of maintaining a knowledge set (i.e., a version space) with all the parameters that are still consistent with  $\theta^*$  and eliminating parts of the knowledge space according to the agents' responses. The knowledge-set-based techniques have been the golden standard in the vanilla (i.e., uncorrupted) contextual search literature in order to obtain the optimal regret bounds (Cohen et al., 2020, Liu et al., 2021). Surprisingly, in Chapter 8 we present a completely different approach for contextual search which is based on maintaining a carefully crafted probability distribution over the original version space for  $\theta^*$ . This approach offers much more efficient algorithms and with much better regret guarantees for corruption-robust contextual search.

#### 1.2.4 PART IV: FAIRNESS CONSIDERATIONS IN INCENTIVE-AWARE ML

The models in the incentive-aware learning literature that we have seen so far (and most – if not all – models on that matter) typically make a *full transparency* assumption; that is, the agents *fully observe* the deployed scoring/decision rule (Hardt et al., 2016, Dong et al., 2018, Chen et al., 2020b, Bechavod et al., 2021, Shavit et al., 2020, Hu et al., 2019, Kleinberg and Raghavan, 2020). However, in reality, such a full-transparency assumption is *far-fetched*. For example, since credit scoring rules are often proprietary, banks or financial agencies never make their ML models fully transparent to outsiders. Instead, they may only provide some *labeled examples* (e.g., past applicants granted loans) or *explanations* (e.g., ways to improve one's credit score).

As the actual scoring rule in place is not directly observable, agents naturally attempt to infer it using other sources of information, which may differ greatly across different individuals.

This is the case when the population is naturally clustered (due to e.g., their demographic, geographic, and cultural differences) and people have the tendency of *observation learning* (Bandura, 2008, Apesteguia et al., 2007)—that is, agents learn by observing others within their communities. For example, when applying for a loan at a specific bank, individuals may learn from the past experiences of their peers/friends (i.e., their applications and loan decisions) in order to gauge the decision rule. As a result, individuals from different peer-networks may form different ideas about the decision rule, which can lead to disparities in strategic investments and outcomes. To make things worse, there is often a regulatory requirement that the *same* decision rule be used on all subgroups (due to e.g., the risk of redlining Hunt (2005)) and thus, the decision-maker cannot use group-specific decision rules to mitigate the adverse effects of information discrepancy.

In Part IV, our goal is to study the societal impact that a benevolent decision-maker has on the welfare of the different subgroups given the information discrepancy that arises due to observation learning. We think of the welfare as a measure of the quality improvement that the agents experience (e.g., the amount by which their creditworthiness actually increases by participating in this game). Specifically, we identify necessary and sufficient conditions so that the quality improvement is equalized among subgroups and find that it is possible in some important cases of interest (e.g., when the subpopulations have either minimal or full information overlap).

### 1.2.5 PART V: INCENTIVE-AWARE MACHINE LEARNING BEYOND MYOPIC AGENTS

Lastly, in Part V, we focus on individuals/agents who are non-myopic but instead have some sort of “memory” when deciding on their current actions. Contrary to the viewpoint adopted in all previous chapters, when agents are non-myopic, their decisions at the present round may somehow influence their decisions at later rounds. Alternatively, non-myopic agents do not just base their decisions on only the present round but consider a horizon of rounds for which they have maintained “memory”.

The workhorse application of the results of Part V is online advertising, which generates billions of dollars in yearly revenue (pwc report, 2020). As we mentioned earlier, there are three key “players” in online ad auctions, and oftentimes their objectives are not fully aligned. First, the decision-maker wishes to decide the optimal placing of ads in a way that simultaneously generates

good revenue and guarantees that the users of the system will want to engage with the present and future ads. Second, the users want ads that are useful, engaging, and generate value for them. Third, the advertisers want to both have their ads shown but also want to understand at what price they should bid in order to balance their budget constraints. In Part V we focus on the ML problems that arise as part of this three-way tension.

In Chapter 10, we address online advertising settings where the value of the advertiser/bidder is unknown to her, evolving in an arbitrary manner and observed only if the bidder wins an allocation. We leverage the structure of the utility of the bidder and the partial feedback that bidders typically receive in auctions, in order to provide algorithms with regret rates against the best fixed bid in hindsight, that are exponentially faster in convergence in terms of dependence on the action space, than what would have been derived by applying a generic bandit algorithm and almost equivalent to what would have been achieved in the full information setting. Our results are enabled by analyzing a new online learning setting with outcome-based feedback, which generalizes learning with feedback graphs. We provide an online learning algorithm for this setting, of independent interest, with regret that grows only logarithmically with the number of actions and linearly only in the number of potential outcomes (the latter being very small in most auction settings). Last but not least, we show that our algorithm outperforms the bandit approach experimentally and that this performance is robust to dropping some of our theoretical assumptions or introducing noise in the feedback that the bidder receives.

We next look at the problem from the side of the decision-maker in Chapter 11. The success of the chosen ads depends crucially on the increased engagement of the users with them, who do so according to their “value”/“utility” for the content it represents. Understanding better what drives user engagement has been a major research question since the advent of online advertising not just because of its potential to drive revenue, but also, due to its potential to increase user satisfaction. Despite the proliferation of models put forth to explain user behavior, most of them have focused on users that are short-sighted/myopic; i.e., users who make engagement decisions not caring about their future interactions with the system.

A landmark paper by Hohnhold, O’Brien, and Tang (2015) defines a model of user behavior that accounts for *long-term effects* and empirically evaluates it in the context of the Google auction.

Hohnhold et al. (2015) describe the phenomenon of *ad-blindness* and *ad-sightedness*, in which a user changes their inherent propensity to click on or interact with ads based on the quality of previously viewed ads. For example, click-baits may be more likely to generate a click now, but are also likely to decrease the user’s happiness with the system and hence, click less often in the future (ad blindness). Instead, a high quality ad may lead to higher user engagement in the future (ad sightedness). In Chapter 11, we study the user behavior model of (Hohnhold et al., 2015) from a completely different angle: that of bandit optimization. The learner can choose between which ads to show to a user and each ad has both an intrinsic clickability as well as an effect on the users propensity to click on future ads. Both are initially unknown to the learner, who can only observe clicks. Our main contribution is an algorithm for scheduling optimally the sequence of ads in a way that accounts for the long-term effects from user satisfaction.

### 1.3 BIBLIOGRAPHIC REMARKS

The results of this dissertation have been part of collaborations of the author with a set of wonderful researchers. The results of Chapter 3 have appeared in (Chen et al., 2018) and were part of joint work with Yiling Chen, Ariel D. Procaccia, and Nisarg Shah. We also include an extension of the original paper which is currently being prepared for a journal submission jointly with the original team of authors and Ioannis Caragiannis and Panagiotis Tsamopoulos. The results of Chapter 4 have appeared in (Freeman et al., 2020) and were part of joint work with Rupert Freeman, David Pennock, and Jennifer Wortman Vaughan. Chapter 5 appeared in (Chen et al., 2020b) and was joint work with Yiling Chen and Yang Liu. Chapter 6 appeared in (Podimata and Slivkins, 2021) and was joint work with Aleksandrs Slivkins. Chapter 7 appeared in (Krishnamurthy et al., 2021) and was joint work with Akshay Krishnamurthy, Thodoris Lykouris, and Robert Schapire. Chapter 8 appeared in (Paes Leme et al., 2022) and were joint work with Renato Paes Leme and Jon Schneider. Chapter 9 appeared in (Bechavod et al., 2022) and was joint work with Yahav Bechavod, Steven Wu, and Juba Ziani. Chapter 10 appeared in (Feng et al., 2018) and was joint work with Zhe Feng<sup>¶</sup> and Vasilis Syrgkanis. Finally, Chapter 11 is a currently working paper with Renato Paes Leme and Khashayar Khosravi (Khosravi et al., 2022).

---

<sup>¶</sup>This paper was also discussed in Zhe Feng’s PhD dissertation.

#### 1.4 FUNDING

The work presented in this dissertation has been partially supported by a MSR Dissertation Grant, a Siebel Scholarship. In the earlier years of my PhD, my work was being supported in part under grant No. CCF-1718549 of the National Science Foundation and the Harvard Data Science Initiative.

# 2

## Background

Before we delve into the specifics of each chapter, we outline here some background notions that we will frequently use. Most of it focuses on ML-theoretic concepts. We include also a short background on some basics in Algorithmic Game Theory, which also shed light on the different regret notions that have appeared in the literature.

### 2.1 OFFLINE LEARNING

Offline learning is one of the fundamental primitives in ML. In its most basic form (the agnostic supervised learning model), the learner receives access to a set of labeled training data  $S = (x_i, y_i)_{i \in [n]} \sim_{\mathcal{D}} \mathcal{X} \times \mathcal{Y}$ , where  $n$  is the size of the training set  $\mathcal{X}$  is the feature space,  $\mathcal{Y}$  is the label space (usually  $\mathcal{Y} = \{-1, +1\}$ ) and  $\mathcal{D}$  is a fixed joint distribution used for sampling the training

data. For any hypothesis  $h$  from a hypothesis class  $\mathcal{H}$  (say e.g., all linear functions) we distinguish between the true and the empirical error of  $h$  which are defined respectively as:

$$\text{err}_{\mathcal{D}}(h) = \Pr_{(x,y) \sim \mathcal{D}} [h(x) \neq y] \quad \text{and} \quad \text{err}_S(h) = \sum_{i \in [n]} \mathbb{1}\{h(x_i) \neq y_i\}$$

Of course, the learner wants to minimize the true error  $\text{err}_{\mathcal{D}}(h)$  by identifying an appropriate hypothesis  $h$ . In the absence of full knowledge of  $\mathcal{D}$ , the best that the learner can do instead is identify a hypothesis  $h$  that minimizes the empirical error on sample dataset  $S$ . Traditionally, this is done using the Empirical Risk Minimization Algorithm (ERM) which returns hypothesis  $\tilde{h} = \arg \min_{h \in \mathcal{H}} \text{err}_S(h)$ .

There are of course a lot of interesting and deep theoretical results in offline learning. We choose to stop our exposition here, because this really is all we need for the purposes of this dissertation. We use offline learning in Chapter 3.

## 2.2 ONLINE LEARNING

Online learning is another fundamental primitive in ML, used to capture situations where the environment is changing dynamically and evolving over time. Any online learning interaction happens over a sequence of rounds, denoted by  $t = 1, \dots, T$ . In its most standard form, at round  $t$  the learner chooses an action  $x_t \in \mathcal{X}$  and nature (aka the “adversary”) chooses a loss function  $\ell_t \in \mathcal{L} : \mathcal{X} \rightarrow [0, 1]$ . Then, there are two cases. First, if the adversary reveals the whole function  $\ell_t$ , then we have a *fully information* setting. If the adversary reveals only  $\ell_t(x_t)$ , then we have a *partial information* setting. Regardless of whether we have a full or partial information setting, the standard performance measure is called *regret* and is defined as:

$$\text{Regret}(T) := \mathbb{E} \left[ \sum_{t \in [T]} \ell_t(x_t) - \min_{x^* \in \mathcal{X}} \sum_{t \in [T]} \ell_t(x^*) \right] \quad (2.2.1)$$

where the expectation is taken with respect to the learner’s randomness. The learner’s goal is to choose a sequence of actions  $\{x_t\}_{t \in [T]}$  which achieve sublinear regret in  $T$ . These algorithms are called *no regret*, since their average regret  $\text{Regret}/T$  goes to 0 as  $T$  grows large (i.e.,  $T \rightarrow \infty$ ).

An original formulation of online learning in the full formulation setting was studied in the groundbreaking works of [Hannan \(1957\)](#) and [Blackwell \(1954, 1956\)](#) who provided no-regret algorithms with regret  $\mathcal{O}(T \cdot \text{poly}(|\mathcal{X}|))$ . The subsequent and seminal works of [Littlestone and Warmuth \(1994\)](#), [Freund and Schapire \(1996\)](#) and [Vovk \(1998\)](#) provided improved regret guarantees  $\mathcal{O}(\sqrt{T \cdot \log |\mathcal{X}|})$  for the problem. Improved regret guarantees can be obtained when the family of the loss functions  $\mathcal{L}$  is more structured. For example, if  $\mathcal{L}$  corresponds to Lipschitz continuous loss functions, then [Zinkevich \(2003\)](#) shows that there are algorithms with  $\mathcal{O}(\sqrt{T})$  regret or if  $\mathcal{L}$  corresponds to strongly convex functions, then [Hazan et al. \(2007\)](#) show algorithms with regret rates  $\mathcal{O}(\log T)$ .

As for the partial information setting, when  $\mathcal{L}$  is the space of all Lipschitz functions then the optimal regret bounds are  $\tilde{\mathcal{O}}(T^{(z+1)/(z+2)})$ ,<sup>\*</sup> where  $z$  is a scalar called the *zooming dimension* which roughly encodes how “well-behaved” the sequence of  $\{\ell_t\}_t$  is ([Kleinberg et al., 2019](#), [Bubeck et al., 2008](#)). In the worst-case, it holds that  $z = d$ . When  $\mathcal{L}$  is the set of linear functions, [Dani et al. \(2007\)](#) provided an algorithm in the partial information setting with regret  $\mathcal{O}(\text{poly}(d)\sqrt{T})$  and subsequently, [Abernethy et al. \(2008\)](#) obtained an efficient algorithm with the same regret bound. Extensive research has also been devoted in the version of the problem of online learning with partial information when  $\mathcal{L}$  corresponds to the set of convex functions. [Flaxman et al. \(2005\)](#) were the first ones to provide an algorithm with regret  $\mathcal{O}(d^{1/2}T^{5/6})$  for general bounded convex functions and  $\mathcal{O}(d^{1/2} \cdot T^{3/4})$  in the case where the functions are also Lipschitz. Optimizing for the dependence on the time horizon  $T$ , the best known bound currently is due to [Bubeck et al. \(2015\)](#) and achieves a regret bound of  $\tilde{\mathcal{O}}(d^{9.5}\sqrt{T})$  with a polynomial time algorithm.

We make extensive use of online learning in the present dissertation. Each chapter includes more in depth introductions regarding the specifics of each setting considered; for example, we discuss prediction with expert advice in Chapter 4). As we discuss in the following chapters, although there is a very good understanding of “structured” loss functions against stochastic and adversarial adversaries, this is far from the truth whenever the adversaries that we face are strategic.

---

<sup>\*</sup>Where the  $\tilde{\mathcal{O}}(\cdot)$  notation hides terms polylogarithmic in  $T$ , as is customary.

### 2.3 MULTI-ARMED BANDITS

A particular subclass of problems within online learning with partial information that has individually received significant attention is the one of *Multi-Armed Bandits* (MAB). Multi-armed bandits is a simple yet powerful model for decision-making under uncertainty, extensively studied since the 1950ies and exposed in several books, e.g., (Bubeck and Cesa-Bianchi, 2012, Slivkins, 2019, Lattimore and Szepesvári, 2020). In the most standard form of a MAB setting, there is a set of  $K$  actions (aka “arms”) from which the learner can choose to play. At each round  $t$ , the adversary chooses (stochastically or adversarially) the losses for each arm  $\ell_{i,t} \in [0, 1], \forall i \in [K]$ , the learner chooses to play an action  $I_t \in [K]$ , and subsequently receives *bandit feedback*  $\ell_{I_t,t}$ . The MAB problem has been analyzed in numerous other variations (we even study one in this very dissertation in Chapter 11).

We outline next a fundamental algorithm for adversarial MAB due to Auer et al. (2002b) called EXP3 which we will keep referring to at various points in the rest of the dissertation. At a high level, the algorithm works by balancing *exploration* and *exploitation*. It does so by maintaining unbiased estimates of the loss of each arm and then feeding these estimates in the computation of the probability distribution from which each round’s action is chosen. In reality, EXP3 is the version of the well-known Hedge/MWU algorithms (discussed extensively in Chapter 4) with unbiased loss estimates, rather than exact knowledge of the losses.

---

**Algorithm 2.1:** Vanilla EXP3

---

- 1 Set learning parameter  $\eta = \sqrt{2 \ln K / K}$ .
- 2 Let  $\pi_1$  be the uniform probability distribution over the  $K$  arms.
- 3 **for** rounds  $t = 1, \dots, T$  **do**
- 4     Draw arm  $I_t$  from distribution  $\pi_t$ .
- 5     Compute the estimated losses for the arms:

$$\forall i \in [K] : \quad \tilde{\ell}_{i,t} = \frac{\ell_{i,t} \cdot \mathbb{1}\{I_t = i\}}{\pi_t}$$

- 6 Update probability distribution as:

$$\forall i \in [K] : \quad \pi_{i,t} = \frac{\exp\left(-\eta \sum_{\tau \in [t]} \tilde{\ell}_{i,\tau}\right)}{\sum_{j \in [K]} \exp\left(-\eta \sum_{\tau \in [t]} \tilde{\ell}_{j,\tau}\right)}$$


---

EXP3 achieves regret  $\mathcal{O}(\sqrt{TK \ln K})$ . Note that in this expository/background part, we have hid under the rug the distinction between *pseudo-* and *expected* regret as these details are not important for now. That said, the regret defined in Equation (2.2.1) corresponds to the expected regret.

## 2.4 ALGORITHMIC GAME THEORY BACKGROUND

We review here some basic background on Algorithmic Game Theory (AGT) directly setup in the context of ML. In any AGT setting, players/agents have *incentives* to act according to a particular utility  $u(\cdot) \in [0, 1]$  which governs their behavior. Throughout this dissertation, we will be encountering mostly *quasi-linear* utilities of the form:

$$u(x, x') = \text{value}(x') - \text{cost}(x, x')$$

where  $x$  is the *true* datapoint of the agent, and  $x'$  is the *reported* one (i.e., potentially  $x' \neq x$ ). For example, if we are modeling a linear regression setting, where the output hyperplane is denoted by its normal vector  $\alpha$ , and by  $x, x' \in \mathbb{R}^d$  we denote the original and the reported feature vector of the agent, then a standard formulation for the value function is  $\text{value}(x') = \langle \alpha, x' \rangle$  (i.e., it assigns higher numerical value to the points further away from  $\alpha$ ) and for the cost function  $\text{cost}(x, x') = \|x - x'\|_2$  (i.e., the cost is measured as the *L2* distance between the true and the reported point). We oftentimes call the “reported” point the ‘manipulated’ one.

As we said,  $x'$  is the reported point and can be (in general) different than the true one. In this dissertation, we posit that agents are *best-responding*, i.e.,  $x'$  is chosen such as to maximize  $u(x, x')$ :  $x' = \arg \max_{\tilde{x}} u(x, \tilde{x})$ . Roughly speaking, an algorithm is *incentive-compatible* if truthtelling (i.e.,  $x' = x$ ) is the agent’s best strategy. We discuss extensively truthfulness and incentive compatibility in Part I of the present dissertation.

## 2.5 REGRET NOTIONS AGAINST STRATEGIC AGENTS

We conclude this background section by discussing how the standard definition of regret gets changed in settings where the learner interacts with a strategic agent (rather than nature or a worst-case adversary). We define the model of an online Stackelberg game played between the learner

(aka “leader” in Stackelberg games) and the agent (aka “follower” in Stackelberg games). At each round  $t$ , the learner commits to an action  $\alpha_t \in \mathcal{A}$  (where  $\mathcal{A}$  is an abstract allowable action set for the learner), the agent observes  $\alpha_t$  and reports  $\hat{x}_t(\alpha_t) \in \mathcal{X}$  (where  $\mathcal{X}$  is an abstract allowable action set for the agent). The agent’s  $\hat{x}_t(\alpha_t)$  is chosen as the best response to their underlying utility function  $u(x_t, x', \alpha_t)$ , i.e.,  $\hat{x}_t(\alpha_t) = \arg \max_{x' \in \mathcal{X}} u(x_t, x', \alpha_t)$ . In general, the utility function can be different at each round, but we use the same utility here for simplification. After the agent best-responds, the learner experiences loss  $\ell_t(\alpha_t, \hat{x}_t(\alpha_t))$ . In this setting, where the agent is strategic when replying to the learner, there are three different ways to define regret.

The first way is the standard regret definition (see also Equation (2.2.1)) also known as the *external regret*, which has as a benchmark the learner’s best fixed action in hindsight, had the agents not been able to best-respond to it:

$$\text{Regret}(T) := \mathbb{E} \left[ \sum_{t \in [T]} \ell_t(\alpha_t, \hat{x}_t(\alpha_t)) - \min_{\alpha_E \in \mathcal{A}} \sum_{t \in [T]} \ell_t(\alpha_E, \hat{x}_t(\alpha_t)) \right]$$

The second way is called *strategic regret*. In strategic regret, the benchmark that the learner compares to is the learner’s best fixed action in hindsight if the agents were truthful:

$$\text{Regret}_{\text{strat}}(T) := \mathbb{E} \left[ \sum_{t \in [T]} \ell_t(\alpha_t, \hat{x}_t(\alpha_t)) - \min_{\alpha_{\text{strat}} \in \mathcal{A}} \sum_{t \in [T]} \ell_t(\alpha_{\text{strat}}, x_t) \right]$$

By definition, if the learner’s algorithm is incentive-compatible, then the two notions of external and strategic regret coincide (see more in Chapter 4). Finally, the third way to define regret in strategic settings is through the *Stackelberg regret* where the benchmark is the best-fixed action for the learner in hindsight, had they given the opportunity to the agents to best-respond, i.e.,

$$\text{Regret}_{\text{Stackelberg}}(T) := \mathbb{E} \left[ \sum_{t \in [T]} \ell_t(\alpha_t, \hat{x}_t(\alpha_t)) - \min_{\alpha^* \in \mathcal{A}} \sum_{t \in [T]} \ell_t(\alpha^*, \hat{x}_t(\alpha^*)) \right]$$

We remark here that although it may look like there is a clear hierarchy between the aforementioned regret notions, in Chapter 5 we show that this is not true for general settings.

## **Part I**

# **Robustness to Strategic Individuals**

# 3

## Strategyproof Linear Regression in High Dimensions

### 3.1 CHAPTER OVERVIEW

As we have seen in Chapter 1, standard ML algorithms may inadvertently incentivize agents to misreport their data, in efforts to benefit from the altered outputs of the algorithms. In this chapter, we subscribe to the agenda of robustifying the Machine Learning algorithms by aligning the agents' incentives and advance it in the context of the ubiquitous problem of linear regression, i.e., fitting a hyperplane through given data. In our setting, the agents can manipulate their dependent variables to minimize their vertical distance from the output hyperplane, and our goal is to design

strategyproof regression mechanisms without payments.

A bit more formally, we study a linear regression setting in which the task is to fit a hyperplane through data points  $(\mathbf{x}_i, y_i)$  for  $i \in \{1, \dots, n\}$ , where  $\mathbf{x}_i \in \mathbb{R}^d$  are the independent variables and  $y_i \in \mathbb{R}$  is the dependent variable. Following [Dekel et al. \(2010\)](#) and [Perote and Perote-Peña \(2004\)](#), we assume that the independent variables are public information, but dependent variable  $y_i$  is held privately by agent  $i$ . A mechanism elicits the private information of the agents, and returns a hyperplane represented by vector  $\beta = (\beta_1, \beta_0) \in \mathbb{R}^{d+1}$ . Under this outcome, the residual for agent  $i$  is  $r_i = y_i - \beta_1^\top \mathbf{x}_i - \beta_0$ , and, loosely speaking, agents wish to minimize  $|r_i|$  (see Chapter 3.2 for a precise description of agent preferences).

Our starting point is the work of [Dekel et al. \(2010\)](#), who show that empirical risk minimization (ERM) with the  $L_1$  loss (in short,  $L_1$ -ERM), coupled with a specific tie-breaking rule, is group strategyproof, that is, no coalition of agents can be weakly better off by misreporting.  $L_1$ -ERM regression algorithms are members of a much wider class of algorithms known as *quantile regression* algorithms. We show that quantile regressions (even extended with convex regularizers) are *group strategyproof*. We next seek a broader understanding of what is possible in terms of strategyproof regression mechanisms in our setting.

To that end, we look to the work of [Perote and Perote-Peña \(2004\)](#), who focus on the two-dimensional case (known as *simple linear regression*), i.e., fitting a line through points on a plane. They propose a wide family of strategyproof mechanisms, which they call *clockwise repeated median* (CRM) mechanisms. These mechanisms are parametrized by two subsets of agents  $S$  and  $S'$ . [Perote and Perote-Peña \(2004\)](#) establish conditions on  $S$  and  $S'$  under which they claim that CRM mechanisms are strategyproof. We identify a mistake in this result, present counterexamples showing violation of strategyproofness under their conditions, and identify three stricter conditions under which we can recover strategyproofness — in fact, we prove group strategyproofness. Under one of our conditions, CRM mechanisms coincide with a family of mechanisms from the statistics literature known as *resistant line mechanisms* [Johnstone and Velleman \(1985\)](#). The present chapter therefore establishes the group strategyproofness of these mechanisms.

Our main result is that we generalize the CRM family to higher dimensions. We introduce the family of *generalized resistant hyperplane* (GRH) mechanisms, which, to the best of our knowledge,

is the first extension of resistant line mechanisms beyond the plane. In  $d + 1$  dimensions, GRH mechanisms are parametrized by  $d + 1$  subsets of agents. Through a surprising connection to the literature on the Ham Sandwich Theorem, we find a condition on the subsets under which GRH mechanisms are group strategyproof. Strikingly, our proof of this general *group strategyproofness* result in *any number* of dimensions is much shorter than the (incorrect) proof of [Perote and Perote-Peña \(2004\)](#) for the *strategyproofness* of CRM mechanisms in *two dimensions*.

We also study a property called impartiality, which is stricter than strategyproofness. We establish the existence of a wide family of impartial mechanisms, which, unlike our generalized  $L_1$ -ERM and generalized resistant hyperplane mechanisms, are strategyproof but not group strategyproof (except for constant functions). Building upon the work of [Moulin \(1980\)](#), we also provide two non-constructive characterizations of strategyproof mechanisms for linear regression.

Strategyproofness is not the sole desideratum; constant functions (e.g., hyperplane  $y = 0$ ) are strategyproof but not necessarily desirable. We would also like the mechanism to have good statistical efficiency. For that, we compare (families of) strategyproof mechanisms in terms of their approximation of the optimal squared loss, leveraging our characterization. Most importantly, we establish a lower bound of 2 on the approximation ratio of any strategyproof mechanism, which means that any mechanism that is even close to *ordinary least squares* must be manipulable.

### 3.1.1 RELATED WORK

As discussed above, the results of this chapter are most closely related to that of [Perote and Perote-Peña \(2004\)](#) and [Dekel et al. \(2010\)](#). Here, we give a broader picture of the state of research on machine learning algorithms that are *robust* to strategic noise. This research can be categorized using three key axes: (i) manipulable information, (ii) goal of the agents, and (iii) use of payments and incentive guarantees.

On the first axis, similarly to this chapter, one strand of literature assumes that the independent variables (or *feature vectors* in the language of classification) are public information, and the dependent variables (labels) are private, manipulable information ([Dekel et al., 2010](#), [Meir et al., 2012](#), [Perote and Perote-Peña, 2003, 2004](#), [Hossain and Shah, 2020](#)). We have already highlighted the works of [Dekel et al. \(2010\)](#) and [Perote and Perote-Peña \(2004\)](#). [Meir et al. \(2012\)](#) provide strong

positive results for designing strategyproof classifiers when there are either only two classifiers, or the agents are interested in a shared set of input points. Similar to our setting, [Hossain and Shah \(2020\)](#) also study linear regression settings and show that when manipulations are bounded, then every empirical risk minimization algorithm with a convex regularizer admits a pure Nash equilibrium. In this chapter, however, agents can have unbounded manipulations, and we provide mechanisms with the stronger guarantee of strategyproofness. The other strand of literature is mostly known under the name “strategic classification”. We thoroughly analyze the literature around strategic classification and the different perspectives it entails in Chapters 5 and 9.

On the second axis, one line of research focuses on agents motivated by privacy concerns, with a tradeoff between accuracy and privacy ([Cummings et al., 2015](#), [Cai et al., 2015](#)). Another focuses on agents who want the algorithm to make accurate assessment on their own sample, even if this reduces the overall accuracy. This form of strategic manipulation has been studied for estimation ([Caragiannis et al., 2016](#)), classification ([Meir et al., 2010, 2011, 2012](#)), and regression ([Perote and Perote-Peña, 2004](#), [Dekel et al., 2010](#), [Hossain and Shah, 2020](#)) problems. A third line of research focuses on agents who wish to achieve better outcomes for *themselves* irrespective of how accurate these outcomes are to their true sample ([Hardt et al., 2016](#), [Dong et al., 2018](#), [Chen et al., 2020b](#), [Ahmadi et al., 2021](#), [Milli et al., 2019](#), [Hu et al., 2019](#), [Braverman and Garg, 2020](#)). The problem discussed in this chapter falls squarely into the second category.

Finally, on the third axis, various papers differ on whether monetary payments to agents are allowed ([Cai et al., 2015](#)), and on how strongly to guarantee truthful reporting: the stronger strategyproofness requirement ([Perote and Perote-Peña, 2003, 2004](#), [Meir et al., 2012](#)) versus the weaker Bayes-Nash incentive compatibility ([Ioannidis and Loiseau, 2013](#), [Cummings et al., 2015](#)). This chapter falls into the literature of mechanism design without money; we study linear regression mechanisms that enforce strategyproofness without paying, or asking the agents to pay.

## 3.2 MODEL

Let  $[k] \triangleq \{1, \dots, k\}$  be the set of first  $k$  natural numbers, and  $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$  be the extended real line. Given numbers  $t_1, \dots, t_k \in \overline{\mathbb{R}}$ , let  $\min(t_1, \dots, t_k)$  denote the smallest value, and  $\min^j(t_1, \dots, t_k)$  denote the  $j^{\text{th}}$  smallest value. Let  $\text{med}(t_1, \dots, t_k)$  denote their median: when  $k$  is odd, this is equal

to  $\min^{(k+1)/2}(t_1, \dots, t_k)$ , but when  $k$  is even, this could be either  $\min^{k/2}(t_1, \dots, t_k)$  (the “left median”) or  $\min^{k/2+1}(t_1, \dots, t_k)$  (the “right median”).\*

Our work focuses on the problem of linear regression, i.e., fitting a hyperplane through given data. Let  $N = [n]$ . We are given a collection of data points  $\mathcal{D} = (\mathbf{x}_i, y_i)_{i \in N}$ , where  $\mathbf{x}_i \in \mathbb{R}^d$  and  $y_i \in \mathbb{R}$  are called the *independent* and *dependent* variables of point  $i$ , respectively. Let  $\bar{\mathbf{x}}_i = (\mathbf{x}_i, 1)$ . Our goal is to find a vector  $\beta = (\beta_1, \beta_0) \in \mathbb{R}^{d+1}$  such that  $\beta^\top \bar{\mathbf{x}}_i = \beta_1^\top \mathbf{x}_i + \beta_0$  is a good approximation of  $y_i$  for each  $i \in N$ . The quantity  $r_i = y_i - \beta^\top \bar{\mathbf{x}}_i$  is called the residual of point  $i$ .

**Strategic setting.** We study a setting in which each data point  $p_i = (\mathbf{x}_i, y_i)$  is provided by a strategic agent  $i \in N$ . Following [Perote and Perote-Peña \(2004\)](#) and [Dekel et al. \(2010\)](#), we assume that the independent variables  $\mathbf{x} = (\mathbf{x}_i)_{i \in N}$  constitute *public* information, which the agents cannot manipulate. Each agent  $i$  holds the dependent variable  $y_i$  as private information, and may report a different value  $\tilde{y}_i$  in order to receive a more preferred outcome. Thus, the principal observes the reported data points  $\tilde{\mathcal{D}} = (\mathbf{x}_i, \tilde{y}_i)_{i \in N}$ . Let us denote  $\mathbf{y} = (y_i)_{i \in N}$  and  $\tilde{\mathbf{y}} = (\tilde{y}_i)_{i \in N}$ .

**Mechanisms.** Because the agents cannot change  $\mathbf{x}$ , we can effectively treat it as fixed. A mechanism for linear regression  $M^x$  is therefore defined for given public information  $\mathbf{x}$ , takes as input reported private information  $\tilde{\mathbf{y}}$ , and returns a vector  $\beta$ . We omit  $\mathbf{x}$  when it is clear from the context.

**Agent preferences.** When a mechanism returns  $\beta$ , we say that the outcome for agent  $i$  is  $\hat{y}_i(\beta) = \beta^\top \mathbf{x}_i$ . We omit  $\beta$  when it is clear from the context. The agent only cares about her own outcome  $\hat{y}_i$ , and would like it to be as close to  $y_i$  as possible. Formally, we assume that agent  $i$  has *single-peaked preferences* [Black \(1958\)](#), [Moulin \(1980\)](#) over  $\hat{y}_i$  with peak at  $y_i$ . We represent the weak preference relation by  $\succ_i$  and the strict preference relation by  $\succ_i$ . Formally, for all  $a, b \in \mathbb{R}$ ,  $y_i > a \geq b$  or  $y_i < a \leq b$  must imply  $y_i \succ_i a \succ_i b$ .

**Game-theoretic desiderata.** Our goal is to prevent agents from misreporting their private information. The game theory literature offers a strong desideratum under which agents have no incentive to misreport even if they know what the other agents would report.

**Definition 3.1** (Strategyproofness). *A mechanism  $M^x$  is called strategyproof (SP) if each agent weakly*

---

\*This is different from the standard definition, which takes the average of the left and right medians, but necessary to ensure incentive guarantees.

*prefers truthfully reporting her private information to misreporting it, regardless of the reports of the other agents. Formally, for each  $i \in N$ ,  $y_i \in \mathbb{R}$ , and  $\tilde{\mathbf{y}} \in \mathbb{R}^n$ , we need  $\hat{y}_i(M^x(y_i, \tilde{\mathbf{y}}_{-i})) \succ_i \hat{y}_i(M^x(\tilde{\mathbf{y}}))$ . Note that this must hold for any possible single-peaked preferences the agent may have.*

While no individual agent can benefit from misreporting under a strategyproof mechanism, a group of agents may still be able to collude, and benefit by simultaneously misreporting. This can be prevented by imposing a stronger desideratum.

**Definition 3.2** (Group Strategyproofness). *A mechanism  $M^x$  is called group strategyproof (GSP) if no coalition of agents can simultaneously misreport in a way that no agent in the coalition is strictly worse off and some agent in the coalition is strictly better off, irrespective of the reports of the other agents. Formally, for each  $S \subseteq N$ ,  $\mathbf{y}_S = (y_i)_{i \in S} \in \mathbb{R}^{|S|}$ , and  $\tilde{\mathbf{y}} \in \mathbb{R}^n$ , it should not be the case that  $\hat{y}_i(M^x(\tilde{\mathbf{y}})) \succ_i \hat{y}_i(M^x(\mathbf{y}_S, \tilde{\mathbf{y}}_{N \setminus S}))$  for every  $i \in S$ , and the preference is strict for at least one  $i \in S$ .*

The game theory literature also considers a weaker notion of group strategyproofness in which not all the agents in a manipulating coalition should be strictly better off. We do not consider this notion because our group strategyproof mechanisms are able to satisfy the stronger notion.

Note that we do not assume that the data points are generated by an underlying statistical process. The results of this chapter are independent of how the data points were generated.

### 3.3 FAMILIES OF STRATEGYPROOF MECHANISMS

In this section, we analyze families of (group) strategyproof mechanisms for linear regression. Our results generalize existing families of mechanisms, and propose novel families.

#### 3.3.1 QUANTILE REGRESSION

Let  $\mathcal{R}$  be the class of empirical risk functions  $\hat{R} : \mathcal{F} \times \mathbb{R}_{>0}^{n \times 2} \times \mathbb{R}^{d+1} \rightarrow \mathbb{R}$  defined as follows:

$$\hat{R}(f, \mathbf{q}, \mathcal{D}) = \sum_{i \in N: y_i \geq f(\mathbf{x}_i)} q_{i1} \cdot |y_i - f(\mathbf{x}_i)| + \sum_{i \in N: y_i < f(\mathbf{x}_i)} q_{i2} \cdot |y_i - f(\mathbf{x}_i)|$$

for a function  $f \in \mathcal{F}$ , a coefficient matrix  $\mathbf{q} \in \mathbb{R}_{>0}^{n \times 2}$ , and a set of points  $\mathcal{D} = (\mathbf{x}_i, y_i)_{i \in N}$ . Denoting by  $(x)^+ = \max\{0, x\}$ , we can alternatively write  $\widehat{R}(f, \mathbf{q}, \mathcal{D})$  as:

$$\widehat{R}(f, \mathbf{q}, \mathcal{D}) = \sum_{i \in N} (q_{i2} \cdot (f(\mathbf{x}_i) - y_i) + (q_{i1} + q_{i2}) \cdot (y_i - f(\mathbf{x}_i))^+).$$

Let  $\|\cdot\|$  be a strictly convex function such that  $\|\cdot\| : \mathcal{F} \rightarrow \mathbb{R}$ . We define the binary relation  $\succ_{\|\cdot\|, \mathbf{q}, \mathcal{D}} \subseteq \mathcal{F}^2$  (which depends on the coefficient matrix  $\mathbf{q}$  and the dataset  $\mathcal{D}$ ) as follows: for two functions  $f, g \in \mathcal{F}$ ,  $f \succ_{\|\cdot\|, \mathbf{q}, \mathcal{D}} g$  when  $\widehat{R}(f, \mathbf{q}, \mathcal{D}) < \widehat{R}(g, \mathbf{q}, \mathcal{D})$  or  $\widehat{R}(f, \mathbf{q}, \mathcal{D}) = \widehat{R}(g, \mathbf{q}, \mathcal{D})$  and  $\|f\| < \|g\|$ . In words,  $f \succ_{\|\cdot\|, \mathbf{q}, \mathcal{D}} g$  when the total risk of  $f$  on  $\mathcal{D}$  is smaller than that of  $g$  on  $\mathcal{D}$ , or the norm of  $f$  is smaller than the one of  $g$ .

We define the class  $\mathcal{M}$  of regression mechanisms that use a strictly convex function over  $\mathcal{F}$  and a coefficient matrix as parameters. A mechanism  $M \in \mathcal{M}$  takes as input  $\mathcal{D}$  and uses  $\mathbf{q}$  and  $\|\cdot\|$  to return function  $f \in \mathcal{F}$  such that  $f \succ_{\|\cdot\|, \mathbf{q}, \mathcal{D}} g$  for every other function  $g \in \mathcal{F} \setminus \{f\}$ . We show below that any mechanism in  $\mathcal{M}$  is group strategyproof.

Note that our definition includes the standard  $L_1$ -ERM, the generalizations defined by [Chen et al. \(2018\)](#), as well as (generalizations of) quantile regression.

### Theorem 3.1

Every regression mechanism in  $\mathcal{M}$  is group strategyproof.

*Proof.* We use  $\widehat{R}(f, \mathcal{D})$  instead of  $\widehat{R}(f, \mathbf{q}, \mathcal{D})$  and  $\succ_{\|\cdot\|, \mathcal{D}}$  instead of  $\succ_{\|\cdot\|, \mathbf{q}, \mathcal{D}}$ , when clear from context. Also, by slightly abusing notation, we define the loss for each agent  $i$  in  $N$  as:

$$\widehat{R}(f, \mathcal{D}_i) = q_{i2} \cdot (f(\mathbf{x}_i) - y_i) + (q_{i1} + q_{i2}) \cdot (y_i - f(\mathbf{x}_i))^+, \quad (3.3.1)$$

From this definition, it is clear that  $\widehat{R}(f, \mathcal{D}) = \sum_{i \in N} \widehat{R}(f, \mathcal{D}_i)$ .

We prove Theorem 3.1 by contradiction. To do so, we assume that there is a set of agents  $S \subseteq N$  controlling some subset of the input points, called the *manipulators*, who are the only ones that misreport. The manipulators misreport the value of their dependent variable so that all of them weakly prefer the outcome returned by the mechanism  $M$  to the original one, and at least one of them strictly prefers the new outcome to the original one.

Let  $\mathcal{D}' = (\mathbf{x}_i, y'_i)_{i \in N}$ , where  $y'_i = y_i \forall i \in N \setminus S$ , denote the new profile. We use  $f_0$  to denote the function that mechanism  $M$  returns on input dataset  $\mathcal{D}$ , and  $f_1$  the one returned on  $\mathcal{D}'$ . Let  $h \in \mathcal{F}$  be defined as  $h = (f_0 + f_1)/2$ . By the definition of  $M$ , we have

$$f_0 \succ_{\mathcal{D}} f_1, f_0 \succ_{\mathcal{D}} h, f_1 \succ_{\mathcal{D}'} h \text{ and } f_1 \succ_{\mathcal{D}'} f_0. \quad (3.3.2)$$

We next classify the agents into 5 different classes according to their values on functions  $f_0, f_1$  and  $h$ . These classes are useful in the next step of the proof, where we construct the misreports for the manipulators. The 5 different classes are:

- Class  $N_0$  consists of agents  $i$  with  $f_0(\mathbf{x}_i) = f_1(\mathbf{x}_i)$ .
- Class  $N_1$  consists of agents  $i$  with  $f_0(\mathbf{x}_i) < f_1(\mathbf{x}_i)$  and  $y_i < h(\mathbf{x}_i)$ .
- Class  $N_2$  consists of agents  $i$  with  $f_0(\mathbf{x}_i) < f_1(\mathbf{x}_i)$  and  $y_i \geq h(\mathbf{x}_i)$ .
- Class  $N_3$  consists of agents  $i$  with  $f_1(\mathbf{x}_i) < f_0(\mathbf{x}_i)$  and  $y_i > h(\mathbf{x}_i)$ .
- Finally, class  $N_4$  consists of agents  $i$  with  $f_1(\mathbf{x}_i) < f_0(\mathbf{x}_i)$  and  $y_i \leq h(\mathbf{x}_i)$ .

Observe now that agents in classes  $N_1$  and  $N_3$  do not contain any manipulators as they prefer outcome  $f_0(\mathbf{x}_i)$  to  $f_1(\mathbf{x}_i)$ , i.e.,  $|f_0(\mathbf{x}_i) - y_j| < |f_1(\mathbf{x}_i) - y_j|$ . Indeed, for all agents  $j \in N_1$ , note that  $y_j < h(\mathbf{x}_j) < f_1(\mathbf{x}_j)$ . If  $y_j \leq f_0(\mathbf{x}_j)$ , then since  $f_0(\mathbf{x}_j) < f_1(\mathbf{x}_i)$ , it holds that  $f_0(\mathbf{x}_j) - y_j < f_1(\mathbf{x}_j) - y_j$ . Otherwise (i.e.,  $f_0(\mathbf{x}_j) < y_j < f_1(\mathbf{x}_j)$ ) since  $y_j < h(\mathbf{x}_j)$  then:

$$2y_j < f_0(\mathbf{x}_j) + f_1(\mathbf{x}_j) \Leftrightarrow y_j - f_0(\mathbf{x}_j) < f_1(\mathbf{x}_j) - y_j$$

The proof is similar for the agents of group  $N_3$ . We next construct the following two auxiliary datasets  $\tilde{\mathcal{D}} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i \in N}$  and  $\tilde{\mathcal{D}'} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i \in N}$  as follows:

- For agents  $i \in N_0$ , we set  $\tilde{y}_i = \tilde{y}'_i = f_0(\mathbf{x}_i)$ .
- For agents  $i \in N_1$ , we set  $\tilde{y}_i = \tilde{y}'_i = f_0(\mathbf{x}_i)$  if  $y_i < f_0(\mathbf{x}_i)$  and  $\tilde{y}_i = \tilde{y}'_i = y_i$  otherwise.
- For agents  $i \in N_2$ , we set  $\tilde{y}_i = h(\mathbf{x}_i)$  and  $\tilde{y}'_i = f_1(\mathbf{x}_i)$ .

- For agents  $i \in N_3$ , we set  $\tilde{y}_i = \tilde{y}'_i = f_0(\mathbf{x}_i)$  if  $y_i > f_0(\mathbf{x}_i)$  and  $\tilde{y}_i = \tilde{y}'_i = y_i$  otherwise.
- Finally, for agents  $i \in N_4$ , we set  $\tilde{y}_i = h(\mathbf{x}_i)$  and  $\tilde{y}'_i = f_1(\mathbf{x}_i)$ .

Figure 3.1 includes an example of how  $\tilde{\mathcal{D}}$  and  $\tilde{\mathcal{D}}'$  are constructed for the 5 different classes.

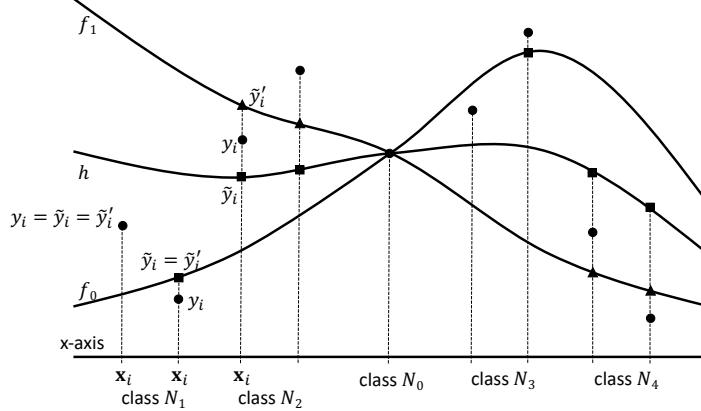


Figure 3.1: An example of the classification of agents into  $N_0$ ,  $N_1$ ,  $N_2$ ,  $N_3$ , and  $N_4$ , and the construction of the datasets  $\tilde{\mathcal{D}}$  and  $\tilde{\mathcal{D}}'$  used in the proof. Each vertical dotted line corresponds to an agent. The ball points correspond to the  $\mathcal{D}$  dataset with  $x_i$  being single-dimensional. Each square point indicates the corresponding point  $(x_i, \tilde{y}_i)$  of set  $\tilde{\mathcal{D}}$  and each triangle indicates the corresponding point  $(x_i, \tilde{y}'_i)$  of set  $\tilde{\mathcal{D}}'$ . In cases where both square and triangle points are missing, they coincide with the ball point (i.e., in the first and sixth agent from the left). In cases where the triangle point is missing, it coincides with the square point (i.e., in the second and seventh point).

Based on the construction of the datasets and the assumptions that we have made so far, we prove that given the reports in  $\tilde{\mathcal{D}}$ , function  $f_0$  has smaller error than  $h$ .

**Lemma 3.1.**  $f_0 \succ_{\tilde{\mathcal{D}}} h$ .

*Proof.* Since  $f_0 \succ_{\mathcal{D}} h$ , it must be either  $\widehat{R}(f_0, \mathcal{D}) < \widehat{R}(h, \mathcal{D})$ , or  $\widehat{R}(f_0, \mathcal{D}) = \widehat{R}(h, \mathcal{D})$  and  $\|f_0\| < \|h\|$ . So, in order to prove the lemma, it suffices to prove the equality

$$\widehat{R}(f_0, \tilde{\mathcal{D}}) - \widehat{R}(h, \tilde{\mathcal{D}}) = \widehat{R}(f_0, \mathcal{D}) - \widehat{R}(h, \mathcal{D}),$$

or equivalently, that quantity:

$$Q_i = \widehat{R}(f_0, \tilde{\mathcal{D}}_i) - \widehat{R}(h, \tilde{\mathcal{D}}_i) - \widehat{R}(f_0, \mathcal{D}_i) + \widehat{R}(h, \mathcal{D}_i) \quad (3.3.3)$$

is zero for every agent  $i \in N$ . Indeed, if  $Q_i = 0, \forall i$ , then, by summing across all agents in  $N$ , we obtain Equation (3.3.3). We distinguish five cases depending on the class of agent  $i$ .

- If  $i \in N_0$ , then by definition it holds that  $f_0(\mathbf{x}_i) = h(\mathbf{x}_i)$ . As a result,  $\widehat{R}(f_0, \widetilde{\mathcal{D}}_i) - \widehat{R}(h, \widetilde{\mathcal{D}}_i) = \widehat{R}(f_0, \mathcal{D}_i) - \widehat{R}(h, \mathcal{D}_i) = 0$  and  $Q_i = 0$ .
- If  $i \in N_1$ , we distinguish two more cases. If  $y_i = \tilde{y}_i$ , then  $\widehat{R}(f_0, \widetilde{\mathcal{D}}_i) = \widehat{R}(f_0, \mathcal{D}_i)$  and  $\widehat{R}(h, \widetilde{\mathcal{D}}_i) = \widehat{R}(h, \mathcal{D}_i)$  and, hence,  $Q_i = 0$ . If, instead,  $y_i \neq \tilde{y}_i$ , then  $y_i, \tilde{y}_i \leq f_0(\mathbf{x}_i) < h(\mathbf{x}_i)$ , and Equation (3.3.3) yields

$$\begin{aligned} Q_i &= q_{i2}(f_0(\mathbf{x}_i) - \tilde{y}_i) - q_{i2}(h(\mathbf{x}_i) - \tilde{y}_i) - q_{i2}(f_0(\mathbf{x}_i) - y_i) + q_{i2}(h(\mathbf{x}_i) - y_i) \quad (\text{Eq. (3.3.1)}) \\ &= -q_{i2}(h(\mathbf{x}_i) - f_0(\mathbf{x}_i)) - q_{i2}(f_0(\mathbf{x}_i) - y_i) + q_{i2}(h(\mathbf{x}_i) - y_i) = 0 \quad (\tilde{y}_i = f_0(\mathbf{x}_i)) \end{aligned}$$

- If  $i \in N_2$ , then  $y_i, \tilde{y}_i \geq h(\mathbf{x}_i) > f_0(\mathbf{x}_i)$  and Equations (3.3.1) and (3.3.3)) yield:

$$\begin{aligned} Q_i &= q_{i1}(\tilde{y}_i - f_0(\mathbf{x}_i)) - q_{i1}(\tilde{y}_i - h(\mathbf{x}_i)) - q_{i1}(y_i - f_0(\mathbf{x}_i)) + q_{i1}(y_i - h(\mathbf{x}_i)) \\ &= q_{i1}(h(\mathbf{x}_i) - f_0(\mathbf{x}_i)) - q_{i1}(y_i - f_0(\mathbf{x}_i)) + q_{i1}(y_i - h(\mathbf{x}_i)) = 0 \quad (\tilde{y}_i = h(\mathbf{x}_i)) \end{aligned}$$

- If  $i \in N_3$ , we again distinguish between two subcases depending on the value of  $y_i$ . If  $y_i = \tilde{y}_i$ , then  $\widehat{R}(f_0, \widetilde{\mathcal{D}}_i) = \widehat{R}(f_0, \mathcal{D}_i)$  and  $\widehat{R}(h, \widetilde{\mathcal{D}}_i) = \widehat{R}(h, \mathcal{D}_i)$  and, hence,  $Q_i = 0$ . If, instead,  $y_i \neq \tilde{y}_i$ , then  $y_i, \tilde{y}_i \geq f_0(\mathbf{x}_i) > h(\mathbf{x}_i)$  and Equations (3.3.1) and (3.3.3) yield

$$\begin{aligned} Q_i &= q_{i1}(\tilde{y}_i - f_0(\mathbf{x}_i)) - q_{i1}(\tilde{y}_i - h(\mathbf{x}_i)) - q_{i1}(y_i - f_0(\mathbf{x}_i)) + q_{i1}(y_i - h(\mathbf{x}_i)) \\ &= -q_{i1}(f_0(\mathbf{x}_i) - h(\mathbf{x}_i)) - q_{i1}(y_i - f_0(\mathbf{x}_i)) + q_{i1}(y_i - h(\mathbf{x}_i)) = 0 \quad (\tilde{y}_i = f_0(\mathbf{x}_i)) \end{aligned}$$

- Finally, if  $i \in N_4$ , then  $y_i, \tilde{y}_i \leq h(\mathbf{x}_i) < f_0(\mathbf{x}_i)$  and Equations (3.3.1)) and (3.3.3) yield

$$\begin{aligned} Q_i &= q_{i2}(f_0(\mathbf{x}_i) - \tilde{y}_i) - q_{i2}(h(\mathbf{x}_i) - \tilde{y}_i) - q_{i2}(f_0(\mathbf{x}_i) - y_i) + q_{i2}(h(\mathbf{x}_i) - y_i) \\ &= q_{i2}(f_0(\mathbf{x}_i) - h(\mathbf{x}_i)) - q_{i2}(f_0(\mathbf{x}_i) - y_i) + q_{i2}(h(\mathbf{x}_i) - y_i) \quad (\tilde{y}_i = h(\mathbf{x}_i)) \end{aligned}$$

■

Next, we show that based on the construction of the datasets and the assumptions we have made so far, for the reports in  $\tilde{\mathcal{D}'}$  function  $f_1$  has smaller error than  $f_0$ .

**Lemma 3.2.**  $f_1 \succ_{\tilde{\mathcal{D}'}} f_0$ .

*Proof.* Given the construction of  $\mathcal{D}'$ , it holds that  $f_1 \succ_{\mathcal{D}'} f_0$  (Equation (3.3.2)), and so by definition it is true that either  $\widehat{R}(f_1, \mathcal{D}') < \widehat{R}(f_0, \mathcal{D}')$ , or  $\widehat{R}(f_1, \mathcal{D}') = \widehat{R}(f_0, \mathcal{D}')$  and  $\|f_1\| < \|f_0\|$ . Hence, in order to prove the lemma, it suffices to prove the inequality

$$\widehat{R}(f_1, \tilde{\mathcal{D}'}) - \widehat{R}(f_0, \tilde{\mathcal{D}'}) \leq \widehat{R}(f_1, \mathcal{D}') - \widehat{R}(f_0, \mathcal{D}'). \quad (3.3.4)$$

or equivalently, that the quantity

$$Q_i = \widehat{R}(f_1, \tilde{\mathcal{D}'}_i) - \widehat{R}(f_0, \tilde{\mathcal{D}'}_i) - \widehat{R}(f_1, \mathcal{D}'_i) + \widehat{R}(f_0, \mathcal{D}'_i) \quad (3.3.5)$$

is non-positive for every agent  $i \in N$ . We distinguish five cases depending on the class of agent  $i$ .

- If  $i \in N_0$ , then  $\widehat{R}(f_1, \tilde{\mathcal{D}'}_i) - \widehat{R}(f_0, \tilde{\mathcal{D}'}_i) = \widehat{R}(f_1, \mathcal{D}'_i) - \widehat{R}(f_0, \mathcal{D}'_i) = 0$  and  $Q_i = 0$ .
- Class  $N_1$  contains no manipulators, so  $y'_i = y_i$  for every  $i \in N_1$ . We distinguish two subcases depending on the value of  $y'_i$ . If  $y'_i = \tilde{y}'_i$ , then  $\widehat{R}(f_1, \tilde{\mathcal{D}'}_i) = \widehat{R}(f_1, \mathcal{D}'_i)$  and  $\widehat{R}(f_0, \tilde{\mathcal{D}'}_i) = \widehat{R}(f_0, \mathcal{D}'_i)$  and, hence,  $Q_i = 0$ . If  $y'_i \neq \tilde{y}'_i$ , then  $y'_i, \tilde{y}'_i \leq f_0(\mathbf{x}_i) < f_1(\mathbf{x}_i)$  and Equations (3.3.1) and (3.3.5) yield

$$\begin{aligned} Q_i &= q_{i2}(f_1(\mathbf{x}_i) - \tilde{y}'_i) - q_{i2}(f_0(\mathbf{x}_i) - \tilde{y}'_i) - q_{i2}(f_1(\mathbf{x}_i) - y'_i) + q_{i2}(f_0(\mathbf{x}_i) - y'_i) \\ &= q_{i2}(f_1(\mathbf{x}_i) - f_0(\mathbf{x}_i)) - q_{i2}(f_1(\mathbf{x}_i) - y'_i) + q_{i2}(f_0(\mathbf{x}_i) - y'_i) = 0 \quad (\tilde{y}'_i = f_0(\mathbf{x}_i)) \end{aligned}$$

- If  $i \in N_2$ , recall that  $\tilde{y}'_i = f_1(\mathbf{x}_i) > f_0(\mathbf{x}_i)$ . We distinguish three subcases depending on the value of  $y'_i$ . If  $y'_i < f_0(\mathbf{x}_i) < f_1(\mathbf{x}_i)$ , Equations (3.3.1) and (3.3.5) yield

$$\begin{aligned} Q_i &= q_{i1}(\tilde{y}'_i - f_1(\mathbf{x}_i)) - q_{i1}(\tilde{y}'_i - f_0(\mathbf{x}_i)) - q_{i2}(f_1(\mathbf{x}_i) - y'_i) + q_{i2}(f_0(\mathbf{x}_i) - y'_i) \\ &= -(q_{i1} + q_{i2})(f_1(\mathbf{x}_i) - f_0(\mathbf{x}_i)) < 0. \end{aligned}$$

If  $f_0(\mathbf{x}_i) \leq y'_i < f_1(\mathbf{x}_i) = \tilde{y}'_i$ , Equations (3.3.1) and (3.3.5) yield

$$\begin{aligned} Q_i &= q_{i2}(f_1(\mathbf{x}_i) - \tilde{y}'_i) - q_{i1}(\tilde{y}'_i - f_0(\mathbf{x}_i)) - q_{i2}(f_1(\mathbf{x}_i) - y'_i) + q_{i1}(y'_i - f_0(\mathbf{x}_i)) \\ &= -(q_{i1} + q_{i2})(\tilde{y}'_i - y'_i) < 0. \end{aligned}$$

Finally, if  $y'_i \geq f_1(\mathbf{x}_i) = \tilde{y}'_i > f_0(\mathbf{x}_i)$ , Equations (3.3.1) and (3.3.5) yield

$$Q_i = q_{i2}(f_1(\mathbf{x}_i) - \tilde{y}'_i) - q_{i1}(\tilde{y}'_i - f_0(\mathbf{x}_i)) - q_{i1}(y'_i - f_1(\mathbf{x}_i)) + q_{i1}(y'_i - f_0(\mathbf{x}_i)) = 0.$$

- Class  $N_3$  contains no manipulators either (hence,  $y'_i = y_i$  for every  $i \in N_3$ ). We distinguish two subcases depending on the value of  $y'_i$ . If  $y'_i = \tilde{y}'_i$ , then  $\widehat{R}(f_1, \tilde{\mathcal{D}}'_i) = \widehat{R}(f_1, \mathcal{D}'_i)$  and  $\widehat{R}(f_0, \tilde{\mathcal{D}}'_i) = \widehat{R}(f_0, \mathcal{D}'_i)$  and, hence,  $Q_i = 0$ . If  $y'_i \neq \tilde{y}'_i$ , then  $y'_i, \tilde{y}'_i \geq f_0(\mathbf{x}_i) > f_1(\mathbf{x}_i)$  and Equations (3.3.1) and (3.3.5) yield

$$\begin{aligned} Q_i &= q_{i1}(\tilde{y}'_i - f_1(\mathbf{x}_i)) - q_{i1}(\tilde{y}'_i - f_0(\mathbf{x}_i)) - q_{i1}(y'_i - f_1(\mathbf{x}_i)) + q_{i1}(y'_i - f_0(\mathbf{x}_i)) \\ &= q_{i1}(f_0(\mathbf{x}_i) - f_1(\mathbf{x}_i)) - q_{i1}(y'_i - f_1(\mathbf{x}_i)) + q_{i1}(y'_i - f_0(\mathbf{x}_i)) = 0 \quad (f_0(\mathbf{x}_i) = \tilde{y}'_i) \end{aligned}$$

- Finally, if  $i \in N_4$ , then  $\tilde{y}'_i = f_1(\mathbf{x}_i) < f_0(\mathbf{x}_i)$ . We distinguish three subcases depending on the value of  $y'_i$ . If  $y'_i > f_0(\mathbf{x}_i) > f_1(\mathbf{x}_i) = \tilde{y}'_i$ , then Equations (3.3.1) and (3.3.5) yield

$$\begin{aligned} Q_i &= q_{i2}(f_1(\mathbf{x}_i) - \tilde{y}'_i) - q_{i2}(f_0(\mathbf{x}_i) - \tilde{y}'_i) - q_{i1}(y'_i - f_1(\mathbf{x}_i)) + q_{i1}(y'_i - f_0(\mathbf{x}_i)) \\ &= -(q_{i1} + q_{i2})(f_0(\mathbf{x}_i) - f_1(\mathbf{x}_i)) < 0. \end{aligned}$$

If  $\tilde{y}'_i = f_1(\mathbf{x}_i) < y'_i \leq f_0(\mathbf{x}_i)$  (i.e.,  $y'_i > \tilde{y}'_i$ ), Equations (3.3.1) and (3.3.5) yield

$$\begin{aligned} Q_i &= q_{i1}(\tilde{y}'_i - f_1(\mathbf{x}_i)) - q_{i2}(f_0(\mathbf{x}_i) - \tilde{y}'_i) - q_{i1}(y'_i - f_1(\mathbf{x}_i)) + q_{i2}(f_0(\mathbf{x}_i) - y'_i) \\ &= -(q_{i1} + q_{i2})(y'_i - f_1(\mathbf{x}_i)) < 0. \end{aligned}$$

Finally, if  $y'_i = \tilde{y}'_i \leq f_1(\mathbf{x}_i) < f_0(\mathbf{x}_i)$ , Equations (3.3.1) and (3.3.5) yield

$$Q_i = q_{i2}(f_1(\mathbf{x}_i) - \tilde{y}'_i) - q_{i2}(f_0(\mathbf{x}_i) - \tilde{y}'_i) - q_{i2}(f_1(\mathbf{x}_i) - y'_i) + q_{i2}(f_0(\mathbf{x}_i) - y'_i) = 0.$$

■

From the definition of the binary relation  $\succ$ . and Lemmas 3.1 and 3.2 we have that:

$$\widehat{R}(f_0, \tilde{\mathcal{D}}) - \widehat{R}(h, \tilde{\mathcal{D}}) \leq 0 \quad (3.3.6)$$

$$\widehat{R}(f_1, \tilde{\mathcal{D}}') - \widehat{R}(f_0, \tilde{\mathcal{D}}') \leq 0. \quad (3.3.7)$$

In fact, at least one of these inequalities is *strict*. To see this, observe that if none of them is strict, then it must be the case that  $\|f_0\| < \|h\|$  and  $\|f_1\| < \|f_0\|$ , due to the definition of  $\succ$ . But if  $\|f_1\| < \|f_0\|$ , then, by the definition of  $h$ :  $\|h\| = \frac{1}{2}\|f_0 + f_1\| \leq \frac{1}{2}(\|f_0\| + \|f_1\|)$ , where the last derivation is due to the triangle inequality. This is a contradiction with the fact that  $\|f_0\| < h$ .

Using the fact that at least one of the inequalities in Equations (3.3.6) and (3.3.7), multiplying both sides of Equation (3.3.7) with  $1/2$ , and summing with Equation (3.3.6) we get that:

$$\widehat{R}(f_0, \tilde{\mathcal{D}}) - \widehat{R}(h, \tilde{\mathcal{D}}) + \frac{1}{2}\widehat{R}(f_1, \tilde{\mathcal{D}}') - \frac{1}{2}\widehat{R}(f_0, \tilde{\mathcal{D}}') < 0 \quad (3.3.8)$$

We show in the next lemma that the inequality in Equation (3.3.8) cannot be true. This is the contradiction for the assumption that there exists a group of manipulators who gain by misreporting.

**Lemma 3.3.**  $\widehat{R}(f_0, \tilde{\mathcal{D}}) - \widehat{R}(h, \tilde{\mathcal{D}}) + \frac{1}{2}\widehat{R}(f_1, \tilde{\mathcal{D}}') - \frac{1}{2}\widehat{R}(f_0, \tilde{\mathcal{D}}') \geq 0$ .

*Proof.* To prove the lemma, it suffices to show that the quantity:

$$Q_i = \widehat{R}(f_0, \tilde{\mathcal{D}}_i) - \widehat{R}(h, \tilde{\mathcal{D}}_i) + \frac{1}{2}\widehat{R}(f_1, \tilde{\mathcal{D}}'_i) - \frac{1}{2}\widehat{R}(f_0, \tilde{\mathcal{D}}'_i) \quad (3.3.9)$$

is non-negative for every  $i \in N$ . Indeed, if  $Q_i \geq 0$ , then by summing across all  $i \in N$  we obtain the desired result. We distinguish five cases based on the class of agent  $i$ :

- If  $i \in N_0$ , then  $\widehat{R}(f_0, \tilde{\mathcal{D}}_i) = \widehat{R}(h, \tilde{\mathcal{D}}_i)$  and  $\widehat{R}(f_1, \tilde{\mathcal{D}}'_i) = \widehat{R}(f_0, \tilde{\mathcal{D}}'_i)$  and, hence,  $Q_i = 0$ .
- If  $i \in N_1$ , recall that  $f_0(\mathbf{x}_i) \leq \tilde{y}_i = \tilde{y}'_i < h(\mathbf{x}_i) < f_1(\mathbf{x}_i)$ . Hence, Equations (3.3.1) and (3.3.5), as well as the fact  $h(\mathbf{x}_i) = (f_0(\mathbf{x}_i) + f_1(\mathbf{x}_i))/2$  yield

$$\begin{aligned} Q_i &= q_{i1}(\tilde{y}_i - f_0(\mathbf{x}_i)) - q_{i2}(h(\mathbf{x}_i) - \tilde{y}_i) + \frac{q_{i2}}{2}(f_1(\mathbf{x}_i) - \tilde{y}'_i) - \frac{q_{i1}}{2}(\tilde{y}'_i - f_0(\mathbf{x}_i)) \\ &= \frac{q_{i1} + q_{i2}}{2}(\tilde{y}_i - f_0(\mathbf{x}_i)) \geq 0. \end{aligned}$$

- If  $i \in N_2$ , recall that  $f_0(\mathbf{x}_i) < \tilde{y}_i < f_1(\mathbf{x}_i) = \tilde{y}'_i$ . Then, Equations (3.3.1) and (3.3.9) and the relationship between functions  $h$ ,  $f_0$ , and  $f_1$  yield

$$\begin{aligned} Q_i &= q_{i1}(\tilde{y}_i - f_0(\mathbf{x}_i)) - q_{i1}(\tilde{y}_i - h(\mathbf{x}_i)) + \frac{q_{i2}}{2}(\tilde{y}'_i - f_1(\mathbf{x}_i)) - \frac{q_{i1}}{2}(\tilde{y}'_i - f_0(\mathbf{x}_i)) \\ &= \frac{q_{i1}}{2}(2 \cdot h(\mathbf{x}_i) - f_0(\mathbf{x}_i) - f_1(\mathbf{x}_i)) = 0. \end{aligned}$$

- If  $i \in N_3$ , recall that  $f_1(\mathbf{x}_i) < h(\mathbf{x}_i) < \tilde{y}_i = \tilde{y}'_i \leq f_0(\mathbf{x}_i)$ . Then, Equations (3.3.1) and (3.3.9) and the relationship between functions  $h$ ,  $f_0$ , and  $f_1$  yield

$$\begin{aligned} Q_i &= q_{i2}(f_0(\mathbf{x}_i) - \tilde{y}_i) - q_{i1}(\tilde{y}_i - h(\mathbf{x}_i)) + \frac{q_{i1}}{2}(\tilde{y}'_i - f_1(\mathbf{x}_i)) - \frac{q_{i2}}{2}(f_0(\mathbf{x}_i) - \tilde{y}'_i) \\ &= \frac{q_{i1} + q_{i2}}{2}(f_0(\mathbf{x}_i) - \tilde{y}_i) \geq 0. \end{aligned}$$

- Finally, if  $i \in N_4$ , recall that  $\tilde{y}'_i = f_1(\mathbf{x}_i) < h(\mathbf{x}_i) = \tilde{y}_i < f_0(\mathbf{x}_i)$ . Then, Equations (3.3.1) and (3.3.9) and the relation between functions  $h$ ,  $f_0$ , and  $f_1$  yield

$$\begin{aligned} Q_i &= q_{i2}(f_0(\mathbf{x}_i) - \tilde{y}_i) - q_{i1}(\tilde{y}_i - h(\mathbf{x}_i)) + \frac{q_{i1}}{2}(\tilde{y}'_i - f_1(\mathbf{x}_i)) - \frac{q_{i2}}{2}(f_0(\mathbf{x}_i) - \tilde{y}'_i) \\ &= \frac{q_{i2}}{2}(f_0(\mathbf{x}_i) + f_1(\mathbf{x}_i) - 2 \cdot h(\mathbf{x}_i)) = 0. \end{aligned}$$

■

This concludes the proof of Theorem 3.1. ■

**Remark 3.1.** Note that the proof regarding the quantile regression being group strategyproof carries over even in the case where the empirical risk function  $\widehat{R}(f, \mathbf{q}, \mathcal{D})$  is adjusted using a convex regularizer  $\rho : \mathcal{F} \rightarrow \mathbb{R}$ , i.e.,

$$\widehat{R}(f, \mathbf{q}, \mathcal{D}) = \sum_{i \in N: y_i \geq f(\mathbf{x}_i)} q_{i1} \cdot |y_i - f(\mathbf{x}_i)| + \sum_{i \in N: y_i < f(\mathbf{x}_i)} q_{i2} \cdot |y_i - f(\mathbf{x}_i)| + \rho(f)$$

To see this, note that the addition of  $\rho(f)$  in the definition of the risk function does not affect Lemmas 3.1 and 3.2 as the regularizer appears on both sides and hence, cancels out. Additionally, Lemma 3.3 carries over to the case with the  $\rho$  regularizer due to the regularizer's convexity.

**Corollary 3.1.** The generalized  $L_1$ -ERM algorithm (i.e., regularized ERM with a weighted  $L_1$  loss) is part of the quantile regression family of mechanisms covered by Theorem 3.1 with convex regularizer and hence, it is group strategyproof.

### 3.3.2 GENERALIZED RESISTANT HYPERPLANE MECHANISMS

In this section, we introduce a novel family of strategyproof mechanisms for linear regression. Our family extends the known family of resistant line mechanisms from the statistics literature [Johnstone and Velleman \(1985\)](#), which were only defined for simple linear regression ( $d = 1$ ), to higher dimensions. We first take a slight detour through a previous approach in the literature.

#### A DETOUR THROUGH CLOCKWISE REPEATED MEDIAN MECHANISMS

[Perote and Perote-Peña \(2004\)](#) introduced a novel family of mechanisms, which they termed *Clockwise Repeated Median* (CRM) mechanisms. CRM mechanisms are only defined for the special case of *simple linear regression*, i.e., for fitting a straight line through a set of points on a plane. In describing these mechanisms, we use scalar notations where possible. For instance, we use  $x_i$  to denote the x-coordinate of agent  $i$ , and  $\beta_1$  to denote the slope of the regression line. For CRM mechanisms to be well defined, we also need to assume that the set of points is “admissible”.

**Definition 3.3** (Admissible Set). A collection of data points  $\mathcal{D} = (x_i, y_i)_{i \in N}$  is called admissible if  $x_i \neq x_j$  for all distinct  $i, j \in N$ .

The CRM family is parametrized by two subsets of agents,  $S, S' \subseteq N$ . These subsets must be chosen based on the public information  $x$ , and therefore can be treated as fixed. Informally, given  $S, S' \subseteq N$ , the  $(S, S')$ -CRM mechanism first computes the median *clockwise angle* (CWA), defined below, from each point  $i \in S$  to points in  $S'$ . Then, it chooses the point  $i^* \in S$  whose median CWA is the median of the median CWAs from all points in  $S$ . If the median CWA from point  $i^*$  is towards point  $j^* \in S'$ , then the mechanism returns the straight line passing through points  $i^*$  and  $j^*$ . Formally, the mechanism is defined as follows. [Perote and Perote-Peña \(2004\)](#) established the equivalence of this formal definition and the aforementioned informal description.

**Definition 3.4** (CRM Mechanisms). *Define the clockwise angle (CWA) from  $(x_i, y_i)$  to  $(x_j, y_j)$  as:*

$$CWA((x_i, y_i), (x_j, y_j)) = \pi + \text{sign}(x_j - x_i) \cdot \frac{\pi}{2} + \text{sign}\left(\frac{y_j - y_i}{x_j - x_i}\right) \left| \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) \right|. \quad (3.3.10)$$

Given  $\mathcal{D} = (x_i, y_i)_{i \in N}$  and  $S, S' \subseteq N$ , let the directing angle be defined as:

$$DA(S, S') = \underset{i \in S}{\text{med}} \underset{j \in S': j \neq i}{\text{med}} CWA((x_i, y_i), (x_j, y_j)). \quad (3.3.11)$$

Then, the  $(S, S')$ -CRM mechanism returns the line  $\beta = (\beta_1, \beta_0)$  given by:

$$\begin{aligned} \beta_1 &= \tan\left[DA(S, S') - \pi - \frac{\pi}{2} \cdot \text{sign}(DA(S, S') - \pi)\right], \\ \beta_0 &= \underset{i \in S}{\text{med}} (y_i - \beta_1 \cdot x_i). \end{aligned} \quad (3.3.12)$$

First, we notice that the definition of the CRM family uses three medians: two to define the directing angle  $DA(S, S')$ , and one to define the  $y$ -intercept  $\beta_0$ . Each median, when taken over an even number of values, can be the left median or the right median. While [Perote and Perote-Peña \(2004\)](#) do not mention how these choices should be made, it is easy to check that in order to achieve the desired incentive properties, these choices cannot be made independently of each other. Later, we present a generalization which captures the different feasible choices in a simpler form.

[Perote and Perote-Peña \(2004\)](#) claimed that the  $(S, S')$ -CRM mechanism is strategyproof when  $S \subseteq S'$  or  $S \cap S' = \emptyset$ , and provided an involved, geometric proof. However, we have identified a mistake in their proof. In fact, we have found two counterexamples, one with  $S \subseteq S'$  and one with

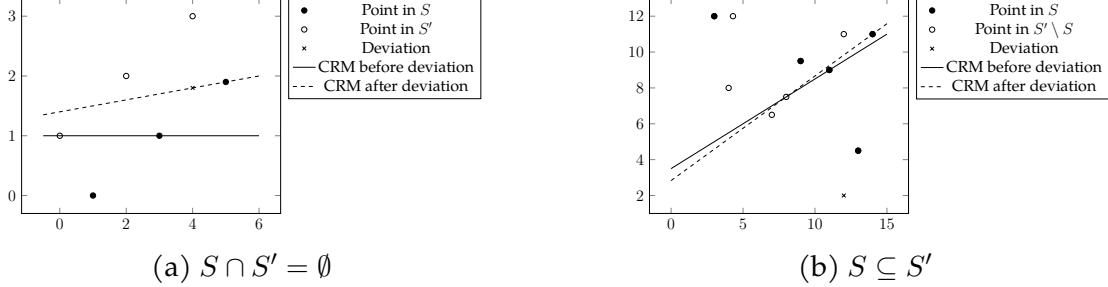


Figure 3.2: Examples showing violation of strategyproofness of  $(S, S')$ -CRM mechanisms. Fig. 3.2a shows a case with  $S \cap S' = \emptyset$ , while Fig. 3.2b shows a case with  $S \subseteq S'$ .

$S \cap S' = \emptyset$ , for which the corresponding  $(S, S')$ -CRM mechanisms violate strategyproofness, thus disproving their claim. These counterexamples are presented in Figure 3.2.

**Example 3.1** ( $S \cap S' = \emptyset$ ). This example is shown in Figure 3.2a. Points in filled dots are in  $S$ , while points in empty dots are in  $S'$ . The coordinates of these points are as follows.

$$S = \{(1, 0), (3, 1), (5, 1.9)\} \quad \& \quad S' = \{(0, 1), (2, 2), (4, 3)\}.$$

Notice that  $S \cap S' = \emptyset$ . Also,  $|S|$  and  $|S'|$  are odd, alleviating the need to choose between left and right medians in the CRM definition. When the agents truthfully report, one can check that CRM returns the line connecting points  $(3, 1)$  from  $S$  and  $(0, 1)$  from  $S'$ . This line is given by the equation  $y = 1$ . Suppose that the agent  $i$  controlling the point at  $x = 4$  misreports  $\tilde{y}_i = 1.8$  instead of  $y_i = 3$ . The new point is depicted with a cross. One can check that this causes the CRM mechanism to switch to the dashed line ( $y = 0.1 \cdot x + 1.4$ ), which makes agent  $i$  strictly better off, and violates strategyproofness.

**Example 3.2** ( $S \subseteq S'$ ). In Figure 3.2b, points in  $S$  (thus also in  $S'$ ) are depicted with filled dots, while points in  $S' \setminus S$  are depicted with empty dots. The coordinates of these points are:

$$S = \{(3, 12), (9, 9.5), (11, 9), (13, 4.5), (14, 11)\} \quad \& \\ S' = S \cup \{(4, 8), (4.3, 12), (7, 6.5), (8, 7.5), (12, 11)\}$$

Notice that  $S \subseteq S'$ . Further,  $|S|$  is odd, and  $|S'|$  is even (thus, for each  $i \in S$ ,  $|S' \setminus \{i\}|$  is odd), once again eliminating the need to choose between the left and the right medians in the CRM definition.

When all points are reported truthfully, one can see that the CRM mechanism chooses the solid line ( $3y =$

$2x + 8$ ). Suppose now that agent  $i$  with point  $(12, 11)$  reports  $\tilde{y}_i = 0$ , instead of  $y_i = 11$ . Then, the CRM mechanism chooses the dashed line, which makes agent  $i$  strictly better off, again violating strategyproofness.

Nevertheless, we have been able to identify a subset of the CRM family, for which we can establish strategyproofness (in fact, group strategyproofness). In particular, we replace  $S \subseteq S'$  with the more restrictive condition  $S = S'$ , and for  $S \cap S' = \emptyset$ , we either add  $|S| = 1$  or  $|S'| = 1$ , or replace it with a stricter condition that we define below.

**Definition 3.5** (Separable Sets of Points in a Plane). *Let  $S, S'$  be two sets of points in  $\mathbb{R}^2$ . We say that  $S$  and  $S'$  are separable if  $\max_{i \in S} x_i < \min_{j \in S'} x_j$  or  $\max_{j \in S'} x_j < \min_{i \in S} x_i$ . In other words, it should be possible to separate them by a vertical line.*

Note that separability of  $S$  and  $S'$  implies  $S \cap S' = \emptyset$ . We now present a corrected version of the result of [Perote and Perote-Peña \(2004\)](#), and claim the stronger guarantee of group strategyproofness. We do not present a proof as we later introduce a much broader family of mechanisms, and prove their group strategyproofness directly.

### Theorem 3.2

Given  $S, S' \subseteq N$ , the  $(S, S')$ -CRM mechanism is group strategyproof if one of the following conditions holds. 1.  $S = S'$ , 2.  $S$  and  $S'$  are separable, and 3.  $S \cap S' = \emptyset$  and  $\min(|S|, |S'|) = 1$ .

The third condition partially resembles dictatorship as the agent in the singleton set is guaranteed to have zero residual (i.e., be on the regression line).

## GENERALIZED RESISTANT LINE MECHANISMS ON A PLANE

In this section, our goal is to introduce a novel family of group strategyproof mechanisms that include, as special cases, the mechanisms covered in the three cases of Theorem 3.2. Our starting point is the family of *resistant line* (RL) mechanisms from the statistics literature [Johnstone and Velleman \(1985\)](#), which [Perote and Perote-Peña \(2004\)](#) showed to be equivalent to the case of separable  $S$  and  $S'$ .

The standard formulation of the RL mechanism involves three sets  $L, M, R \subseteq N$  such that  $\max_{i \in L} x_i < \min_{i \in M} x_i$  and  $\max_{i \in M} x_i < \min_{i \in R} x_i$ , and returns a line  $\beta = (\beta_1, \beta_0)$  given by

$$\text{med}_{i \in L} y_i - \beta_1 \cdot x_i - \beta_0 = \text{med}_{i \in R} y_i - \beta_1 \cdot x_i - \beta_0 = 0.$$

That is, the line makes the median residuals in  $L$  and  $R$  zero. It is known that this equation yields a unique solution [Johnstone and Velleman \(1985\)](#). [Perote and Perote-Peña \(2004\)](#) showed that this is identical to the  $(L, R)$ -CRM mechanism. Indeed, separability of  $L$  and  $R$  makes clockwise angles from points in  $L$  to points in  $R$  monotonic in (and thus replaceable by) slopes, yielding the following formulation for the  $(L, R)$ -CRM mechanism.

$$\beta_1 = \text{med}_{i \in L} \text{med}_{j \in R} \frac{y_j - y_i}{x_j - x_i} \quad \& \quad \beta_0 = \text{med}_{i \in L} y_i - \beta_1 x_i = \text{med}_{j \in R} y_j - \beta_1 x_j.$$

The alternative definition of  $\beta_0 = \text{med}_{j \in R} (y_j - \beta_1 \cdot x_j)$  follows from the fact that if the line passes through  $i^* \in L$ , it is directed towards the point in  $R$  which is at the median angle or slope, and thus bisects  $R$  in addition to bisecting  $L$ .

Along with Theorem 3.2, this observation establishes group strategyproofness of all resistant line mechanisms. Two popular mechanisms from this family are the Brown-Mood mechanism [Brown et al. \(1951\)](#), in which  $L$  and  $R$  each contain half of the points while  $M$  is empty, and the Tukey mechanism [Tukey et al. \(1977\)](#), in which  $L, M$ , and  $R$  each contain a third of the points.

Our next step is to extend this family. A natural idea is that instead of making the *median* residuals from  $S$  and  $S'$  zero, we make the  $k^{\text{th}}$  smallest residual in  $S$  and the  $(k')^{\text{th}}$  smallest residual in  $S'$  zero, for fixed  $k \in [|S|]$  and  $k' \in [|S'|]$ .

**Definition 3.6** (Generalized Resistant Line (GRL) Mechanisms). *Given separable sets  $S, S' \subseteq N$ ,  $k \in [|S|]$ , and  $k' \in [|S'|]$ , the  $(S, S', k, k')$ -generalized resistant line (GRL) mechanism returns the line  $\beta = (\beta_1, \beta_0)$  given by*

$$\min_{i \in S}^k y_i - \beta_1 x_i - \beta_0 = \min_{j \in S'}^{k'} y_j - \beta_1 x_j - \beta_0 = 0. \quad (3.3.13)$$

We show that these mechanisms are well defined (i.e., unique solution to Equation (3.3.13) ex-

ists), and they are group strategyproof. Once again, we omit the proof because we later introduce an even broader family of mechanisms, for which we prove these results directly.

**Theorem 3.3: Strategyproofness of GRL**

For separable sets  $S, S' \subseteq N$ ,  $k \in [|S|]$  and  $k' \in [|S'|]$ , the  $(S, S', k, k')$ -generalized resistant line mechanism is well defined and group strategyproof.

While it is clear that generalized resistant line mechanisms cover the second case of Theorem 3.2 (i.e., separable  $S$  and  $S'$ ), we surprisingly find that they also cover the first case ( $S = S'$ ) and the third case ( $S \cap S' = \emptyset$  and  $\min(|S|, |S'|) = 1$ ). That is, Theorem 3.3 strictly generalizes Theorem 3.2.

**Lemma 3.4.** *The  $(S, S')$ -CRM mechanism is a generalized resistant line mechanism when 1.  $S = S'$ , 2.  $S$  and  $S'$  are separable, or 3.  $S \cap S' = \emptyset$  and  $\min(|S|, |S'|) = 1$ .*

*Proof.* We refer to the point in  $S$  which has the median of all median CWAs (i.e., DA) as the directing point, and the point in  $S'$  to which this DA is pointing as the directed point.

First, we show that for any  $S \subseteq N$ , the  $(S, S)$ -CRM mechanism is  $(L, R, k, k')$ -GRL mechanism for some  $L, R, k, k'$ . Without loss of generality, we can assume  $S = N$  as the other points are simply ignored. Thus, we will refer to the  $(N, N)$ -CRM mechanism.

First, consider the case where  $n$  is even. Let  $L$  (resp.  $R$ ) be the set of  $n/2$  points with the smallest (resp. largest)  $x$  coordinates. We show equivalence of the  $(N, N)$ -CRM mechanism to the  $(L, R, k, k')$ -GRL mechanism for appropriate  $k$  and  $k'$ . Let  $(\beta_1, \beta_0)$  be the line returned by the CRM mechanism.

Choose  $x^* \in (\max_{i \in L} x_i, \min_{i \in R} x_i)$ , and define the following sets.

- $A = \{i : x_i < x^*, y_i \geq \beta_1 x_i + \beta_0\}$
- $B = \{i : x_i > x^*, y_i > \beta_1 x_i + \beta_0\}$
- $C = \{i : x_i < x^*, y_i < \beta_1 x_i + \beta_0\}$
- $D = \{i : x_i > x^*, y_i \leq \beta_1 x_i + \beta_0\}$

Note that  $A \cup C = L$  and  $B \cup D = R$ . For  $i \in N$ , let  $MCWA_i$  denote the median CWA from  $i$  to points in  $N \setminus \{i\}$ . Note that for each  $i \in L$ , there are strictly more points in  $N \setminus \{i\}$  to the right of it, than to the left of it, implying that  $MCWA_i \in [\pi, 2\pi]$ . Similarly, for each  $i \in R$ , we have  $MCWA_i \in [0, \pi]$ .

Let  $DA$  be the directing angle under the CRM mechanism. Then,  $DA = \min_{i \in L} MCWA_i$  or  $DA = \max_{i \in R} MCWA_i$  based on whether the outer median in the directing angle definition uses the right median or the left median. Assume it uses the left median, so  $DA = \max_{i \in R} MCWA_i$ . The proof for the other case is symmetric.

We now show that in this case,  $B = C = \emptyset$ . This would imply that the mechanism is equivalent to  $(L, R, |L|, 1)$ -GRL because every point in  $L$  has a non-positive residual while every point in  $R$  has a non-negative residual.

For contradiction, let  $B \neq \emptyset$  and take a point  $i_B \in B$ . Note that  $MCWA_{i_B} \leq \max_{i \in R} MCWA_i = DA$ . The directing point  $i^*$  is on the regression line, and hence  $i^* \in D$ . Then, one can check that if  $x_{i_B} < x_{i^*}$ , then  $x_{i_B}$  has strictly less number of points to which its angle is less than  $MCWA_{i_B}$  than  $x_{i^*}$  has to which its angle is less than  $MCWA_{i^*} = DA$ . In the case  $x_{i_B} > x_{i^*}$ , the same happens but for points with angle greater than MCWA. This is a contradiction because each point has exactly  $(n - 2)/2$  points with angle more or less than its MCWA. Hence,  $B = \emptyset$ . Using a symmetric argument, we can establish  $C = \emptyset$ , which completes the proof.

We now consider the case where  $n$  is odd. In this case, let  $L$  (resp.  $R$ ) be the set of  $(n - 1)/2$  points with the smallest (resp. largest)  $x$ -coordinate, and let  $i^*$  be the point with the median  $x$ -coordinate. Once again, we have that  $MCWA_i \in [\pi, 2\pi]$  for each  $i \in L$ , and  $MCWA_i \in [0, \pi]$  for each  $i \in R$ . We add  $i^*$  to  $L$  if  $MCWA_{i^*} \in [\pi, 2\pi]$ , and to  $R$  otherwise. Suppose we add it to  $R$ , and let  $R' = R \cup \{i^*\}$ . Then using an argument similar to above, we can check that the CRM mechanism is equivalent to  $(L, R', k, k')$  for appropriate  $k, k'$ .

The case where  $S \cap S' = \emptyset$  and  $\min(|S|, |S'|) = 1$  is much simpler. Again, without loss of generality, we can consider  $S \cup S' = N$ , and for simplicity, we consider the case where  $n$  is even and  $|S| = 1$ . The other cases are similar. Let  $S = \{i^*\}$ . Without loss of generality, suppose there are more points to the right of  $i^*$  than to the left of it. Let  $R$  be the set of points to the right of  $i^*$ , and  $L$  be the set of points to the left of  $i^*$ . Then, it is easy to see that when we take the median CWA

from  $i^*$  (say, the left median, i.e., the  $(n/2 - 1)^{\text{th}}$  smallest CWA), it will always be towards a point in  $R$ . Moreover, it will be the  $(n/2 - 1 - |S|)^{\text{th}}$  smallest CWA towards points in  $R$ . However, CWAs towards points in  $R$  are monotonic in slopes to points in  $R$ . Hence, the regression line will make the  $(n/2 - 1 - |S|)^{\text{th}}$  smallest residual in  $R$  zero. In other words, the mechanism is equivalent to  $(\{i^*\}, R, 1, n/2 - 1 - |S|)$ -GRL. ■

## GENERALIZED RESISTANT HYPERPLANE MECHANISMS IN HIGH DIMENSIONS

The statistics literature does not offer an extension of resistant line mechanisms to higher dimensions. In our efforts to do so, we quickly realized that this is a non-trivial task. In two dimensions, a generalized resistant line mechanism takes two subsets of data points separable by a vertical line, and returns the regression line which makes prescribed percentiles of residuals in each set zero. In  $d + 1$  dimensions (recall that  $x_i \in \mathbb{R}^d$  and  $y_i \in \mathbb{R}$ ), it seems natural to take  $d + 1$  “separable” subsets of data points, and return the regression hyperplane which makes prescribed percentiles of residuals in each set zero. However, the separability condition must now ensure existence of a unique hyperplane with this property, even if we ignore our game-theoretic desiderata.

In resolving this issue, we make a connection to the literature on the *Ham Sandwich Theorem* and its generalizations. Hereinafter, given a hyperplane  $H$ , we denote by  $H^+$  and  $H^-$  its positive and negative closed half-spaces, respectively. A basic version of the ham sandwich theorem due to [Stone and Tukey \(1942\)](#) states that given  $k$  continuous measures  $\mu_1, \dots, \mu_k$  on  $\mathbb{R}^k$ , there exists a hyperplane  $H$  such that  $\mu_i(H^+) = 1/2$  for each  $i \in [k]$ . A discrete version of this result due to [Elton and Hill \(2011\)](#) states that given  $k$  finite sets  $S_1, \dots, S_k \subseteq \mathbb{R}^k$ , there exists a hyperplane  $H$  such that for each  $i \in [k]$ ,  $H$  “bisects”  $S_i$  and  $H \cap S_i \neq \emptyset$ . Here, we say that a hyperplane  $H$  bisects a set of points  $S$  if each *closed* half-space of  $H$  contains at least  $\lceil |S|/2 \rceil$  points.

For linear regression, this implies that given  $S_1, \dots, S_{d+1} \subseteq \mathcal{D}$ , there exists a “resistant hyperplane” which makes the median residual from  $S_t$  zero, for each  $t \in [d + 1]$ . While this seems like a natural generalization of resistant line mechanisms, it is easy to check that such a hyperplane is not always unique, even in two dimensions. Further, if the median is replaced by other percentiles, the existence is no longer guaranteed.<sup>†</sup>

---

<sup>†</sup>Recall that even in 2d, we needed an extra condition on  $S$  and  $S'$ : separability by a vertical line.

Steiger and Zhao (2010) provide a generalization that *almost* perfectly fits our needs. They show that under certain conditions on  $S_1, \dots, S_{d+1}$ , there exists a unique hyperplane  $H$  which contains a given number of points from each set in its negative closed half-space. This discrete result builds upon previous continuous variants Bárány et al. (2008), Breuer (2010). We first define a condition they require, which also plays a key role in our result.

**Definition 3.7** (Well Separable Sets Kermér and Németh (1973)). *Given  $t \in [k+1]$ , finite sets  $S_1, \dots, S_t$  of points in  $\mathbb{R}^k$  are called well separable if for all disjoint  $I, J \subseteq [t]$ , there exists a hyperplane  $H$  such that  $S_i \subset H^+ \setminus H$  for each  $i \in I$  and  $S_j \subset H^- \setminus H$  for each  $j \in J$ , i.e.,  $H$  separates  $\cup_{i \in I} S_i$  from  $\cup_{j \in J} S_j$  by putting them in different open half-spaces.*

Well separable sets are sometimes called *affinely independent* sets Breuer (2010). Well separability is equivalent to various other conditions Breuer (2010), Steiger and Zhao (2010). In what follows,  $\text{conv}(\cdot)$  denotes the convex hull.

**Proposition 3.1.** *For  $t \in [k+1]$ , finite sets  $S_1, \dots, S_t \subset \mathbb{R}^k$  are well separable if and only if:*

1. *For all choices of  $(x_i \in \text{conv}(S_i))_{i \in [t]}$ , the affine hull of  $x_1, \dots, x_t$  is a  $(t-1)$ -dimensional flat.*
2. *No  $(t-2)$ -dimensional flat has a nonempty intersection with  $\text{conv}(S_i)$  for each  $i \in [t]$ .*
3.  *$\text{conv}(S_1), \dots, \text{conv}(S_t)$  are well separable.*

Steiger and Zhao (2010) impose an additional condition, which we eliminate in our work.

**Definition 3.8** (Weak General Position). *Finite sets  $S_1, \dots, S_k \subset \mathbb{R}^k$  are said to have weak general position if for every choice of  $(x_i \in S_i)_{i \in [k]}$ , the affine hull of  $x_1, \dots, x_k$  is a  $(k-1)$ -dimensional flat which contains no other point of  $\cup_{i \in [k]} S_i$ .*

#### Theorem 3.4: Steiger and Zhao (2010)

If finite sets  $S_1, \dots, S_k \subset \mathbb{R}^k$  are well separable and have weak general position, then given any choice of  $k_i \in [|S_i|]$  for  $i \in [k]$ , there exists a unique hyperplane  $H$  such that for each  $i \in [k]$ ,  $H \cap S_i \neq \emptyset$  and  $|H^- \cap S_i| = k_i$ .

This result gives us *almost* what we want for linear regression in  $\mathbb{R}^{d+1}$ . Given a family of sets  $S_1, \dots, S_{d+1} \subseteq \mathcal{D}$  that are well separable and have weak general position, and  $k_t \in [|S_t|]$  for  $t \in [d+1]$ , it ensures the existence of a unique hyperplane which makes the  $k_t^{\text{th}}$  smallest residual in each set  $S_t$  zero. However, it falls short of our requirements in two key aspects.

- Theorem 3.4 allows the assignment of points in  $\mathcal{D}$  to sets  $S_1, \dots, S_{d+1}$  to depend on the private information  $y$ . For strategyproofness, we need this assignment to be based solely on the public information  $x$ . Recall that in two dimensions, we required sets  $S$  and  $S'$  to be separable by a *vertical* line. We choose the  $d+1$  sets so that they are well separable in the  $d$ -dimensional public information space,<sup>‡</sup> and establish group strategyproofness using a technical lemma, which may be of independent interest.
- While we only want to make the  $k_t^{\text{th}}$  smallest residual in each  $S_t$  zero, [Steiger and Zhao \(2010\)](#) aim for something stronger: they want the number of points from each  $S_t$  in the negative closed halfspace to be exactly  $k_t$ . This necessitates their weak general position assumption, which we relax.

We are now ready to present our results. They closely mirror, but do not make use of, the results of [Steiger and Zhao \(2010\)](#). We revert to using notation of our linear regression setting. Recall that a hyperplane  $\beta = (\beta_1, \beta_0)$  passes through  $(x_i, \beta^\top \bar{x}_i)$  for each  $i \in N$ , where  $\bar{x}_i = (x_i, 1)$ .

**Definition 3.9.** *Given a family  $\mathcal{S} = (S_1, \dots, S_k)$  of nonempty, pairwise disjoint subsets of  $N$ , and a set of points  $P = (p_i)_{i \in N}$ , define the partition function  $\mathcal{P}(P, \mathcal{S}) = (P_t)_{t \in [k]}$ , where  $P_t = (p_i)_{i \in S_t}$  for each  $t \in [k]$ . That is,  $\mathcal{P}(P, \mathcal{S})$  partitions the set of points  $P$  based on index sets from  $\mathcal{S}$ .*

**Definition 3.10** (Publicly Separable Sets of Agents). *We say that a family  $\mathcal{S} = (S_1, \dots, S_{d+1})$  of nonempty, pairwise disjoint subsets of  $N$  is publicly separable if  $\mathcal{P}(x, \mathcal{S})$  is well separable.*

**Definition 3.11** (Generalized Resistant Hyperplane). *Given a family  $\mathcal{S} = (S_1, \dots, S_{d+1})$  of publicly separable sets of agents, and  $\mathbf{k} = (k_1, \dots, k_{d+1})$  with  $k_t \in [|S_t|]$  for  $t \in [d+1]$ , the  $(\mathcal{S}, \mathbf{k})$ -generalized resistant hyperplane (GRH) mechanism returns a hyperplane  $\beta$  such that  $\min_{i \in S_t}^{k_t} (r_i \triangleq y_i - \beta^\top \bar{x}_i) = 0$  for each  $t \in [d+1]$ . That is, it makes the  $k_t^{\text{th}}$  smallest residual from every set  $S_t \in \mathcal{S}$  zero.*

---

<sup>‡</sup>While Theorem 3.4 uses  $d+1$  well separable sets in  $\mathbb{R}^{d+1}$ , even  $\mathbb{R}^d$  allows up to  $d+1$  well separable sets.

We first need to establish that the GRH mechanisms are well defined, i.e., the hyperplane they seek is guaranteed to exist and be unique. To that end, we prove a useful technical lemma, which may be of independent interest.

**Lemma 3.5** (Hyperplane Comparison Lemma). *Given a family  $\mathcal{S} = (S_1, \dots, S_{d+1})$  of publicly separable sets of agents, and two distinct hyperplanes  $\beta^1$  and  $\beta^2$  in  $\mathbb{R}^{d+1}$ , there exists a set  $S_t \in \mathcal{S}$  such that either  $(\beta^1)^\top \bar{x}_i < (\beta^2)^\top \bar{x}_i$  for all  $i \in S_t$ , or  $(\beta^1)^\top \bar{x}_i > (\beta^2)^\top \bar{x}_i$  for all  $i \in S_t$ .*

*Proof.* Consider the intersection of the two hyperplanes in  $\mathbb{R}^{d+1}$ , and let  $W$  be its projection on  $\mathbb{R}^d$  (the public information space). Note that  $W$  is a  $(d-1)$ -dimensional hyperplane in  $\mathbb{R}^d$ . Given an open half-space of  $W$  (say  $W^+$ ), let  $Z$  be the set of points  $\mathbb{R}^{d+1}$  whose projection on  $\mathbb{R}^d$  lies in  $W^+$ . Then, either  $(\beta^1)^\top \bar{p} > (\beta^2)^\top \bar{p}$  for all  $p \in Z$ , or  $(\beta^1)^\top \bar{p} < (\beta^2)^\top \bar{p}$  for all  $p \in Z$ , where  $\bar{p} = (p, 1)$ . Let  $\mathcal{P}(x, \mathcal{S}) = (X_1, \dots, X_{d+1})$ . Because  $\mathcal{S}$  is publicly separable,  $X_1, \dots, X_{d+1}$  are well separable. By Proposition 3.1, no  $(d-1)$ -dimensional flat has a nonempty intersection with  $\text{conv}(X_t)$  for each  $t \in [d+1]$ . Because  $W$  is a  $(d-1)$ -dimensional flat, there exists  $t \in [d+1]$  such that  $W$  does not intersect  $\text{conv}(X_t)$ , i.e.,  $X_t$  lies entirely in an open half-space of  $W$ . Using the previous argument, either  $(\beta^1)^\top \bar{x}_i < (\beta^2)^\top \bar{x}_i$  for all  $i \in S_t$ , or  $(\beta^1)^\top \bar{x}_i > (\beta^2)^\top \bar{x}_i$  for all  $i \in S_t$ . ■

**Proposition 3.2.** *Generalized resistant hyperplane mechanisms are well defined. That is, given a family  $\mathcal{S} = (S_1, \dots, S_{d+1})$  of publicly separable sets of agents, and  $k = (k_1, \dots, k_{d+1})$  with  $k_t \in [|S_t|]$  for  $t \in [d+1]$ , there exists a unique hyperplane  $\beta$  for which  $\min_{i \in S_t} y_i - \beta^\top \bar{x}_i = 0$  for each  $t \in [d+1]$ .*

*Proof.* First, we show that if such a hyperplane exists, it must be unique. Suppose for contradiction that there are two distinct hyperplanes  $\beta^1$  and  $\beta^2$  which make the  $k_t^{\text{th}}$  smallest residual from every  $S_t \in \mathcal{S}$  zero. By the hyperplane comparison lemma (Lemma 3.5), there exists  $S_t \in \mathcal{S}$  such that either  $(\beta^1)^\top \bar{x}_i < (\beta^2)^\top \bar{x}_i$  for all  $i \in S_t$ , or  $(\beta^1)^\top \bar{x}_i > (\beta^2)^\top \bar{x}_i$  for all  $i \in S_t$ . Without loss of generality, suppose it is the former. Then, at least  $k_t$  points in  $S_t$  which have a non-positive residual under  $\beta^2$  have a negative residual under  $\beta^1$ , contradicting the fact that  $\beta^1$  makes the  $k_t^{\text{th}}$  smallest residual from  $S_t$  zero.

For proving existence, we use a counting technique. Create two bipartite graphs  $G = (V \cup W, E)$  and  $G' = (V' \cup W, E')$ . Let  $V$  (resp.  $V'$ ) contain a vertex  $v_k$  (resp.  $v'_k$ ) corresponding to each  $k = (k_1, \dots, k_{d+1})$  such that  $k_t \in [|S_t|]$  for each  $t \in [d+1]$ . Thus,  $|V| = |V'| = \prod_{t=1}^{d+1} |S_t|$ . Let

$W$  contain a vertex  $w_\beta$  corresponding to every *traversal* hyperplane  $\beta$ , i.e., every hyperplane that passes through at least one point from each set  $S_t \in \mathcal{S}$ .

In graph  $G$ , we draw an edge between  $v_k$  and  $w_\beta$  if  $\beta$  makes the  $k_t^{\text{th}}$  smallest residual zero in each  $S_t \in \mathcal{S}$ . For constructing graph  $G'$ , we fix an arbitrary ordering of points in each set, so that we can write  $S_t = \{i_1^t, \dots, i_{|S_t|}^t\}$ . Then, we draw an edge in  $G'$  between  $v'_k$  and  $w_\beta$  if  $\beta$  passes through point  $i_{k_t}^t$  for each  $t \in [d+1]$ .

Our goal is to show that each vertex  $v_k \in V$  has exactly one incident edge in graph  $G$ . We prove this through a sequence of claims. First, we argue that each vertex  $v'_k \in V'$  has exactly one incident edge in graph  $G'$ . The fact that it has *at least* one incident edge follows from the fact that any set of  $d+1$  points in  $\mathbb{R}^{d+1}$  (in particular,  $T = \{i_{k_t}^t\}_{t \in [d+1]}$ ) lie on a hyperplane. If  $v'_k$  has two or more incident edges, then there exist two distinct hyperplanes  $\beta^1$  and  $\beta^2$  which pass through all points in  $T$ . Then, their intersection  $\beta^*$ , which is a  $(d-1)$ -dimensional flat in  $\mathbb{R}^{d+1}$ , must also pass through all points in  $T$ . Let  $\mathcal{P}(x, \mathcal{S}) = (X_1, \dots, X_{d+1})$ . Then, the projection of  $\beta^*$  on the public information space  $\mathbb{R}^d$  is a  $(d-1)$ -dimensional hyperplane in  $\mathbb{R}^d$  which intersects each  $X_t$  (and thus each  $\text{conv}(X_t)$ ). However,  $\mathcal{S}$  is a publicly separable family, i.e.,  $X_1, \dots, X_{d+1}$  are well separable in  $\mathbb{R}^d$ . This violates the first condition of Proposition 3.1.

Since each vertex in  $V'$  has exactly one incident edge, we have  $|E'| = |V'| = \prod_{t=1}^{d+1} |S_t|$ . We next argue that  $|E| = |E'|$ . Take a vertex  $w_\beta \in W$ . Note that if hyperplane  $\beta$  passes through  $a_t$  points from each  $S_t \in \mathcal{S}$ , then it has degree  $\prod_{t=1}^{d+1} a_t$  in both  $G$  and  $G'$ . Since each vertex in  $W$  has the same degree in both graphs, we have  $|E| = |E'| = |V'| = |V|$ .

Finally, we already established that if there is a hyperplane which makes the  $k_t^{\text{th}}$  smallest residual in each  $S_t$  zero, then it must be unique. Thus, each vertex in  $V$  has *at most* one incident edge in  $G$ . Together with  $|E| = |V|$ , this implies that each vertex in  $V$  has *exactly* one incident edge in  $G$ . ■

We are now ready to present our main contribution.

### Theorem 3.5: Strategyproofness of GRH

Every generalized resistant hyperplane mechanism is group strategyproof.

*Proof.* Consider an  $(\mathcal{S}, k)$ -generalized resistant hyperplane mechanism. Consider a set of data points  $\mathcal{D} = (x_i, y_i)_{i \in N}$ . Suppose a coalition  $S \subseteq N$  of agents changes their report to  $(\tilde{y}_i)_{i \in S}$ , and

changes the resulting hyperplane from  $\beta$  to  $\tilde{\beta}$ . Set  $\tilde{y}_i = y_i$  for  $i \in N \setminus S$ , and let  $\tilde{\mathcal{D}} = (\mathbf{x}_i, \tilde{y}_i)_{i \in N}$ .

By the hyperplane comparison lemma (Lemma 3.5), there exists  $S_t \in \mathcal{S}$  such that either  $\beta^\top \bar{\mathbf{x}}_i < \tilde{\beta}^\top \bar{\mathbf{x}}_i$  for all  $i \in S_t$ , or  $\beta^\top \bar{\mathbf{x}}_i > \tilde{\beta}^\top \bar{\mathbf{x}}_i$  for all  $i \in S_t$ .

Without loss of generality, suppose it is the former. The  $k_t^{\text{th}}$  smallest residual from  $S_t$  is zero under  $\beta$  in  $\mathcal{D}$ , and under  $\tilde{\beta}$  in  $\tilde{\mathcal{D}}$ . If  $S \cap S_t = \emptyset$ , or if every manipulator in  $S \cap S_t$  has a positive residual under  $\beta$  in  $\mathcal{D}$ , then at least  $k_t$  non-manipulators in  $N \setminus S$  have a non-positive residual under  $\beta$  in  $\mathcal{D}$ , and thus a strictly negative residual under  $\tilde{\beta}$  in  $\tilde{\mathcal{D}}$ , which contradicts the fact that  $\tilde{\beta}$  makes the  $k_t^{\text{th}}$  smallest residual in  $S_t$  zero in  $\tilde{\mathcal{D}}$ .

In other words, there must exist a manipulator  $i \in S \cap S_t$  who has a non-positive residual under  $\beta$  in  $\mathcal{D}$ . Thus,  $\tilde{\beta}^\top \bar{\mathbf{x}}_i > \beta^\top \bar{\mathbf{x}}_i \geq y_i$ , implying that the manipulator is strictly worse off after the manipulation. Hence, the mechanism is group strategyproof. ■

For two dimensions ( $d = 1$ ), we already argued that our sub-family of group strategyproof CRM mechanisms given by Theorem 3.2 is part of the larger family of GRL mechanisms (Lemma 3.4). It is easy to see that GRL mechanisms are precisely GRH mechanisms in two dimensions. Indeed, GRH mechanisms would require two subsets of agents  $S_1, S_2$  that are publicly separable, i.e., well separable on the  $x$ -axis. Note that this coincides with the separability definition used by GRL mechanisms (Definition 3.5). Hence, the  $(S, S', k, k')$ -GRL mechanism is precisely the  $(\mathcal{S}, \mathbf{k})$ -GRH mechanism with  $\mathcal{S} = (S, S')$  and  $\mathbf{k} = (k, k')$ . In three or more dimensions, we do not know if, given  $\mathbf{x}$ , one can always construct a family  $\mathcal{S}$  of publicly separable sets of agents such that each set  $S_t \in \mathcal{S}$  contains at least a constant fraction of the agents.

### 3.3.3 STRATEGYPROOFNESS VS GROUP STRATEGYPROOFNESS

In the single dimensional setting ( $d = 0$ ), Moulin (1980) proved that all strategyproof mechanisms are also group strategyproof. This alternatively follows from a result by Barberà et al. (2010), who gave a sufficient condition on the underlying domain for the sets of strategyproof and group strategyproof mechanisms to coincide.

Interestingly, all known strategyproof mechanisms for the multidimensional linear regression setting (including generalized  $L_1$ -ERM and generalized resistant hyperplane mechanisms) are

group strategyproof as well. However, it is easy to check that the linear regression setting does not satisfy the sufficient condition of Barberà et al. (2010). Is it still true that all strategyproof mechanisms for linear regression are also group strategyproof? We answer this question *negatively*.

**Example 3.3.** Consider the simple linear regression setting ( $d = 1$ ) with  $n = 2$  agents. Fix the public information  $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ , and consider the mechanism  $M$  that, on input  $\mathbf{y} = (y_1, y_2)$ , returns the line passing through points  $(x_1, y_2)$  and  $(x_2, y_1)$ . Under this mechanism, the outcome for each agent is independent of the agent's report: indeed, the outcome for agent 1 (resp. agent 2) is  $\hat{y}_1 = y_2$  (resp.  $\hat{y}_2 = y_1$ ). Hence, the mechanism is clearly strategyproof. However, group strategyproofness is violated because when  $y_1 \neq y_2$ , the two agents can collude, and report  $\tilde{\mathbf{y}} = (y_2, y_1)$ . This makes the resulting line pass through both agents, making both strictly better off.

The requirement that the outcome for each agent be independent of the agent's report, called *impartiality* in mechanism design, is stricter than (i.e., logically implies) strategyproofness, and has been studied for aggregating opinions or dividing rewards (De Clippel et al., 2008, Holzman and Moulin, 2013, Tamura and Ohseto, 2014, Fischer and Klimm, 2015, Kurokawa et al., 2015).

**Definition 3.12** (Impartial Mechanisms). A mechanism  $M$  is called impartial if the outcome for each agent is independent of the agent's report. Formally, for every agent  $i \in N$ , reports  $\mathbf{y}$ , and alternative report  $\mathbf{y}'_i$  by agent  $i$ , we require that  $\hat{y}_i(M(\mathbf{y})) = \hat{y}_i(M(\mathbf{y}'_i, \mathbf{y}_{-i}))$ .

In linear regression, when the number of agents is  $n = d + 1$ , we can easily characterize all impartial mechanisms because we can set  $\hat{y}_i$  to be an arbitrary function of  $\mathbf{y}_{-i}$ , and return a hyperplane passing through the resulting  $d + 1$  points  $(\mathbf{x}_i, \hat{y}_i)_{i \in N}$ .

**Proposition 3.3.** For  $n = d + 1$ , mechanism  $M$  is impartial if and only if there exist functions  $f_1, \dots, f_n : \mathbb{R}^{n-1} \rightarrow \mathbb{R}$  such that given  $\mathbf{y}$ ,  $M$  returns a hyperplane passing through  $(\mathbf{x}_i, f_i(\mathbf{y}_{-i}))_{i \in N}$ .

Note that functions  $f_i$  can even be discontinuous, which can make the regression hyperplane discontinuous in the input  $\mathbf{y}$ . However, we later show (Theorem 3.7) that under any strategyproof mechanism, the outcome  $\hat{y}_i$  for agent  $i$  must be a continuous function of  $y_i$  (it is a constant function of  $y_i$  in case of impartial mechanisms).

With  $n > d + 1$  points, the question of whether impartial mechanisms even exist is non-trivial. While we still need to set each  $\hat{y}_i$  as a function of  $\mathbf{y}_{-i}$ , it cannot be done arbitrarily as the resulting points  $(\mathbf{x}_i, \hat{y}_i)_{i \in N}$  may no longer lie on a hyperplane. In other words, setting  $\hat{y}_i$  as a function of  $\mathbf{y}_{-i}$  for  $d + 1$  agents already determines the hyperplane, and thus  $\hat{y}_j$  for all remaining agents  $j$ . The mechanism must ensure that these  $\hat{y}_j$  are also independent of  $y_j$ . At first glance, this may seem impossible, except in the trivial case where a constant hyperplane is returned regardless of  $\mathbf{y}$ .

Nonetheless, we show that there exists a wide family of non-trivial impartial mechanisms for linear regression. Our family provides a full characterization of impartial mechanisms for  $d = 1$  (i.e., for simple linear regression). In the result below, we use the notation  $\langle \mathbf{a}, \mathbf{b} \rangle$  instead of  $\mathbf{a}^\top \mathbf{b}$  for the sake of simplicity.

### Theorem 3.6: Impartial Mechanisms

Given  $\mathbf{x}$ , mechanism  $M^{\mathbf{x}}$  for linear regression is impartial if there exist functions  $\{g_i^{\mathbf{x}} : \mathbb{R} \rightarrow \mathbb{R}^d\}_{i \in N}$  and constant  $c^{\mathbf{x}} \in \mathbb{R}$  such that for all  $\mathbf{y}$ , we have  $M^{\mathbf{x}}(\mathbf{y}) = \boldsymbol{\beta} = (\beta_1, \beta_0)$ , where

$$\beta_1 = \sum_{i \in N} g_i^{\mathbf{x}}(y_i), \quad \beta_0 = c^{\mathbf{x}} - \sum_{i \in N} \langle g_i^{\mathbf{x}}(y_i), \mathbf{x}_i \rangle. \quad (3.3.14)$$

For  $d = 1$  and an admissible set of points, this characterizes all impartial mechanisms.

Before presenting the proof of Theorem 3.6, we first present a useful definition.

**Definition 3.13** (Completely Additively Separable). *Function  $f : \mathbb{R}^k \rightarrow \mathbb{R}$  is called completely additively separable if there exist functions  $\{g_i\}_{i=1}^k$  such that  $f(t_1, \dots, t_k) = \sum_{i=1}^k g_i(t_i)$  for all  $\mathbf{t} = (t_1, \dots, t_k) \in \mathbb{R}^k$ .*

It is well known that  $f$  is completely additively separable if and only if for all  $\mathbf{t} \in \mathbb{R}^k$ ,  $i \in [k]$ , and  $t'_i \in \mathbb{R}$ ,  $f(t_i, \mathbf{t}_{-i}) - f(t'_i, \mathbf{t}_{-i})$  is independent of  $\mathbf{t}_{-i}$ .

*Proof of Theorem 3.6.* We omit  $\mathbf{x}$  from all superscripts for simplicity. Suppose mechanism  $M$  is given by Equation (3.3.14). Then:

$$\begin{aligned} \hat{y}_i(\boldsymbol{\beta}) &= \langle \boldsymbol{\beta}, \bar{\mathbf{x}}_i \rangle = \langle \sum_{j \in N} g_j(y_j), \mathbf{x}_i \rangle + c - \sum_{j \in N} \langle g_j(y_j), \mathbf{x}_j \rangle \\ &= c + \sum_{j \in N \setminus \{i\}} \langle g_j(y_j), \mathbf{x}_i - \mathbf{x}_j \rangle. \end{aligned}$$

Note that  $\hat{y}_i(\beta)$  is independent of  $y_i$ , which implies that  $M$  is impartial.

We now prove the converse for simple linear regression ( $d = 1$ ) with an admissible set of points. Suppose mechanism  $M$  is impartial. Given  $\mathbf{y}$ , let  $\beta_1(\mathbf{y})$  be the slope of the line returned by  $M$ , and  $f_i(\mathbf{y}) = \hat{y}_i(M(\mathbf{y}))$  be the outcome for agent  $i$ . Because  $M$  is impartial,  $f_i$  is independent of  $y_i$ . Hence, we denote the outcome for agent  $i$  by  $f_i(\mathbf{y}_{-i})$ .

We want to show that  $h$  is completely additively separable. Equivalently, for every  $\mathbf{y}$  and  $\tilde{\mathbf{y}}$  such that  $\mathbf{y}_{-i} = \tilde{\mathbf{y}}_{-i}$ , we want to show that  $\beta_1(\mathbf{y}) - \beta_1(\tilde{\mathbf{y}})$  is independent of  $\mathbf{y}_{-i}$ . Choose  $j \in N \setminus \{i\}$  arbitrarily. By the definition of the slope of a line, we have

$$\beta_1(\mathbf{y}) = \frac{f_j(\mathbf{y}_{-j}) - f_i(\mathbf{y}_{-i})}{x_j - x_i}, \quad \beta_1(\tilde{\mathbf{y}}) = \frac{f_j(\tilde{\mathbf{y}}_{-j}) - f_i(\tilde{\mathbf{y}}_{-i})}{x_j - x_i}.$$

Taking the difference, and noting that  $\mathbf{y}_{-i} = \tilde{\mathbf{y}}_{-i}$ , we get

$$\beta_1(\mathbf{y}) - \beta_1(\tilde{\mathbf{y}}) = \frac{f_j(\mathbf{y}_{-j}) - f_j(\tilde{\mathbf{y}}_{-j})}{x_j - x_i}.$$

Note that the RHS is independent of  $y_j$ . Since we chose  $j \in N \setminus \{i\}$  arbitrarily, it follows that  $\beta_1(\mathbf{y}) - \beta_1(\tilde{\mathbf{y}})$  is independent of  $\mathbf{y}_{-i}$ , implying that  $h$  is completely additively separable. Thus, there must exist functions  $\{g_i\}_{i \in N}$  such that  $\beta_1(\mathbf{y}) = \sum_{i \in N} g_i(\mathbf{y})$ .

We now want to calculate  $\beta_0$ . Recall that for every  $i \in N$ , the outcome for agent  $i$  is

$$f_i(\mathbf{y}_{-i}) = \beta_1(\mathbf{y}) \cdot x_i + \beta_0 = g_i(y_i) \cdot x_i + \sum_{j \in N \setminus \{i\}} g_j(y_j) \cdot x_i + \beta_0.$$

Since the LHS is independent of  $y_i$ , so must be the RHS. Hence,  $\beta_0 + g_i(y_i) \cdot x_i$  must be independent of  $y_i$  for each  $i \in N$ . This implies  $\beta_0 = c - \sum_{i \in N} g_i(y_i) \cdot x_i$  for some constant  $c$ , as desired. ■

Impartial mechanisms are not compelling from a statistical viewpoint. For instance, in the standard two-dimensional stochastic model where the data points are assumed to be generated by taking points on an underlying line and introducing i.i.d. errors in the dependent variables, it is easy to show that no impartial mechanism can produce an unbiased estimator of the underlying line. Nonetheless, impartial mechanisms help us establish the existence of a rather wide family of strategyproof mechanisms that are *not* group strategyproof. In fact, the next result shows that

almost all impartial mechanisms violate group strategyproofness.

**Proposition 3.4.** *For simple linear regression ( $d = 1$ ) with an admissible set of points, an impartial mechanism is group strategyproof if and only if it is a constant function (i.e., it returns a fixed regression line regardless of its input).*

*Proof.* By Theorem 3.6, an impartial mechanism for simple linear regression with an admissible set of points must be of the form given in Equation (3.3.14). We want to show that function  $g_i^x$  is constant for each  $i \in N$ . Suppose for contradiction that for some agent  $i \in N$ , function  $g_i^x$  is not constant. Thus, there exist  $y_i^1$  and  $y_i^2$  such that  $g_i^x(y_i^1) \neq g_i^x(y_i^2)$ . Fix an agent  $j \in N \setminus \{i\}$  and  $\mathbf{y}_{-\{i,j\}} \in \mathbb{R}^{n-2}$ . Let  $\hat{y}_j^1$  and  $\hat{y}_j^2$  denote the outcomes for agent  $j$  under the impartial mechanism when agent  $i$  reports  $y_i^1$  and  $y_i^2$ , respectively, and agents in  $N \setminus \{i, j\}$  report  $\mathbf{y}_{-\{i,j\}}$ . That is,

$$\hat{y}_j^t = g_i^x(y_i^t) \cdot (x_j - x_i) + \sum_{k \in N \setminus \{i,j\}} g_k^x(y_k) \cdot (x_j - x_k) + c^x, \forall t \in \{1, 2\}.$$

Note that  $g_i^x(y_i^1) \neq g_i^x(y_i^2)$  and  $x_i \neq x_j$  imply that  $\hat{y}_j^1 \neq \hat{y}_j^2$ . Now, suppose that the private values of the agents are  $(y_i^1, \hat{y}_j^2, \mathbf{y}_{-\{i,j\}})$ . In this case, the outcome for agent  $j$  is  $\hat{y}_j^1$ , which is different from her private value  $\hat{y}_j^2$ . If agent  $i$  changes her report to  $y_i^2$ , her own outcome would not change, but the outcome for agent  $j$  would change to  $\hat{y}_j^2$ , making agent  $j$  strictly better off. Thus, the coalition  $\{i, j\}$  successfully manipulates their reports, showing a violation of group strategyproofness.

For the reverse direction, note that all constant functions are group strategyproof. ■

### 3.4 CHARACTERIZING STRATEGYPROOF MECHANISMS

As mentioned in Chapter 3.3.1, Moulin (1980) studied the one-dimensional setting ( $d = 0$ ), and analytically characterized all strategyproof mechanisms for  $n$  agents. While we are unable to provide an analytical characterization for multidimensional linear regression, we provide two non-constructive characterizations, and discuss their implications.

Interestingly, to characterize strategyproof mechanisms for linear regression with  $n$  agents, we use the characterization of strategyproof mechanisms for the one-dimensional setting with a single agent. In this case, Moulin (1980) shows that a mechanism is strategyproof if and only if there exist

constants  $\alpha^1, \alpha^2 \in \overline{\mathbb{R}}$  such that when the agent reports  $y$ , the mechanism returns  $\hat{y} = \text{med}(y, \alpha^1, \alpha^2)$ . Constants  $\alpha^1$  and  $\alpha^2$  are called *phantoms*. First, we extend this result by providing an alternative characterization, which uses the following definition.

**Definition 3.14** (Locally Constant Function). *For  $A, B \subseteq \mathbb{R}$ , function  $f : A \rightarrow B$  is called locally constant at  $x \in A$  if there exists  $\varepsilon > 0$  such that for all  $x' \in [x - \varepsilon, x + \varepsilon]$ ,  $f(x') = f(x)$ .*

**Lemma 3.6.** *Suppose mechanism  $\pi : \mathbb{R} \rightarrow \mathbb{R}$  for the one-dimensional setting with a single agent elicits private value  $y$  from the agent and returns  $\pi(y)$ . Then,  $\pi$  being strategyproof is equivalent to each of the following conditions.*

- (a) *There exist constants  $\alpha^1, \alpha^2 \in \overline{\mathbb{R}} \triangleq \mathbb{R} \cup \{-\infty, \infty\}$  such that for all  $y \in \mathbb{R}$ ,  $\pi(y) = \text{med}(y, \alpha^1, \alpha^2)$ .*
- (b)  *$\pi$  is continuous, and for every  $y \in \mathbb{R}$ , either  $\pi(y) = y$  or  $\pi$  is locally constant at  $y$ .*

*Proof.* Part (a) is precisely the characterization of strategyproof mechanisms due to [Moulin \(1980, Proposition 3\)](#), applied to the case of a single agent.<sup>§</sup>

We would like to show that part (b) is equivalent to part (a). It is easy to check that a function  $\pi$  satisfying part (a) satisfies the conditions of part (b). We now show the converse.

Suppose that  $\pi$  is continuous, and for every  $y \in \mathbb{R}$ , either  $\pi(y) = y$  or  $\pi$  is locally constant at  $y$ . Let  $O = \{y \in \mathbb{R} : \pi \text{ is locally constant at } y\}$ . We first show that  $O$  is an open set, i.e., if  $y \in O$ , there exists a  $\delta > 0$  such that  $(y - \delta, y + \delta) \subseteq O$ . Fix a  $y \in O$ . Since  $\pi$  is locally constant at  $y$ , there exists an  $\varepsilon > 0$  such that  $\pi$  is constant in  $[y - \varepsilon, y + \varepsilon]$ . Set  $\delta = \varepsilon/2$ , and pick an arbitrary  $y' \in (y - \delta, y + \delta)$ . We want to show that  $y' \in O$ . Note that for  $\varepsilon' = \varepsilon/2$ ,  $[y' - \varepsilon', y' + \varepsilon'] \subseteq [y - \varepsilon, y + \varepsilon]$ . Hence,  $\pi$  is constant in  $[y' - \varepsilon', y' + \varepsilon']$ , implying that  $y' \in O$ . This concludes the proof that  $O$  is an open set.

Next, we use the well-known fact that any open subset of  $\mathbb{R}$  is a countable union of pairwise disjoint open intervals. That is, we can write  $O = \bigcup_{k \in \mathbb{N}} (a_k, b_k)$ , where  $a_k, b_k \in \overline{\mathbb{R}}$ . For  $k \in \mathbb{N}$ , because  $\pi$  is locally constant over  $(a_k, b_k)$ , and an open interval is a connected metric space, it follows that  $\pi$  is globally constant over  $(a_k, b_k)$ . That is, there exists a value  $t_k \in \mathbb{R}$  such that  $\pi(y) = t_k$  for all  $y \in (a_k, b_k)$ .

---

<sup>§</sup>Equivalently, one can use [Proposition 2](#), which characterizes strategyproof and anonymous mechanisms, as anonymity becomes trivial in case of a single agent.

We now show that for any  $k \in \mathbb{N}$  with  $a_k \neq b_k$  (i.e., the interval  $(a_k, b_k)$  is non-empty), it cannot be the case that both  $a_k$  and  $b_k$  are finite. Suppose for contradiction that both are finite. Note that continuity of  $\pi$  implies that  $\pi(a_k) = \pi(b_k) = t_k$ . However, since  $a_k, b_k \notin O$ , we have  $\pi(a_k) = a_k$  while  $\pi(b_k) = b_k$ , which is a contradiction because  $a_k \neq b_k$ . Hence, for every  $k \in \mathbb{N}$  with  $a_k \neq b_k$ , at least one of the two must lie in  $\{-\infty, \infty\}$ .

This leaves precisely five possibilities for the set  $O$ :  $\emptyset, \mathbb{R}, (-\infty, a)$  for  $a \in \mathbb{R}, (b, \infty)$  for  $b \in \mathbb{R}$ , and  $(-\infty, a) \cup (b, \infty)$  for  $a, b \in \mathbb{R}$  with  $b \geq a$ . We know that  $\pi$  is constant over each interval in  $O$ , and the identity function for every point outside  $O$ . For each of these five cases, we show that  $\pi$  must be of the form given in part (a) by identifying the corresponding constants  $\alpha^1$  and  $\alpha^2$ .

1.  $O = \emptyset$ :  $\pi$  is the identity function everywhere, i.e.,  $\alpha^1 = -\infty$  and  $\alpha^2 = \infty$ .
2.  $O = \mathbb{R}$ : There exists  $t \in \mathbb{R}$  such that  $\pi(y) = t$  for all  $y \in \mathbb{R}$ . This corresponds to  $\alpha^1 = \alpha^2 = t$ .
3.  $O = (-\infty, a)$  for  $a \in \mathbb{R}$ : Then  $\pi(y) = y$  for all  $y \geq a$ . In particular,  $\pi(a) = a$ . Because  $\pi$  is continuous and constant over  $(-\infty, a)$ , we have  $\pi(y) = a$  for  $y \in (-\infty, a)$ . This corresponds to  $\alpha^1 = a$  and  $\alpha^2 = \infty$ .
4.  $O = (b, \infty)$  for  $b \in \mathbb{R}$ : Similarly to case (3), this corresponds to  $\alpha^1 = -\infty$  and  $\alpha^2 = b$ .
5.  $O = (-\infty, a) \cup (b, \infty)$  for finite  $b \geq a$ : As argued in the previous two cases, for  $y \in (-\infty, a)$  we have  $\pi(y) = \pi(a) = a$ , and for  $y \in (b, \infty)$  we have  $\pi(y) = \pi(b) = b$ . For  $y \in [a, b]$ , we have  $\pi(y) = y$ . This corresponds to  $\alpha^1 = a$  and  $\alpha^2 = b$ .

This concludes our proof. ■

In the one-dimensional setting, Moulin (1980) observed that a mechanism is strategyproof if and only if its outcome is strategyproof in the report of each individual agent when other agents' reports are fixed. That is, a mechanism  $\pi : \mathbb{R}^n \rightarrow \mathbb{R}$  for  $n$  agents is strategyproof if and only if

$$\forall i \in [n], \exists \alpha_i^1, \alpha_i^2 \in \overline{\mathbb{R}} \text{ independent of } y_i \text{ s.t. } \pi(y_1, \dots, y_n) = \text{med}(y_i, \alpha_i^1, \alpha_i^2). \quad (3.4.1)$$

Moulin (1980) solved Equation (3.4.1) to derive an elegant analytical expression for  $\pi$  in terms of  $\{y_i\}_{i \in [n]}$ . Note that in this equation, the outcome  $\hat{y} = \pi(y_1, \dots, y_n)$  is common to all agents.

In contrast, in linear regression each agent  $i$  has a potentially different outcome  $\hat{y}_i$ . Like before, strategyproofness requires that each  $\hat{y}_i$  obey the conditions in Lemma 3.6, when seen as a function of  $y_i$ , when other agents' reports are fixed. However, the outcomes for different agents are now constrained so that  $(\mathbf{x}_i, \hat{y}_i)_{i \in N}$  lie on a hyperplane. This added complexity prevented us from solving the equations to derive an analytical characterization, despite significant effort. The only exception was the special case of *impartial* mechanisms, where we further restrict  $\hat{y}_i$  to be independent of  $y_i$  (Theorem 3.6). This corresponds to the case where  $\alpha_i^1 = \alpha_i^2$  for each agent  $i$ . Nonetheless, by simply applying Lemma 3.6 for every agent  $i$ , we obtain the following non-constructive characterization of strategyproof mechanisms for linear regression.

### Theorem 3.7

Given public information  $\mathbf{x}$ , mechanism  $M^\mathbf{x}$  for linear regression being strategyproof is equivalent to each of the following conditions.

- (a) For every  $\mathbf{y}_{-i} \in \mathbb{R}^{n-1}$  and  $i \in N$ , there exist  $\ell_i, h_i \in \overline{\mathbb{R}}$  such that  $\hat{y}_i(M^\mathbf{x}(\mathbf{y})) = \text{med}(y_i, \ell_i, h_i)$  for all  $y_i \in \mathbb{R}$ ;
- (b) For every  $\mathbf{y}_{-i} \in \mathbb{R}^{n-1}$  and  $i \in N$ , function  $f_i(\cdot) = \hat{y}_i(M(\cdot, \mathbf{y}_{-i}))$  is continuous, and for every  $y_i \in \mathbb{R}$ , either  $f_i(y_i) = y_i$  or  $f_i$  is locally constant at  $y_i$ .

The first condition provides an analytical form of  $\hat{y}_i$  in terms of  $y_i$ , and is perhaps the more useful characterization. For instance, we crucially use this characterization in the next section to give a lower bound on the efficiency of strategyproof mechanisms. Our earlier (more complex) proof of group strategyproofness of GRH mechanisms (Theorem 3.5) was also based on this condition, and identified the precise  $\ell_i$  and  $h_i$  for each agent  $i$ .

Note that for fixed  $\mathbf{y}_{-i}$ , we have  $\hat{y}_i = y_i$  when  $y_i \in [\ell_i, h_i]$ . For  $y_i \leq \ell_i$ ,  $\hat{y}_i = \ell_i$  is fixed, and for  $y_i \geq h_i$ ,  $\hat{y}_i = h_i$  is fixed. We therefore say that agent  $i$  is *influential* over the interval  $(\ell_i, h_i)$ , and call  $\ell_i$  and  $h_i$  the *lower* and *upper influence bounds*, respectively. Analysis of influence bounds has received attention in the statistics literature, where it is called *sensitivity analysis*. For instance, [Narula and Wellington \(1985\)](#) observed that under  $L_1$ -ERM, the regression hyperplane is unaffected when the dependent variable of a point is changed so that the point still lies on the same side of the

hyperplane as before. From Theorem 3.7, we can see that for every strategyproof mechanism, doing so should at least keep the outcome for agent  $i$  unchanged. Narula and Wellington (1985) also focused on computing the influence bounds. Theorem 3.7 lends a simple algorithm to compute influence bounds (see Chapter 3.4.1). Finally, note that while  $\hat{y}_i$  must be continuous in  $y_i$ , it need not be continuous in  $\mathbf{y}$  (see our discussion on Proposition 3.3).

### 3.4.1 COMPUTING INFLUENCE BOUNDS

Our characterization result (Theorem 3.7) establishes existence of influence bounds  $\ell_i, h_i \in \overline{\mathbb{R}}$  for each agent  $i$  as a function of the reports of the other agents. In this section, we address the problem of computing these bounds for a given strategyproof mechanism.

---

#### ALGORITHM 3.1: Computing Influence Bounds

---

**Input:** Data points  $\mathcal{D} = (\mathbf{x}_j, y_j)_{j \in N}$ , agent  $i \in N$ .

**Output:**  $\ell_i, h_i$

- 1  $Z \leftarrow$  set of hyperplanes  $\beta$  which pass through  $d + 1$  agents from  $N \setminus \{i\}$ ;
  - 2  $t_\beta \leftarrow \beta^\top \bar{\mathbf{x}}_i, \forall \beta \in Z$ ;
  - 3  $L \leftarrow \min_{\beta \in Z} t_\beta - 1$ ;
  - 4  $H \leftarrow \max_{\beta \in Z} t_\beta + 1$ ;
  - 5  $V_L \leftarrow M^x(L, \mathbf{y}_{-i})$ ;
  - 6  $V_H \leftarrow M^x(H, \mathbf{y}_{-i})$ ;
  - 7 **if**  $V_L = L$  **then**  $\ell_i \leftarrow -\infty$  **else**  $\ell_i \leftarrow V_L$ ;
  - 8 **if**  $V_H = H$  **then**  $h_i \leftarrow +\infty$  **else**  $h_i \leftarrow V_H$ ;
  - 9 **return**  $\ell_i, h_i$ ;
- 

Fix  $\mathbf{y}_{-i}$ . We begin from the simple observation that if  $\ell_i$  is finite, then for a sufficiently low value of  $y_i$  (any  $y_i \leq \ell_i$ ), we have that the outcome for agent  $i$  will be  $\hat{y}_i = \text{med}(y_i, \ell_i, h_i) = \ell_i$ . If  $\ell_i = -\infty$ , then for all  $y_i < h_i$ , the outcome for agent  $i$  will be  $\hat{y}_i = y_i$ . Thus, if we can identify a *sufficiently low* value of  $y_i$ , we can check if  $\hat{y}_i$  is equal to  $y_i$  (in which case  $\ell_i = -\infty$ ), or  $\hat{y}_i$  is equal to some other value (in which case this value must be  $\ell_i$ ). A symmetric observation holds for  $h_i$ .

While it is difficult to pin down a sufficiently low value for an arbitrary strategyproof mechanism, we can do so for the class of strategyproof mechanisms which are guaranteed to pass through  $d + 1$  data points in  $d + 1$  dimensions (e.g., the GRH mechanisms).

In this case, note that  $\ell_i$ , if finite, must be the point where a hyperplane containing *some*  $d + 1$  agents (excluding agent  $i$ ) intersects the vertical line at  $\mathbf{x}_i$ . Thus, if we iterate through all hyper-

planes passing through  $d + 1$  agents except agent  $i$ , and find their intersections with the vertical line at  $x_i$ , then any value lower than the lowest intersection point will work as a sufficiently low value. Once again, a symmetric observation can be made for  $h_i$ .

This provides an algorithm that runs in time that is polynomial in  $n$ , but exponential in  $d$ , and makes two calls to the strategyproof mechanism (one to identify  $\ell_i$  and one for  $h_i$ ). This is presented as Algorithm 3.1.

### 3.5 EFFICIENCY OF STRATEGYPROOF MECHANISMS

Insofar, we studied families of strategyproof mechanisms for linear regression. In the absence of strategic considerations, a popular mechanism for linear regression is the OLS (ordinary least squares), which is the empirical risk minimizer for the squared loss. Under this loss function, which is also called the *residual sum of squares* (RSS), the loss when choosing hyperplane  $\beta$  given data points  $\mathcal{D}$  is  $\text{RSS}(\mathcal{D}, \beta) = \sum_{i \in N} (y_i - \beta^\top \bar{x}_i)^2$ . A classic justification for the OLS is due to the Gauss-Markov theorem, which states that when the errors (deviations of data points from an underlying hyperplane we wish to identify) are stochastic, zero in expectation, uncorrelated, and of equal variance, the OLS is the *best linear unbiased estimator*.

However, in our strategic setting, the OLS is not strategyproof (Dekel et al., 2010). This raises the question: *Is there a strategyproof mechanism that is close to the OLS?* We assess this by the worst-case approximation ratio of a mechanism for the optimal squared loss.

**Definition 3.15** (Efficiency). *Given  $\mathbf{x}$ , we say that mechanism  $M^{\mathbf{x}}$  for linear regression is  $c$ -efficient if for every  $\mathcal{D} = (\mathbf{x}_i, y_i)_{i \in N}$ , we have  $\text{RSS}(\mathcal{D}, M^{\mathbf{x}}(\mathbf{y})) \leq c \cdot \inf_{\beta} \text{RSS}(\mathcal{D}, \beta)$ .*

We show that no strategyproof mechanism that is too close to the OLS can be strategyproof. The proof of the next result leverages our characterization of strategyproof mechanisms (Theorem 3.7).

#### Theorem 3.8

For  $n \geq 4$ , there exist  $\mathbf{x}$  for which no strategyproof mechanism is  $(2 - \epsilon)$ -efficient for any  $\epsilon > 0$ .

*Proof.* For simplicity of notation, we use  $n + 1$  agents instead of  $n$  agents (and assume  $n + 1 \geq 4$ , i.e.,  $n \geq 3$ ). We also consider simple linear regression ( $d = 1$ ); the proof easily extends to higher

dimensions by simply setting all other coordinates to zero. Fix  $n \geq 3$ . Consider a setting with  $n+1$  agents where  $x_i = i$  for  $i \in [n]$ , and  $x_{n+1} = X$ , where  $X$  is the solution of the following equation:

$$\frac{n^3 - n}{2(1 + 3n + 2n^2 + 6X^2 - 6Xn - 6X)} = 1. \quad (3.5.1)$$

Interested readers may note that  $X = \Theta(n^{1.5})$ . Let  $T$  denote the LHS in Equation (3.5.1).

Consider a strategyproof mechanism  $M^x$ . Suppose  $M^x$  is  $c$ -efficient. We want to show that  $c \geq 2$ . We consider a family of inputs  $y$ , in which we fix  $y_i = 0$  for  $i \in [n]$ , and vary  $y_{n+1} = Y$ . First, we note that the optimal RSS, as a function of  $Y$ , is given by

$$f_0(Y) = Y^2 \cdot \frac{n^3 - n}{2 + 5n + 4n^2 + n^3 - 12X - 12nX + 12X^2} = Y^2 \cdot \frac{T}{T+1} = \frac{Y^2}{2},$$

where the first transition is obtained by minimizing  $(Y - X \cdot \beta_1 - \beta_0)^2 + \sum_{i=1}^n (i \cdot \beta_1 + \beta_0)^2$  over all  $(\beta_1, \beta_0)$ , the second one follows through simple algebra, and the final one follows from Equation (3.5.1). For verification of these claims through Mathematica, see Figure 3.3.

```

Minimize[{\sum_{i=1}^n (a*i + (1 - a*X))^2, n > 2 && X > 2}, a]
Minimize[{\left(\sum_{i=1}^n (a*i + b)^2\right) + (a*X + b - 1)^2, n > 2 && X > 2}, {a, b}]
{{{\left[\begin{array}{ll} \frac{-n+n^3}{2 (1+3 n+2 n^2-6 X-6 n X-6 X^2)} & X > 2 \& n > 2 \\ \infty & \text{True} \end{array}\right]}, {{a \rightarrow \left[\begin{array}{ll} \frac{-3-3 n-6 X}{1+3 n+2 n^2-6 X-6 n X-6 X^2} & X > 2 \& n > 2 \\ \text{Indeterminate} & \text{True} \end{array}\right]}}}, {{\left[\begin{array}{ll} \frac{-n+n^3}{2+5 n+4 n^2+n^3-12 X-12 n X+12 X^2} & X > 2 \& n > 2 \\ \infty & \text{True} \end{array}\right]}, {{a \rightarrow \left[\begin{array}{ll} \frac{2 (3+3 n-6 X)}{-2+5 n+4 n^2+n^3-12 X-12 n X+12 X^2} & X > 2 \& n > 2 \\ \text{Indeterminate} & \text{True} \end{array}\right], b \rightarrow \left[\begin{array}{ll} \frac{2+8 n+10 n^2+4 n^3-6 X-12 n X-6 n^2 X}{(1+n) (2+5 n+4 n^2+n^3-12 X-12 n X+12 X^2)} & X > 2 \& n > 2 \\ \text{Indeterminate} & \text{True} \end{array}\right]}}}, {{FullSimplify[\frac{-n+n^3}{2+5 n+4 n^2+n^3-12 X-12 n X+12 X^2}] == \frac{\frac{n^3-n}{2 (1+3 n+2 n^2-6 X (X-n-1))}}{1+\frac{n^3-n}{2 (1+3 n+2 n^2-6 X (X-n-1))}}]}}, True

```

Figure 3.3: Verification of various claims through Mathematica

Recall that we fixed  $y_i$  for  $i \in [n]$ . Due to our characterization result (Theorem 3.7), there exist  $\ell, h \in \overline{\mathbb{R}}$  with  $\ell \leq h$  such that the line returned by the mechanism passes through  $(X, \text{med}(Y, \ell, h))$  for all  $Y$ . We take two cases.

*Case 1:  $h > 0$ .* Set  $Y = h$ . Then, the line returned by the mechanism passes through  $(X, h)$ . In

this case, we can show that the RSS of the mechanism is at least

$$f_1 = h^2 \cdot \frac{n^3 - n}{2(1 + 3n + 2n^2 + 6X^2 - 6Xn - 6X)} = h^2 \cdot T = h^2,$$

where the first transition is obtained by minimizing  $(Y - \beta_1 \cdot X - \beta_0)^2 + \sum_{i=1}^n (\beta_1 \cdot i + \beta_0)^2$  over all  $(\beta_1, \beta_0)$  which satisfy  $\beta_1 \cdot X + \beta_0 = Y$ , and the rest follows from Equation (3.5.1). For verification of these claims through Mathematica, see Figure 3.3. This implies  $c \geq f_1/f_0(h) = 2$ .

*Case 2:*  $h \leq 0$ . Set  $Y = 1$ . Then, the line returned by the mechanism passes through  $(X, h)$ . In this case, the RSS of the mechanism is at least  $f_2 = 1$  because agent  $n+1$  contributes  $(1-h)^2 \geq 1$  to the squared loss. Once again, we have  $c \geq f_2/f_0(1) = 2$ .

The proof is complete as we have  $c \geq 2$  in each case. ■

For  $n = 2$  agents (or  $n = d+1$  agents in  $d+1$  dimensions), there is an obvious 1-efficient strategyproof mechanism which returns a hyperplane passing through all input points. Theorem 3.7 leaves open the case of  $n = 3$  in two dimensions.

We finally remark here that none of the strategyproof mechanisms we study achieve a constant approximation. For instance, it is easy to show that  $L_1$ -ERM is  $n$ -efficient.

**Proposition 3.5.** *The  $L_1$ -ERM mechanism is  $n$ -efficient.*

*Proof.* Fix  $\mathcal{D} = (\mathbf{x}_i, y_i)_{i \in N}$ . Let  $\boldsymbol{\beta}^1$  and  $\boldsymbol{\beta}^*$  be the outputs of  $L_1$ -ERM and OLS, respectively. Then, we have

$$\text{RSS}(\mathcal{D}, \boldsymbol{\beta}^1) \leq \left( \sum_{i \in N} |y_i - (\boldsymbol{\beta}^1)^\top \bar{\mathbf{x}}_i| \right)^2 \leq \left( \sum_{i \in N} |y_i - (\boldsymbol{\beta}^*)^\top \bar{\mathbf{x}}_i| \right)^2 \leq n \cdot \text{RSS}(\mathcal{D}, \boldsymbol{\beta}^*),$$

where the first inequality follows from the power mean inequality, the second inequality holds because  $\boldsymbol{\beta}^1$  minimizes the sum of absolute losses, and the third inequality follows from the Cauchy-Schwarz inequality. This concludes the proof. ■

### 3.6 DISCUSSION AND OPEN QUESTIONS

The results of this chapter leave several open questions. Perhaps the most ambitious one is to find a constructive characterization of all strategyproof or group strategyproof mechanisms for linear

regression, which may allow us to pinpoint the most efficient strategyproof mechanism; [Caragianinis et al. \(2016\)](#) provide a similar analysis in the one-dimensional setting. It is easy to show that  $L_1$ -ERM is  $n$ -efficient (see Proposition 3.5). Does there exist a more efficient strategyproof mechanism? It would also be interesting to analyze efficiency in a stochastic setting where the data points are drawn from an underlying distribution.

The characterization result of [Moulin \(1980\)](#) for strategyproof and anonymous mechanisms in the one-dimensional setting extends the median to generalized medians by adding fixed phantom values, and then taking the median. It is also shown that adding  $n + 1$  phantoms is sufficient to obtain full generality. We can extend all our proposed families of mechanisms by adding a certain number of “phantom points” in  $\mathbb{R}^{d+1}$ , and then applying the mechanisms to the union of data points and phantom points. The resulting mechanism retains the incentive guarantees.<sup>¶</sup> Given  $n$  data points, how many phantoms are sufficient to obtain full generality? Do the phantoms play a role in obtaining the elusive constructive characterization?

Another interesting observation is that our generalized resistant hyperplane mechanisms are guaranteed pass through  $d + 1$  input points in  $d + 1$  dimensions. It is known that at least one minimizer of the  $L_1$  loss also has this property. It would be interesting to identify a generic family of conditions, which, when imposed in addition to the requirement of making  $d + 1$  residuals zero, yield group strategyproofness.

Finally, [Dekel et al. \(2010\)](#) study a regression setting in which a single agent may control multiple data points, show that  $L_1$ -ERM is no longer strategyproof, and provide novel strategyproof mechanisms. It would be useful to see if our ideas can be used to design additional strategyproof mechanisms in this model. Another interesting variant is when only a small number of data points are held by strategic agents, but the mechanism does not know which ones. A similar setting was studied by [Charikar et al. \(2017\)](#), but for classification and with adversarial manipulations.

---

<sup>¶</sup>We also considered adding phantom values directly in the equations where a median is used. However, most such attempts violated strategyproofness.

# 4

## No-Regret and Incentive-Compatible Online Learning

### 4.1 CHAPTER OVERVIEW

In this chapter, we move from the offline settings studied in Chapter 3 to a repeated, online one. Specifically, we study an online learning setting in which a learner makes predictions about a sequence of  $T$  binary events ([Vovk, 1990](#), [Littlestone and Warmuth, 1994](#), [Cesa-Bianchi et al., 1997](#), [Freund and Schapire, 1997](#), [Vovk, 1998](#), [Auer et al., 2002b](#)). The learner has access to a pool of  $K$  experts, each with beliefs about the likelihood of each event occurring. The standard goal of the learner is to output a sequence of predictions almost as accurate as those of the best fixed expert

in hindsight (i.e., *no regret*). As we explained in Chapter 1.2.1, in many applications of the prediction with expert advice paradigm (such as FiveThirtyEight and The Good Judgment Project) it is natural to expect that forecasters may try to strategize to increase their sway over the algorithm’s prediction. Note that standard, state-of-the-art algorithms that achieve no regret —e.g., Multiplicative Weights Update (MWU)— usually violate strategyproofness (see Example 4.1).

We focus on this problem and wish to simultaneously achieve algorithms that are no regret for the learner and incentive compatible for the forecasters both for the full and the bandit information setting. We discover an unexpected link between *wagering mechanisms* (Lambert et al., 2008, 2015), a type of multi-agent scoring rule that allows a principal to elicit the beliefs of a group of agents without taking on financial risk and we use it to achieve our twofold desiderata. For the full information setting, we introduce Weighted-Score Update (WSU), which yields regret  $\mathcal{O}(\sqrt{T \ln K})$ , matching the optimal regret achievable for general loss functions, even without incentive guarantees. For the partial information setting, we introduce Weighted-Score Update with Uniform Exploration (WSU-UX), which achieves regret  $O(T^{2/3}(K \ln K)^{1/3})$ .

We focus mostly on experts that strategize about their influence at the next round. However, we obtain a partial extension for forward-looking experts. Building on a mechanism that was proposed for forecasting competitions (Witkowski et al., 2018), we identify algorithm ELF-X for the full information setting that is incentive-compatible and achieves diminishing regret in simulations.

Our theoretical results are supported by experiments on data gathered from an online prediction contest on FiveThirtyEight. Our algorithms achieve regret almost identical to the classic (and not incentive-compatible) Multiplicative Weights Update (MWU) Freund and Schapire (1997) and EXP3 Auer et al. (2002b) algorithms in the full and partial information settings respectively, though WSU falls short of the optimal regret achieved by Hedge for quadratic loss.

#### 4.1.1 RELATED WORK

Other work has drawn connections between online learning and incentive-compatible forecasting, particularly in the context of prediction markets (Abernethy et al., 2013, Abernethy and Frongillo, 2011, Frongillo et al., 2012, Hu and Storkey, 2014). The results of this chapter are most closely related to the work of Roughgarden and Schrijvers (2017), but differs from theirs in several important

ways. Most crucially, Roughgarden and Schrijvers consider algorithms that maintain *unnormalized* weights over the experts, and they assume that an expert's incentives are only affected by these weights. In the present chapter, incentives are tied to the expert's *normalized* weight—that is, his probability of being selected by the learning algorithm. We argue that normalized weights better reflect experts' incentives in reality, since reputation tends to be relative more than absolute; put another way, doubling the unnormalized weight of every expert should not increase an expert's utility, since his influence over the learner's prediction remains the same. Under Roughgarden and Schrijvers' model, the design problem is fairly simple when the loss function is a proper loss [Reid and Williamson \(2009\)](#)—that is, one that can be elicited by a proper scoring rule ([Savage, 1971](#), [Gneiting and Raftery, 2007](#)), such as the quadratic loss function—and can be solved with a multiplicative weights algorithm. Because of this, they focus primarily on the absolute loss function, which is not a proper loss. In contrast, in our model, the design problem is nontrivial even for these “easier” proper loss functions.

Conceptually, the present chapter builds on work by [Witkowski et al. \(2018\)](#), who use competitive scoring rules—a subclass of wagering mechanisms—to design incentive-compatible forecasting competitions. We discuss their work further in Chapter 4.5. This chapter also has connections with the work of [Orabona and Pál \(2016\)](#), who introduce a class of *coin-betting* algorithms for online learning. Although [Orabona and Pál \(2016\)](#) do not address incentives and do not make a connection with the wagering mechanisms literature, our WSU algorithm can be interpreted as a coin-betting algorithm.\*

## 4.2 MODEL AND PRELIMINARIES

We consider a setting where a learner interacts with a set of  $K$  experts, each making probabilistic predictions about a sequence of  $T$  binary outcomes.<sup>†</sup> At each round  $t \in [T]$ , each expert  $i \in [K]$  has a private belief  $b_{i,t} \in [0, 1]$ , unknown to the learner, about the outcome for that round. Both the experts' beliefs and the sequence of outcomes may be chosen arbitrarily and adversarially.

---

\*In the language of coin betting, in WSU experts wager an  $\eta$  fraction of their wealth on the positive realization of the event, and the actual outcome of each coin flip is the expert's loss minus the weighted average loss of all other experts.

<sup>†</sup>We focus on binary outcomes to simplify presentation, but our techniques can be applied more broadly.

In the full information setting, each expert reports his prediction  $p_{i,t} \in [0, 1]$  to the learner. The learner then chooses her own prediction  $\bar{p}_t \in [0, 1]$  and observes the outcome realization  $r_t \in \{0, 1\}$ . Finally, the learner and the experts incur losses  $\ell_t = \ell(\bar{p}_t, r_t)$  and  $\ell_{i,t} = \ell(p_{i,t}, r_t)$ ,  $\forall i \in [K]$ , where  $\ell : [0, 1] \times \{0, 1\} \rightarrow [0, 1]$  is a bounded loss function.<sup>‡</sup> As is common in the literature, we restrict our attention to algorithms in which the learner maintains a round-specific probability distribution  $\pi_t = (\pi_{1,t}, \dots, \pi_{K,t})$  over the experts, and chooses her prediction  $\bar{p}_t$  according to this distribution. Unless specified, this means that the learner predicts  $\bar{p}_t = p_{i,t}$  with probability  $\pi_{i,t}$ ; some of our results additionally apply when  $\bar{p}_t = \sum_{i \in [K]} \pi_{i,t} p_{i,t}$ .

Under partial information, the protocol remains the same except that the learner is explicitly restricted to choosing a single expert  $I_t$  on each round  $t$  (according to distribution  $\pi_t$ ) and does not observe the predictions of other experts.

The goal of the learner is twofold. First, she wishes to incur a total loss that is not much worse than the loss of the best fixed expert in hindsight. This is captured using the classic notion of *regret*:

$$R = \mathbb{E} \left[ \sum_{t \in [T]} \ell_t - \min_{i \in [K]} \sum_{t \in [T]} \ell_{i,t} \right],$$

where the expectation is taken with respect to randomness in the learner's choice of  $\bar{p}_t$ .

No-regret algorithms have been proposed in both the full and partial information settings. Many, such as Hedge (Freund and Schapire, 1997) and MWU (Arora et al., 2012), yield regret  $\mathcal{O}(\sqrt{T \ln K})$  for general loss functions by maintaining unnormalized weights  $w_{i,t}$  for each expert  $i$  that are updated multiplicatively at each round. Hedge uses the update rule  $w_{i,t+1} = w_{i,t} \exp(-\eta \ell_{i,t})$ , while MWU uses  $w_{i,t+1} = w_{i,t} (1 - \eta \ell_{i,t})$  for appropriately chosen values of  $\eta$ . These weights are then normalized to arrive at distribution  $\pi_t$ . For the case of exp-concave loss functions, such as the quadratic loss, Kivinen and Warmuth (1999) showed that by aggregating experts' predictions and tuning  $\eta$  appropriately, Hedge can achieve regret  $\mathcal{O}(\ln K)$ .

For the partial information setting, the EXP3 algorithm of Auer et al. (2002b) achieves a regret of  $\mathcal{O}(\sqrt{TK \ln K})$ . EXP3 maintains a set of expert weights similar to those of Hedge. However, since the learner can only observe the prediction of the chosen expert, she uses an unbiased estimator

---

<sup>‡</sup>Without loss of generality, the range is  $[0, 1]$ . Indeed, any bounded loss function can be scaled to  $[0, 1]$ .

$\hat{\ell}_{i,t}$  of each expert  $i$ 's loss in her updates in place of  $\ell_{i,t}$ . So:  $w_{i,t+1} = w_{i,t} \exp(-\eta \hat{\ell}_{i,t})$ .

The second goal of the learner is to incentivize experts to truthfully report their private beliefs. In our model, at each round  $t$ , each expert  $i$  chooses his report  $p_{i,t}$  strategically to maximize the probability  $\pi_{i,t+1}$  that he is chosen at round  $t + 1$ . An algorithm is *incentive-compatible* if experts maximize this probability by reporting  $p_{i,t} = b_{i,t}$ , irrespective of the reports of the other experts.

**Definition 4.1** (Incentive Compatibility). *An online learning algorithm is incentive compatible if for every round  $t \in [T]$ , every expert  $i$  with belief  $b_{i,t}$ , every report  $p_{i,t}$ , every vector of reports of the other experts  $\mathbf{p}_{-i,t}$ , and every history of reports  $(\mathbf{p}_{t'})_{t' < t}$  and outcomes  $(r_{t'})_{t' < t}$ ,*

$$\begin{aligned} & \mathbb{E}_{r_t \sim \text{Bern}(b_{i,t})} [\pi_{i,t+1} | (b_{i,t}, \mathbf{p}_{-i,t}), r_t, (r_{t'})_{t' < t}, (\mathbf{p}_{t'})_{t' < t}] \\ & \geq \mathbb{E}_{r_t \sim \text{Bern}(b_{i,t})} [\pi_{i,t+1} | (p_{i,t}, \mathbf{p}_{-i,t}), r_t, (r_{t'})_{t' < t}, (\mathbf{p}_{t'})_{t' < t}]. \end{aligned}$$

where  $r \sim \text{Bern}(b)$  denotes a random variable  $r$  taking value 1 with probability  $b$  and 0 otherwise.

Incentive compatibility guarantees that any regret bounds apply not only with respect to the reports of the experts, but also with respect to their beliefs. This notion of regret is often called *strategic regret*, and in general may be higher or lower than standard regret. For an incentive-compatible algorithm, the two notions coincide.

For incentive compatibility, we restrict attention to proper loss functions [Reid and Williamson \(2009\)](#), referred to in the forecasting literature as proper scoring rules ([McCarthy, 1956](#), [Savage, 1971](#), [Gneiting and Raftery, 2007](#)).

**Definition 4.2.** A loss function  $\ell$  is said to be proper if

$$\mathbb{E}_{r \sim \text{Bern}(b)} [\ell(p, r)] \geq \mathbb{E}_{r \sim \text{Bern}(b)} [\ell(b, r)], \forall p \neq b.$$

Restricting attention to proper loss functions, we are guaranteed that an expert who cares only about his expected loss would truthfully report his beliefs. However, this does not apply for experts who care about their probability of being chosen by the learner, as in our setting. Indeed, known online learning algorithms fail to be incentive-compatible even for proper loss functions. We illustrate this in the following example for MWU with the (proper) quadratic loss function  $\ell(p, r) = (p - r)^2$ .

Here the normalization of weights by the factor  $\sum_{j \in [K]} w_{j,t}$ , which depends on both  $p_{i,t}$  and  $r_t$ , can create incentives for agent  $i$  to deviate. We note that a similar counterexample can be proved for Gradient Descent too, and we include it in Appendix A.1.

**Example 4.1.** Let  $\ell(p, r) = (p - r)^2$ . Under standard initialization for MWU,  $w_{i,1} = 1$  for all  $i \in [K]$ . Suppose that  $b_{1,1} = 0.5$  and  $p_{i,1} = 0$  for all  $i \in \{2, \dots, K\}$ . Then  $\mathbb{E}[\pi_{1,2}]$ , the expected probability that expert 1 is chosen at time 2 under MWU with respect to his own beliefs, is

$$0.5 \left( \frac{1 - \eta(1 - p_{1,1})^2}{K - \eta(1 - p_{1,1})^2 - \eta(K-1)} \right) + 0.5 \left( \frac{1 - \eta p_{1,1}^2}{K - \eta p_{1,1}^2} \right).$$

For  $K \geq 3$  and  $T \geq 9 \ln(3)$ , the denominator in the first term is less than the denominator in the second term, independent of  $p_{1,1}$ . The derivative of  $\mathbb{E}[\pi_{1,2}]$  with respect to  $p_{1,1}$  is therefore strictly positive at 0.5, implying that expert 1 maximizes his utility by reporting some  $p_{1,1} > 0.5$ .

Thus, unlike in the setting of Roughgarden and Schrijvers (2017), using a proper loss function with a standard algorithm is not enough, and new algorithmic ideas are needed. To derive our algorithms, we draw a connection between online learning and *wagering mechanisms*, one-shot elicitation mechanisms that allow experts to bet on the quality of their predictions relative to others. In the one-shot wagering setting introduced by Lambert et al. (2008), each agent  $i \in [K]$  holds a belief  $b_i \in [0, 1]$  about the likelihood of an event. Agent  $i$  reports a probability  $p_i$  and a wager  $w_i \geq 0$ . A wagering mechanism,  $\Gamma$ , maps the reports  $\mathbf{p} = (p_1, \dots, p_K)$ , wagers  $\mathbf{w} = (w_1, \dots, w_K)$ , and the realization  $r$  of the binary event to payments  $\Gamma_i(\mathbf{p}, \mathbf{w}, r)$  for each agent  $i$ . The purpose of the wager is to allow each agent to set a maximum allowable loss, which is captured by imposing the constraint that  $\Gamma_i(\mathbf{p}, \mathbf{w}, r) \geq 0, \forall i \in [K]$ . We restrict our attention to *budget-balanced* wagering mechanisms for which  $\sum_{i \in [K]} \Gamma_i(\mathbf{p}, \mathbf{w}, r) = \sum_{i \in [K]} w_i$ .

A wagering mechanism  $\Gamma$  is said to be *incentive-compatible* if for every agent  $i \in [K]$  with belief  $b_i \in [0, 1]$ , every report  $p_i \in [0, 1]$ , every vector of reports of the other agents  $\mathbf{p}_{-i}$ , and every vector of wagers  $\mathbf{w}$ :

$$\mathbb{E}_{r \sim \text{Bern}(b_i)} [\Gamma_i((b_i, \mathbf{p}_{-i}), \mathbf{w}, r)] \geq \mathbb{E}_{r \sim \text{Bern}(b_i)} [\Gamma_i((p_i, \mathbf{p}_{-i}), \mathbf{w}, r)].$$

Lambert et al. (2008, 2015) proposed the Weighted Score Wagering Mechanisms (WSWMs), which are a class of budget-balanced and incentive-compatible wagering mechanisms. Fixing any proper loss function  $\ell$  bounded in  $[0, 1]$ , agent  $i$  receives

$$\Gamma_i^{\text{WSWM}}(\mathbf{p}, \mathbf{w}, r) = w_i \left( 1 - \ell(p_i, r) + \sum_{j \in [K]} w_j \ell(p_j, r) \right).$$

WSWMs are incentive-compatible because the payment an agent receives is a linear function of his loss, measured by a proper loss function. An agent makes a profit (i.e., receives payment greater than his wager), whenever his loss is smaller than the wager-weighted average agent loss, so accurate agents are more likely to increase their wealth.

### 4.3 THE FULL INFORMATION SETTING

In this section, we present and analyze an online prediction algorithm, Weighted-Score Update (WSU), for the full information setting. We show that WSU is incentive-compatible and achieves regret  $\mathcal{O}(\sqrt{T \ln K})$ .

Our key observation is that we can define a black-box reduction that transforms any budget-balanced wagering mechanism  $\Gamma$  to an online learning algorithm by setting  $\pi_{t+1} = \Gamma(\mathbf{p}_t, \pi_t, r_t)$ . Here we can interpret an expert's weight according to distribution  $\pi_t$  as their currency. Each expert "wagers"  $\pi_t$  at time  $t$  and receives a payoff  $\pi_{t+1}$ , which depends on the reports of the experts  $\mathbf{p}$  and the realization  $r_t$ . It is easy to see that any online prediction algorithm that is derived from an incentive-compatible wagering mechanism will in turn be incentive-compatible, because any misreport that increases weight  $\pi_{t+1}$  would also be a successful misreport in the wagering setting.

One might hope that applying this reduction to the WSM would directly yield a no-regret online learning algorithm. But this is not the case, due to the fact that an expert who makes an inaccurate prediction can lose too much of his wealth (probability) if all other experts have low loss, and it can take a long time to recover from this. To handle this, we allow experts to "wager" only an  $\eta$  fraction of their current probability at each round for some  $\eta \in (0, 0.5]$ . This guarantees that no expert can obtain a probability  $\pi_{i,t}$  close to zero without having made a long series of inaccurate predictions.

Formally, the update rule of our algorithm, the Weighted-Score Update (WSU), is defined by:

$$\pi_{i,t+1} = \eta \Gamma_i^{\text{WSWM}}(\mathbf{p}_t, \boldsymbol{\pi}_t, r_t) + (1 - \eta) \pi_{i,t}, \quad (4.3.1)$$

with weights  $\pi_{i,1}$  initialized to  $\pi_{i,1} = 1/K$  for all  $i$ .

We must show that  $\boldsymbol{\pi}_t$  is a valid probability distribution over experts at each  $t$ . This follows from the WSM being budget-balanced.

**Lemma 4.1.** *The weights  $\boldsymbol{\pi}_t$  of WSU form well-defined probability distributions for all  $t$ .*

*Proof.* To show that a distribution is valid, we must show that the components are non-negative and sum to one. We do this inductively. The base case is satisfied since  $\pi_{i,1} = 1/K$  for all  $i$ . Now assume that  $\boldsymbol{\pi}_t$  is a valid probability distribution. For  $t + 1$ , from Equation 4.3.1, we have  $\pi_{i,t+1} \geq \eta \Gamma_i^{\text{WSWM}}(\mathbf{p}_t, \boldsymbol{\pi}_t, r_t) \geq 0$  where the last inequality follows from the properties of WSM and the assumption that  $\boldsymbol{\pi}_t$  is a valid distribution. Also:

$$\sum_{i \in [K]} \pi_{i,t+1} = \eta \sum_{i \in [K]} \Gamma_i^{\text{WSWM}}(\mathbf{p}_t, \boldsymbol{\pi}_t, r_t) + (1 - \eta) \sum_{i \in [K]} \pi_{i,t} = \eta \sum_{i \in [K]} \pi_{i,t} + (1 - \eta) \sum_{i \in [K]} \pi_{i,t} = 1$$

where the second equality follows from the fact that WSM is budget balanced and the final equality from the assumption that  $\boldsymbol{\pi}_t$  is a valid distribution. ■

Rewriting the WSU update rule in terms of relative loss  $L_{i,t} = \ell_{i,t} - \sum_{j \in [K]} \pi_{j,t} \ell_{j,t}$ , we can see that the form of the update is quite familiar. In particular, from Equation 4.3.1,

$$\begin{aligned} \pi_{i,t+1} &= \eta \pi_{i,t} \left( 1 - \ell_{i,t} + \sum_{j \in [K]} \pi_{j,t} \ell_{j,t} \right) + (1 - \eta) \pi_{i,t} \\ &= \pi_{i,t} (1 - \eta L_{i,t}). \end{aligned} \quad (4.3.2)$$

This resembles the update rule for the (unnormalized) weights maintained by MWU, but with  $L_{i,t}$  in place of  $\ell_{i,t}$ . The D-Prod algorithm of Even-Dar et al. (2008) involves a similar update, but uses loss relative to a single fixed distribution over experts instead of  $\boldsymbol{\pi}_t$ .

We are now ready to prove our guarantees. The proof of Theorem 4.1 proceeds similarly to the

standard proof that MWU satisfies no regret. However, our proof is slightly simpler because we do not need to make a distinction between (unnormalized) weights and (normalized) probabilities. We can therefore avoid introducing the standard potential function used in proofs of no regret.

**Theorem 4.1: Regret of WSU**

WSU is incentive-compatible and for step size  $\eta = \sqrt{\frac{\ln K}{T}}$  yields regret  $R \leq 2\sqrt{T \ln K}$ .

*Proof.* For incentive compatibility, note that from Equation (4.3.1),  $\pi_{i,t+1}$  is a convex combination of a WSM payment and  $\pi_{i,t}$ , which cannot be influenced by  $i$ 's report at time  $t$ . Since truthful reporting (at least weakly) maximizes each of these components, it also maximizes the sum.

For the regret, denoting by  $i^*$  the best expert in hindsight,

$$1 \geq \pi_{i^*,T+1} = \pi_{i^*,T}(1 - \eta L_{i^*,T}) = \pi_{i^*,1} \prod_{t \in [T]} (1 - \eta L_{i^*,t}) = \frac{1}{K} \prod_{t \in [T]} (1 - \eta L_{i^*,t}).$$

Taking the logarithm for both sides of this inequality, we get

$$0 \geq -\ln K + \sum_{t \in [T]} \ln(1 - \eta L_{i^*,t}) \geq -\ln K + \sum_{t \in [T]} (-\eta L_{i^*,t} - \eta^2 L_{i^*,t}^2),$$

where the last inequality comes from the fact that for  $x \leq 1/2$ ,  $\ln(1 - x) \geq -x - x^2$  (see Lemma B.2). Rearranging and dividing both sides by  $\eta$  yields

$$-\sum_{t \in [T]} L_{i^*,t} \leq \frac{\ln K}{\eta} + \eta \sum_{t \in [T]} L_{i^*,t}^2.$$

Since we have  $\sum_{t \in [T]} L_{i^*,t} = \sum_{t \in [T]} \ell_{i^*,t} - \sum_{t \in [T]} \sum_{j \in [K]} \pi_{j,t} \ell_{j,t} = -R$ , this becomes

$$R \leq \frac{\ln K}{\eta} + \eta \sum_{t \in [T]} L_{i^*,t}^2 \leq \frac{\ln K}{\eta} + \eta T.$$

Finally, tuning  $\eta = \sqrt{\ln(K)/T}$  gives us the result. ■

If  $T$  is not known in advance, a standard doubling trick (Auer et al., 2002b) can be applied with only a constant factor increase in regret; see Appendix A.2.2 for details.

The regret and incentive-compatibility guarantees of WSU in Theorem 4.1 hold for all  $[0, 1]$ -bounded proper loss functions  $\ell$ . If  $\ell$  is also convex, then these guarantees carry over to a (more practical) variant of WSU, termed WSU-Aggr, that uses the same update rule but sets  $\bar{p}_t = \sum_{i \in [K]} \pi_{i,t} p_{i,t}$  rather than choosing a single expert. Incentive compatibility is immediate. The regret bound follows from the fact that, by Jensen's inequality,

$$\sum_{t \in [T]} \ell \left( \sum_{i \in [K]} \pi_{i,t} p_{i,t}, r_t \right) \leq \sum_{t \in [T]} \sum_{i \in [K]} \pi_{i,t} \ell(p_{i,t}, r_t).$$

#### 4.4 THE PARTIAL INFORMATION SETTING

The encouraging results of the previous section apply only when the learner has access to the reports of all experts. But what if the learner has only partial information regarding these reports and still wants to incentivize all experts to report their predictions truthfully? In this section, we provide and analyze a novel algorithm, Weighted-Score Update with Uniform Exploration (WSU-UX), that is simultaneously no-regret and incentive-compatible in the bandit setting in which the learner chooses a single expert  $I_t$  at each round and observes only that expert's prediction. We show this algorithm has regret  $\mathcal{O}(T^{2/3}(K \ln K)^{1/3})$ . This guarantee is weaker than that of EXP3, but we see in Chapter 4.6 that WSU-UX can perform similarly to EXP3 in practice with the additional advantage of incentive compatibility.

One might think that the standard trick of replacing the loss  $\ell_{i,t}$  with an unbiased estimator  $\hat{\ell}_{i,t}$  in the WSU update rule would suffice to guarantee both incentive compatibility and a regret rate of  $\mathcal{O}(\sqrt{T \ln K})$ . Specifically, following Auer et al. (2002b), we might consider setting  $\hat{\ell}_{i,t} = 0$  for all experts  $i \neq I_t$  whose predictions we do not observe, and  $\hat{\ell}_{I_t,t} = \frac{\ell_{I_t,t}}{\pi_{I_t,t}}$  for the chosen expert. But since these estimated losses are unbounded, this could lead to weights  $\pi_{i,t}$  moving outside of  $[0, 1]$ , and we would no longer have a valid algorithm.

To solve this, we mix a distribution generated through WSU-style updates with a small amount of the uniform distribution. This does not affect incentives, since the experts cannot alter the uniform distribution, and has the convenient property that the estimated loss function is now bounded. By carefully tuning parameters, we are able to guarantee a valid probability distribution over experts.

The resulting updates are given in Algorithm 4.1.

---

**ALGORITHM 4.1: WSU-UX**


---

- 1 **Parameters:**  $\eta$  and  $\gamma$  such that  $0 < \eta, \gamma < 1/2$  and  $\eta K/\gamma \leq 1/2$
  - 2 Set  $\pi_{i,1} = \frac{1}{K}, \forall i \in [K]$ .
  - 3 **for**  $t \in [T]$  **do**
  - 4     Choose expert  $I_t \sim \tilde{\pi}_{i,t} = (1 - \gamma)\pi_{i,t} + \frac{\gamma}{K}$ .
  - 5     Compute:  $\hat{\ell}_{I_t,t} = \frac{\ell_{I_t,t}}{\tilde{\pi}_{I_t,t}}$  and  $\hat{\ell}_{i,t} = 0, \forall i \neq I_t$ .
  - 6     Update  $\pi_{i,t+1} = \pi_{i,t} \left( 1 - \eta \left( \hat{\ell}_{i,t} - \sum_{j \in [K]} \pi_{j,t} \hat{\ell}_{j,t} \right) \right)$ .
- 

We first prove that this is a valid algorithm, that is, that the distributions  $\tilde{\pi}_t$  from which an expert is selected are valid, under appropriate settings of  $\eta$  and  $\gamma$ .

**Lemma 4.2.** *If  $\frac{\eta K}{\gamma} \leq \frac{1}{2}$ , the WSU-UX weights  $\pi_t$  and  $\tilde{\pi}_t$  are valid probability distributions for all  $t$ .*

*Proof.* We prove this inductively for  $\pi_t$  and  $\tilde{\pi}_t$  simultaneously. The base case is trivial since at time  $t = 1, \forall i \in [K], \pi_{i,1} = \tilde{\pi}_{i,1} = 1/K$ . Now assume that for some  $t$  both  $\pi_t$  and  $\tilde{\pi}_t$  are valid probability distributions. We distinguish two cases. First, suppose  $i \neq I_t$ . Then, since  $\hat{\ell}_{i,t} = 0$ , the WSU-UX update rule becomes

$$\pi_{i,t+1} = \pi_{i,t} \left( 1 - \eta \left( 0 - \pi_{I_t,t} \frac{\ell_{I_t,t}}{\tilde{\pi}_{I_t,t}} \right) \right) \geq 0.$$

Second, suppose  $i = I_t$ . Then

$$\begin{aligned} \pi_{i,t+1} &= \pi_{i,t} \left( 1 - \eta \left( \frac{\ell_{i,t}}{\tilde{\pi}_{i,t}} - \pi_{i,t} \frac{\ell_{i,t}}{\tilde{\pi}_{i,t}} \right) \right) = \pi_{i,t} \left( 1 - \eta \frac{\ell_{i,t}}{\tilde{\pi}_{i,t}} (1 - \pi_{i,t}) \right) \\ &\geq \pi_{i,t} \left( 1 - \frac{\eta}{\tilde{\pi}_{i,t}} \right) \geq \pi_{i,t} \left( 1 - \eta \frac{K}{\gamma} \right) \geq 0, \end{aligned}$$

where the penultimate inequality comes from the fact that  $\tilde{\pi}_{i,t} \geq \gamma/K$ , since by the inductive assumption  $\pi_{i,t} \geq 0$ . The last follows from the assumption that  $\eta K/\gamma \leq 1/2$ . Moreover, for the sum

of probabilities we get:

$$\begin{aligned}
\sum_{i \in [K]} \pi_{i,t+1} &= \sum_{i \in [K]} \pi_{i,t} \left( 1 - \eta \left( \widehat{\ell}_{i,t} - \sum_{j \in [K]} \pi_{j,t} \widehat{\ell}_{j,t} \right) \right) \\
&= \sum_{i \in [K]} \pi_{i,t} - \eta \left( \sum_{i \in [K]} \pi_{i,t} \widehat{\ell}_{i,t} - \sum_{i \in [K]} \pi_{i,t} \sum_{j \in [K]} \pi_{j,t} \widehat{\ell}_{j,t} \right) \\
&= 1 - \eta \left( \sum_{i \in [K]} \pi_{i,t} \widehat{\ell}_{i,t} - \sum_{j \in [K]} \pi_{j,t} \widehat{\ell}_{j,t} \right) = 1.
\end{aligned}$$

Thus  $\pi_{t+1}$  is valid. Since  $\tilde{\pi}_{t+1}$  is a convex combination of two probability distributions, it is also a probability distribution, completing the inductive argument.  $\blacksquare$

We are now ready to state the main theorem. The requirement that  $T \geq K \ln K$  ensures that the precondition of Lemma 4.2 is satisfied for the settings of  $\eta$  and  $\gamma$  used.

**Theorem 4.2: Regret of WSU-UX**

For  $T \geq K \ln K$  and parameters  $\eta = \left( \frac{\ln K}{4K^{1/2}T} \right)^{2/3}$  and  $\gamma = \left( \frac{K \ln K}{4T} \right)^{1/3}$ , WSU-UX is incentive compatible and yields regret  $R \leq 2(4T)^{2/3}(K \ln K)^{1/3}$ .

The proof of the theorem will follow from a series of claims and lemmas. We first examine the moments of  $\widehat{\ell}_{i,t}$  and verify that it is an unbiased estimator of  $\ell_{i,t}$ .

**Lemma 4.3 (Moments).** *Taking expectation with respect to the choice of expert at round  $t$  and keeping all else fixed,  $\forall i \in [K], t \in [T]$ ,  $\mathbb{E}_{I_t \sim \tilde{\pi}_t} [\widehat{\ell}_{i,t}] = \ell_{i,t}$ . Furthermore,*

$$\mathbb{E}_{I_t \sim \tilde{\pi}_t} [\widehat{\ell}_{i,t}^2] = \frac{\ell_{i,t}^2}{\tilde{\pi}_{i,t}} \leq \frac{1}{\tilde{\pi}_{i,t}}. \quad (4.4.1)$$

*Proof.* For the first moment, we have:

$$\mathbb{E}_{I_t \sim \tilde{\pi}_t} [\widehat{\ell}_{i,t}] = \sum_{j \in [K]} \tilde{\pi}_{j,t} \frac{\ell_{i,t} \mathbf{1}\{j=i\}}{\tilde{\pi}_{i,t}} = \ell_{i,t}.$$

For the second moment, we have:

$$\mathbb{E}_{I_t \sim \tilde{\pi}_t} [\hat{\ell}_{i,t}^2] = \sum_{j \in [K]} \tilde{\pi}_{j,t} \frac{\ell_{i,t}^2 \mathbb{1}\{i=j\}}{\tilde{\pi}_{i,t}^2} = \frac{\ell_{i,t}^2}{\tilde{\pi}_{i,t}^2} \leq \frac{1}{\tilde{\pi}_{i,t}},$$

where the last inequality uses the fact that  $\ell_{i,t} \in [0, 1], \forall i \in [K], \forall t \in [T]$ . ■

We next provide a second-order regret bound. It differs from the standard second-order regret bounds presented for bandit algorithms (see e.g., [Bubeck and Cesa-Bianchi \(2012, Chapter 3\)](#)) in that it relates the “estimated regret” of the learner to the second moment of the estimated loss of the best-fixed expert in hindsight.

**Lemma 4.4** (Second-Order Bound). *For WSU-UX, the probability vectors  $\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_T$  and the estimated losses  $\hat{\ell}_{i,t}$  for  $i \in [K], t \in [T]$  induce the following second-order bound:*

$$\sum_{t \in [T]} \sum_{i \in [K]} \pi_{i,t} \hat{\ell}_{i,t} - \sum_{t \in [T]} \hat{\ell}_{i^*,t} \leq \frac{\ln K}{\eta} + \eta \sum_{t \in [T]} \hat{\ell}_{i^*,t}^2 + \eta \sum_{t \in [T]} \sum_{i \in [K]} \pi_{i,t} \hat{\ell}_{i,t}^2$$

where  $i^* = \arg \min_{i \in [K]} \sum_{t \in [T]} \ell_{i,t}$ .

*Proof.* Since  $\boldsymbol{\pi}_{T+1}$  is a valid probability distribution (Lemma 4.2), we have

$$\begin{aligned} 1 &\geq \pi_{i^*,T+1} = \pi_{i^*,T} \left( 1 - \eta \left( \hat{\ell}_{i^*,T} - \sum_{j \in [K]} \pi_{j,T} \hat{\ell}_{j,T} \right) \right) \\ &= \pi_{i^*,1} \prod_{t \in [T]} \left( 1 - \eta \left( \hat{\ell}_{i^*,t} - \sum_{j \in [K]} \pi_{j,t} \hat{\ell}_{j,t} \right) \right) \end{aligned}$$

Taking the logarithm for both sides, and using the fact that  $\pi_{i,1} = 1/K, \forall i \in [K]$ , we get

$$0 \geq -\ln K + \sum_{t \in [T]} \ln \left( 1 - \eta \left( \hat{\ell}_{i^*,t} - \sum_{j \in [K]} \pi_{j,t} \hat{\ell}_{j,t} \right) \right). \quad (4.4.2)$$

We next show that for all  $t \in [T]$  and any  $i \in [K]$   $\eta (\hat{\ell}_{i,t} - \sum_{j \in [K]} \pi_{j,t} \hat{\ell}_{j,t}) \leq 1/2$ . We distinguish two cases. First, if  $i \neq I_t$ , then the inequality holds since  $\hat{\ell}_{i,t} = 0$  and as a result the expression

becomes  $-\eta \cdot \pi_{I_t,t} \hat{\ell}_{I_t,t} \leq 0$ . Second, if  $i = I_t$ , then the expression becomes

$$\begin{aligned}
\eta \frac{\ell_{I_t,t}}{\pi_{I_t,t}} - \eta \pi_{I_t,t} \frac{\ell_{I_t,t}}{\pi_{I_t,t}} &= \eta \frac{\ell_{I_t,t}}{\pi_{I_t,t}} (1 - \pi_{I_t,t}) \\
&\leq \eta \frac{1}{\pi_{I_t,t}} && (\pi_{i,t} \geq 0, \ell_{i,t} \leq 1) \\
&\leq \eta \frac{K}{\gamma} && (\tilde{\pi}_{i,t} \geq \gamma/K, \text{ since } \pi_{i,t} \geq 0) \\
&\leq \frac{1}{2} && (\text{by definition})
\end{aligned}$$

We can now lower bound Equation (4.4.2) using the fact that for  $z \leq 1/2$  it holds that:  $\ln(1-z) \geq -z - z^2$  (Lemma B.2).

$$\begin{aligned}
0 &\geq -\ln K + \sum_{t \in [T]} \left[ -\eta \left( \hat{\ell}_{i^*,t} - \sum_{j \in [K]} \pi_{j,t} \hat{\ell}_{j,t} \right) \right] - \sum_{t \in [T]} \left[ \eta^2 \left( \hat{\ell}_{i^*,t} - \sum_{j \in [K]} \pi_{j,t} \hat{\ell}_{j,t} \right)^2 \right] \\
&\geq -\ln K - \eta \left[ \sum_{t \in [T]} \hat{\ell}_{i^*,t} - \sum_{t \in [T]} \sum_{j \in [K]} \pi_{j,t} \hat{\ell}_{j,t} \right] - \eta^2 \left[ \sum_{t \in [T]} \left( \hat{\ell}_{i^*,t} - \sum_{j \in [K]} \pi_{j,t} \hat{\ell}_{j,t} \right)^2 \right] \\
&\geq -\ln K - \eta \left[ \sum_{t \in [T]} \hat{\ell}_{i^*,t} - \sum_{t \in [T]} \sum_{j \in [K]} \pi_{j,t} \hat{\ell}_{j,t} \right] - \eta^2 \sum_{t \in [T]} \hat{\ell}_{i^*,t}^2 - \eta^2 \sum_{t \in [T]} \left( \sum_{j \in [K]} \pi_{j,t} \hat{\ell}_{j,t} \right)^2 \\
&\geq -\ln K - \eta \left[ \sum_{t \in [T]} \hat{\ell}_{i^*,t} - \sum_{t \in [T]} \sum_{j \in [K]} \pi_{j,t} \hat{\ell}_{j,t} \right] - \eta^2 \sum_{t \in [T]} \hat{\ell}_{i^*,t}^2 - \eta^2 \sum_{t \in [T]} \sum_{j \in [K]} \pi_{j,t} \hat{\ell}_{j,t}^2
\end{aligned}$$

where the second inequality uses the fact that for  $a, b$  non-negative,  $(a - b)^2 \leq a^2 + b^2$  and the last inequality uses Jensen's inequality for function  $f(x) = x^2$ . Rearranging the latter and dividing both sides by  $\eta$  gives the result.  $\blacksquare$

With that we can complete the proof of Theorem 4.2.

*Proof of Theorem 4.2.* It follows from incentive compatibility of WSU that an expert maximizes the expected value of  $\pi_{i,t+1}$  by minimizing the expected value of  $\hat{\ell}_{i,t}$ . From the definition of  $\hat{\ell}_{i,t}$ , it is easy to see that minimizing the expected value of  $\hat{\ell}_{i,t}$  is equivalent to minimizing the expected value of  $\ell_{i,t}$ . By properness of  $\ell$ , this is achieved by truthfully reporting  $p_{i,t} = b_{i,t}$ .

We now show the regret bound. Taking expectations with respect to the choice of expert at

round  $t$  for both sides of the equation in Lemma 4.4, we get

$$\begin{aligned} & \sum_{t \in [T]} \sum_{i \in [K]} \pi_{i,t} \mathbb{E}_{I_t \sim \tilde{\pi}_t} [\hat{\ell}_{i,t}] - \sum_{t \in [T]} \mathbb{E}_{I_t \sim \tilde{\pi}_t} [\hat{\ell}_{i^*,t}] \\ & \leq \eta \sum_{t \in [T]} \mathbb{E}_{I_t \sim \tilde{\pi}_t} [\hat{\ell}_{i^*,t}^2] + \frac{\ln K}{\eta} + \eta \sum_{t \in [T]} \sum_{i \in [K]} \pi_{i,t} \mathbb{E}_{I_t \sim \tilde{\pi}_t} [\hat{\ell}_{i,t}^2]. \end{aligned}$$

Using Lemma 4.3, this gives us

$$\begin{aligned} \sum_{t \in [T]} \sum_{i \in [K]} \pi_{i,t} \ell_{i,t} - \sum_{t \in [T]} \ell_{i^*,t} & \leq \eta \sum_{t \in [T]} \frac{1}{\tilde{\pi}_{i^*,t}} + \frac{\ln K}{\eta} + \eta \sum_{t \in [T]} \sum_{i \in [K]} \pi_{i,t} \frac{1}{\tilde{\pi}_{i,t}} \\ & \leq \eta \sum_{t \in [T]} \frac{K}{\gamma} + \frac{\ln K}{\eta} + 2\eta KT \\ & \leq \frac{\eta KT}{\gamma} + \frac{\ln K}{\eta} + 2\eta KT, \end{aligned}$$

where the second inequality uses the fact that  $\pi_{i,t} \leq 2\tilde{\pi}_{i,t}$ ,  $\forall i \in [K], t \in [T]$  since  $\gamma/K \geq 0$  and  $\gamma \leq 1/2$ . Next, we re-write  $\pi_{i,t} = \frac{\tilde{\pi}_{i,t} - \gamma/K}{1-\gamma}$ , yielding

$$\sum_{t \in [T]} \sum_{i \in [K]} \frac{\tilde{\pi}_{i,t} - \gamma}{1-\gamma} \ell_{i,t} - \sum_{t \in [T]} \ell_{i^*,t} \leq \frac{\eta KT}{\gamma} + \frac{\ln K}{\eta} + 2\eta KT.$$

Since  $1 - \gamma < 1$  and  $\ell(p_{i,t}, r_t) \leq 1$ , this can be relaxed to:

$$\sum_{t \in [T]} \sum_{i \in [K]} \tilde{\pi}_{i,t} \ell(p_{i,t}, r_t) - \sum_{t \in [T]} \ell(p_{i^*,t}, r_t) \leq \gamma T + \frac{\eta KT}{\gamma} + \frac{\ln K}{\eta} + 2\eta KT.$$

Making  $\gamma T = \eta KT/\gamma$  by setting  $\gamma = \sqrt{\eta K}$ , and  $\eta = \left(\frac{\ln K}{4K^{1/2}T}\right)^{2/3}$  we get the result.<sup>§</sup> ■

As in full information, a doubling trick can be applied if  $T$  is unknown (Appendix A.2.3).

We note that, unlike the full information setting in which WSU achieves the optimal regret bound for general loss functions, our regret bound in Theorem 4.2 is not as good as what can be achieved without incentive compatibility. Examining our analysis, one can see that if the loss of the best-fixed expert in hindsight is zero at each round, then the regret guarantee achieved by WSU-UX would be

---

<sup>§</sup>The last derivation requires that  $\eta \leq 1/K$ , which is true for large enough horizons  $T \geq K \ln K$ .

the same as EXP3, i.e.,  $\mathcal{O}(\sqrt{T \ln K})$ . Closing this gap via a tighter analysis of WSU-UX or via a new incentive-compatible algorithm is a compelling question for future work.

#### 4.5 FORWARD-LOOKING EXPERTS

So far we have assumed that the experts are myopic, aiming at time  $t$  to optimize their influence on the algorithm only at time  $t + 1$  with no regard for future rounds. It is natural to ask whether it is possible to design learning algorithms that are no regret while incentivizing truthful reports from forward-looking experts who care about their influence  $\pi_{i,t'}$  at all  $t' > t$ . Neither WSU nor WSU-UX achieve this goal; see Appendix A.3 for examples that illustrate why.<sup>¶</sup>

In order to derive an online learning algorithm that is incentive-compatible for forward-looking experts, we build on work by [Witkowski et al. \(2018\)](#), who studied a forecasting competition setting in which agents make predictions about a series of independent events, competing for a single prize. Unlike in our setting, their goal was to derive an incentive-compatible mechanism for choosing the winning agent; they are agnostic to how the elicited forecasts are aggregated. They defined a mechanism, Event-Lotteries Forecaster Selection Mechanism (ELF), in which, for every predicted event  $\tau$ , every agent  $i$  is assigned a probability of being the event winner based on the quality of their prediction. The winner of the competition is the agent who wins the most events.

We build on this idea to define an online learning algorithm, ELF-X, for the full information setting. Like WSU, ELF-X incorporates WSM payments, but in a different way. The distribution  $\pi_t$  at time  $t$  is defined as the distribution over experts output by the following randomized process:

1. At each round  $\tau \in [t]$ , pick agent  $i$  as the “winner”  $x_\tau$  with probability

$$\frac{1}{K} \left( 1 - \ell_{i,\tau} + \frac{1}{K} \sum_{j \in [K]} \ell_{j,\tau} \right).$$

2. Pick  $\arg \max_{i \in [K]} \sum_{\tau \in [t]} \mathbb{1}(x_\tau = i)$ , the expert who won the most events (tie break uniformly).

It can be shown by a similar argument to that of [Witkowski et al. \(2018\)](#) that ELF-X is incentive-compatible. The proof, along with a formal definition of incentive compatibility for forward-looking

---

<sup>¶</sup>It is worth noting that in these examples, an expert can gain only a negligible amount from misreporting; it is an open question whether WSU satisfies some notion of  $\epsilon$ -incentive compatibility.

experts, is in the appendix.

**Theorem 4.3: ELF-X**

ELF-X is incentive-compatible for forward-looking experts.

While proving that ELF-X is no-regret remains an open problem, in the following section, we present experimental results suggesting that its regret is sublinear in  $T$  in practice.

## 4.6 EXPERIMENTS

In this section, we empirically evaluate the performance of our proposed incentive-compatible algorithms, WSU and WSU-UX, compared with standard no-regret algorithms. We also evaluate the performance of ELF-X, which is incentive-compatible for non-myopic experts. Our code and the datasets we use are publicly available online.<sup>||</sup>

We ran each algorithm on publicly available datasets from a forecasting competition run by FiveThirtyEight<sup>\*\*</sup> in which users (henceforth called “forecasters”) make predictions about the outcomes of National Football League (NFL) games. Before each game, FiveThirtyEight releases information on the past performance of the two opposing teams, and forecasters provide probabilistic predictions about which team will win the game. FiveThirtyEight maintains a public leaderboard with the most accurate forecasters, updated after each game. The datasets for the 2018–2019 and 2019–2020 seasons each include all forecasters’ predictions, labeled with the forecaster’s unique id, information about the corresponding game, and the game’s outcome. Each NFL season has a total of 267 games, so in our setting,  $T = 267$ . For 2018–2019 (respectively, 2019–2020), while 15,702 (15,140) participated, only 302 (375) made predictions for every game. In order to reduce variance, for each value of  $K$ , we sampled 10 groups of  $K$  forecasters from the 302 (respectively, 375), and for each such group, ran each algorithm 50 times.

We evaluate performance using quadratic loss. We compare the cumulative loss of each algorithm against the cumulative loss of the best fixed forecaster in hindsight. For the full information

---

<sup>||</sup>Code: <https://github.com/charapod/noregr-and-ic>. Datasets: <https://github.com/fivethirtyeight/nfl-elo-game>

<sup>\*\*</sup><https://projects.fivethirtyeight.com/2019-nfl-forecasting-game/>

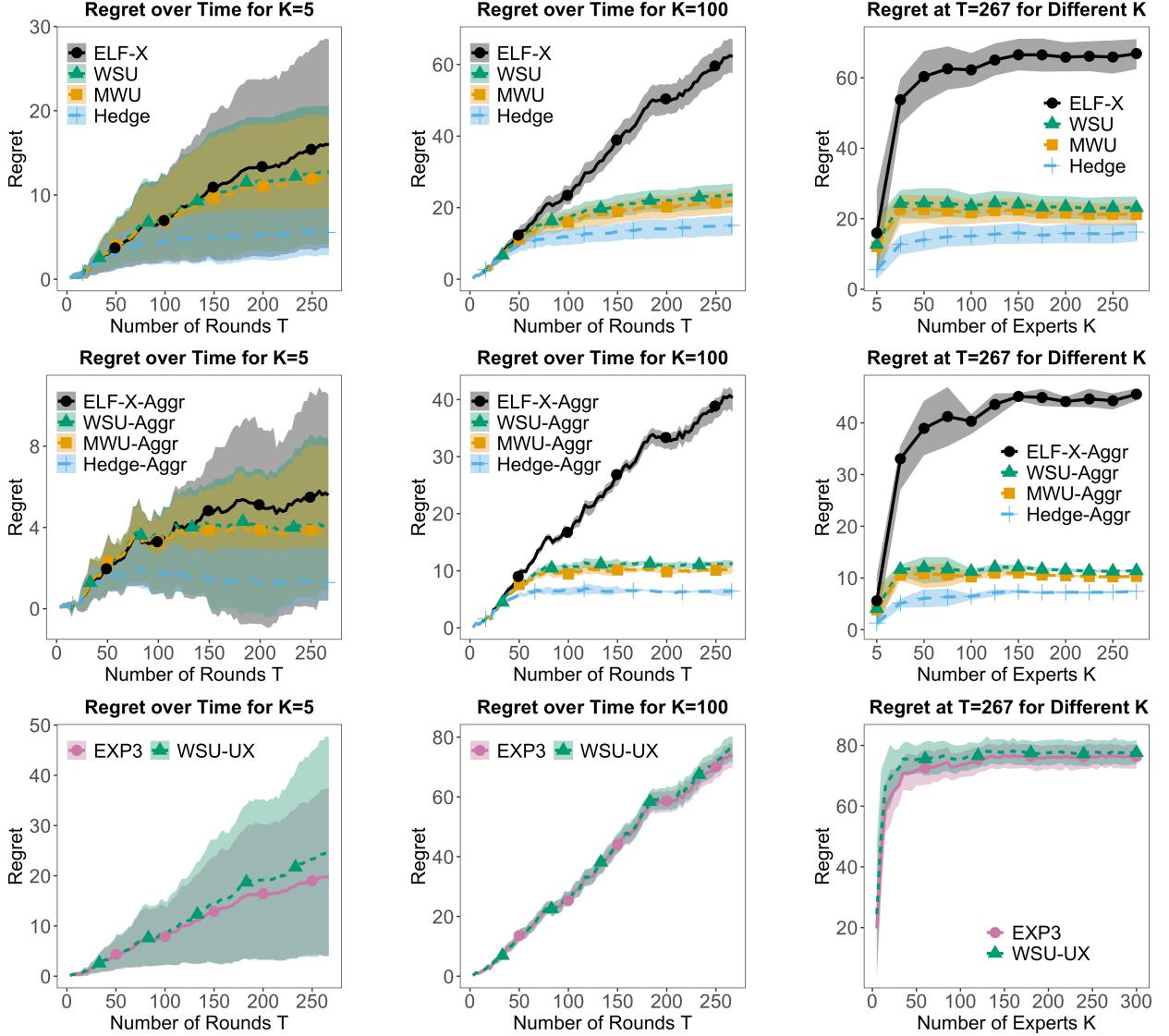


Figure 4.1: Experiments on the 2018–2019 FiveThirtyEight NFL dataset. Top: Full-information setting with  $\bar{p}_t$  the prediction of a single expert chosen according to  $\pi_t$ . Middle: Full-information setting with  $\bar{p}_t = \sum_{i \in [K]} \pi_{i,t} p_{i,t}$ . Bottom: Partial information setting.

setting, we compare WSU and ELF-X against Hedge, which achieves optimal regret guarantees since the quadratic loss is exp-concave, and MWU, which is more similar in form to WSU, in order to evaluate whether anything is lost in terms of regret when incentive compatibility is achieved. For the partial information setting, we compare WSU-UX against EXP3. For each full information algorithm, we run both the variant in which a single expert is selected at each round and the variant in which the learner outputs a weighted combination of expert reports (labeled  $^*$ -Aggr). For ELF-X-Aggr, since  $\pi_t$  cannot be computed in closed form, we approximate it via sampling.

We present the results of our experiments on the 2018–2019 dataset in Figure 4.1; the results on the 2019–2020 dataset are in Appendix A.4.1, and exhibit similar trends. We note that lines correspond to average regret (across all samples of experts and all repetitions), while the error bands correspond to the 20th and 80th percentiles; this leads to much smaller error bands for larger values of  $K$  since the specific sampling of experts has less influence on regret for large  $K$ .

Validating our theoretical results, WSU performs almost identically (in terms of both  $K$  and  $T$ ) to MWU when fed the same set of reports—this, of course, does not take into account that MWU is not incentive compatible and may lead to misreports in practice, potentially degrading predictions. Interestingly, we also see that WSU-Aggr performs almost identically to MWU-Aggr. This suggests that the performance of WSU-Aggr is considerably better than the bound in Chapter 4.3 implies. It is an interesting open question to see whether better regret guarantees can be proved for WSU-Aggr, perhaps with respect to the best fixed distribution of experts. As expected, both WSU and MWU are outperformed by Hedge, which achieves optimal regret bounds for squared loss but offers no incentive guarantees.

ELF-X exhibits diminishing regret on this dataset, particularly for  $K = 5$ . However, ELF-X and ELF-X-Aggr perform worse than WSU and WSU-Aggr respectively when fed the same input, particularly when  $K$  is large. Although ELF-X obtains a stronger incentive guarantee, the violations of incentive compatibility for forward-looking experts exhibited by WSU are very small in our examples. In practice, we expect that WSU is a better choice to ELF-X when balancing regret and incentive properties, even for forward-looking experts.

For the bandit setting, quite encouragingly, we see that the performance of WSU-UX is only slightly worse than that of EXP3, and appears significantly better than the  $\mathcal{O}(T^{2/3})$  regret bound in Chapter 4.4 would suggest. This could be a byproduct of our analysis not being tight, and it remains an open question whether this bound can be improved.

The experiments presented in this section focus on settings with relatively small horizons  $T$  since an NFL season has only 267 matches. In Appendix A.4.2, we present our results (also validating our theoretical results) for Monte Carlo simulations for larger horizons.

#### 4.7 DISCUSSION AND OPEN QUESTIONS

In this chapter, we studied the problem of online learning with strategic experts. We introduced algorithms that are simultaneously no-regret and incentive-compatible, and assessed their performance experimentally on data from FiveThirtyEight. Several open questions arise. In the full-information setting, there is the question of whether an incentive-compatible algorithm exists with better regret bounds for the special case of exp-concave bounded proper loss functions. For the bandit setting, there is the question of whether there exist incentive-compatible algorithms that bridge the gap between the regret of WSU-UX and that of EXP3, and whether a better regret guarantee could be proved for WSU-UX via a tighter analysis. There is also the question of whether ELF-X is indeed no-regret, as our experimental results might suggest. More broadly, the most important research question that we believe needs to be addressed in online learning from strategic agents is the quantification of the trade-off between incentive-incompatibility and standard learning guarantees and how to balance these in practice.

## **Part II**

# **Adaptation to Strategic Individuals**

# 5

## Learning Strategy-Aware Linear Classifiers

### 5.1 CHAPTER OVERVIEW

In settings where incentive-compatibility, the holy grail of incentive alignment, is unattainable, the institution can achieve a form of robustness to strategizing through *adapting* to the agents incentives. In this chapter, we explore this direction further for the task of classification.

Specifically, motivated by the problem of classifying spam, we focus on the problem of learning an unknown *linear* classifier, when the data come in an *online* fashion from *strategic* agents, who can alter their feature vectors to *game* the classifier. We model the interplay between the learner and the strategic agents<sup>\*</sup> as a repeated *Stackelberg game* over  $T$  rounds. As a reminder, in a repeated Stackelberg game, the learner *commits* to an action, and then, the agent best-responds to it, i.e.,

---

<sup>\*</sup>We refer to the learner as a female (she/her/hers) and to the agents as male (he/his/him).

reports something that maximizes his underlying utility. The learner's goal is to minimize her *Stackelberg regret*, which is the difference between her cumulative loss and the cumulative loss of her best-fixed action in hindsight, *had she allowed the agent to best-respond to it*.

We study a general model of learner-agent interaction in strategic classification settings. Specifically, in our model, the agents' true datapoint remains hidden from the learner, the agents can mis-report within a ball of radius  $\delta$  of their true datapoint, and the learner measures her performance using the *binary loss*. We say that our agents are  $\delta$ -*bounded, myopically rational* ( $\delta$ -BMR), when reporting their datapoints. Our model departs significantly from the smooth utility and loss functions assumed for the agent and the learner respectively that have been used previously in the literature for strategic classification (see Chapter 5.2 for further discussion).

We prove that in strategic classification settings against  $\delta$ -BMR agents *simultaneously* achieving sublinear external and Stackelberg regret is in general impossible. To be more precise, we show that external and Stackelberg regret are *strongly incompatible*, i.e., there exists a sequence of datapoints where any algorithm achieving sublinear external regret must incur linear Stackelberg regret and vice versa. To obtain these incompatibility results, we present a formal framework, which may be of independent interest. The implication of this strong incompatibility result is that applying standard no-external regret algorithms for our strategic classification model may be unhelpful. Not only this but due to agents being  $\delta$ -BMR, the learner's loss function is not even Lipschitz in the worst case. As a result, new algorithms and techniques are required to tackle this problem.

Taking advantage of the structure of the responses of  $\delta$ -BMR agents in Chapter 5.4, we propose GRINDER, an adaptive discretization algorithm, which uses access to an oracle. GRINDER works on the learner's dual space which provides information about various *regions* of the learner's action space, *despite never observing the agent's true datapoint*. These regions (polytopes) relate to the partitions that GRINDER creates. We deal with the learner's action space being *continuous* (i.e., containing infinite actions), we use the fact that *all actions* within a polytope share the same *history* of estimated losses. So, passing information down to a recently partitioned polytope becomes a simple volume reweighting. Continuous action spaces come with a lot of peculiarities and are not usually covered in the literature. For that, we present a variant of the standard EXP3 algorithm that takes advantage of the polytope partitioning process. To the best of our knowledge, GRINDER's is

the first adaptive discretization algorithm that assumes no stochasticity on the loss function for the adaptive discretization.

We also prove that the regret guarantees of GRINDER remain *order unchanged* when the learner is given access to a *noisy* oracle, accommodating more settings in practice (Chapter 5.4). To conclude our theoretical results, in Chapter 5.6, we show nearly matching lower bounds for strategic classification against  $\delta$ -BMR agents, thus establishing the near optimality of GRINDER.

Complementing our theoretical analysis, in Chapter 5.5, we provide simulations implementing GRINDER both for continuous and discrete action spaces, and using both an accurate and an approximation oracle. Further, the approximation oracles in these experiments are easy to build from past data and are computationally efficient.

### 5.1.1 RELATED WORK

The results of this chapter are primarily related to the literature on *learning using data from strategic sources* and more specifically, what has come to be known as the “robustness” perspective thereof (e.g., (Cummings et al., 2015, Cai et al., 2015, Perote and Perote-Peña, 2004, Dekel et al., 2010, Chen et al., 2018, Ben-Porat and Tennenholz, 2017, 2018, Liu and Chawla, 2009, Brückner and Scheffer, 2011, Meir et al., 2010, 2011, 2012)). An extensive discussion of the related work on this perspective can be found in Chapter 3. Specifically, this chapter focuses on learning in strategic classification. The offline model of this problem was originally introduced by Hardt et al. (2016), while the online version is due to Dong et al. (2018). Similar to our model, in (Hardt et al., 2016, Dong et al., 2018) the ground truth label remains *unchanged* even *after* the agent’s manipulation. The work of Dong et al. (2018) is orthogonal to the present chapter in one key aspect: they find the appropriate conditions which can guarantee that the best-response of an agent, written as a function of the learner’s action, is concave. As a result, in their model the learner’s loss function becomes *convex* and well known online convex optimization algorithms can be applied (e.g., (Flaxman et al., 2005, Bubeck et al., 2017)) in conjunction with the mixture feedback that the learner receives. The foundation of this chapter, however, is settings with less smooth utility and loss functions for the agents and the learner respectively, where incompatibility issues arise.

Closer to the model studied in this chapter is the work of Ahmadi et al. (2021). Specifically, they

consider agents who are a subset of the  $\delta$ -BMR agents we define in this chapter. They also consider the setting where the original datapoints (i.e., prior to strategizing) are separable by a margin. As a result, they do not fall in the lower bound regime and are able to provide much better regret bounds based on modifying the well-known Perceptron algorithm. More distantly related to this chapter is the work of [Jagadeesan et al. \(2021\)](#) which was inspired by the instability that is often observed in strategic classification settings that follow standard models from Microeconomics for the agents' utilities. For that, they propose alternative microfoundations based on a set of desiderata and obtain robust solutions, which also impose a lower social burden at optimality.

There has also been recent interest in strategic classification settings where the agents invest effort and manage to alter their labels ([Bechavod et al., 2021](#), [Shavit et al., 2020](#), [Perdomo et al., 2020](#)). These models are especially applicable in cases where in order to alter their feature vector  $x_t$  (e.g., qualifications for getting in college) the agents have to *improve* their ground truth label (e.g., actually *try* to become a better candidate ([Ustun et al., 2019](#))). We refer the interested reader to Chapter 9 for a deep dive on issues pertaining to how information discrepancy in strategic classification disparately affects the effort exertion of different subpopulations. In contrast, in the present chapter we focus on a single, homogeneous population, and we think of the misreports as "manipulations" that aim at gaming the system without altering.

The results of this chapter are also related to learning in Stackelberg Security Games (SSGs) ([Letchford et al., 2009](#), [Marecki et al., 2012](#), [Blum et al., 2014](#)) and especially, the work of [Balcan et al. \(2015\)](#), who study information theoretic sublinear Stackelberg regret algorithms for the learner. That said the formal definition of Stackelberg regret was only later introduced by [Dong et al. \(2018\)](#). In SSGs, all utilities are linear, a property not present in strategic classification against  $\delta$ -BMR agents.

From the online learning literature, this chapter is related to learning with *partial/bandit* feedback (see ([Bubeck and Cesa-Bianchi, 2012](#), [Slivkins, 2019](#), [Lattimore and Szepesvári, 2020](#))) and *graph-structured* feedback ([Alon et al., 2015](#), [Cohen et al., 2016](#)). Adaptive discretization algorithms were studied for stochastic Lipschitz settings in ([Kleinberg et al., 2008b](#), [Bubeck et al., 2008](#)), but in learning against  $\delta$ -BMR agents, the loss is neither stochastic nor Lipschitz. After the original publication of the results of the present chapter, [Podimata and Slivkins \(2021\)](#) created the first, general adaptive discretization algorithm for continuous Lipschitz functions. We discuss this and all other

works related to adaptive discretization of Lipschitz functions in Chapter 6.

Finally, strong incompatibility between two different regret notions (specifically the *external* and the *policy* regret) was also studied in (Arora et al., 2018). One of the key differences between Stackelberg and policy regret is the horizon of which the strategizing of the player occurs in each one. As a result, the results for the policy regret do not translate to our setting.

## 5.2 MODEL AND PRELIMINARIES

We use the spam email application as a running example to setup our model. Each agent has an email that is either a spam or a non-spam. Given a classifier, the agent can alter his email to a certain degree in order to bypass the email filter and have his email be classified as non-spam. Such manipulation is costly. Each agent chooses a manipulation to maximize his overall utility.

**Interaction Protocol.** Let  $d \in \mathbb{N}$  denote the dimension of the problem and  $\mathcal{A} \subseteq [-1, +1]^{d+1}$  the learner's action space<sup>†</sup>. Actions  $\alpha \in \mathcal{A}$  correspond to hyperplanes, written in terms of their normal vectors, and we assume that the  $(d+1)$ -th coordinate encodes information about the intercept. Let  $\mathcal{X} \subseteq ([0, 1]^d, 1)$  be the feature vector space, where by  $([0, 1]^d, 1)$  we denote the set of all  $(d+1)$ -dimensional vectors with values in  $[0, 1]$  in the first  $d$  dimensions and value 1 in the  $(d+1)$ -th. Each feature vector has an associated label  $y \in \mathcal{Y} = \{-1, +1\}$ . Formally, the interaction protocol at round  $t$  is given in Protocol 5.1, where by  $\sigma_t$  we denote the tuple (feature vector, label).

---

### Protocol 5.1: Learner-Agent Interaction at Round $t$

---

- 1 Nature adversarially selects feature vector  $\mathbf{x}_t \in \mathcal{X} \subseteq ([0, 1]^d, 1)$ . // agent's original email
  - 2 The learner chooses action  $\alpha_t \in \mathcal{A}$ , and commits to it. // learner's linear classifier
  - 3 Agent observes  $\alpha_t$  and  $\sigma_t = (\mathbf{x}_t, y_t)$ , where  $y_t \in \mathcal{Y}$ . //  $y_t = +1$ , if non-spam originally
  - 4 Agent reports feature vector  $\mathbf{r}_t(\alpha_t, \sigma_t) \in \mathcal{X}$  (potentially,  $\mathbf{r}_t(\alpha_t, \sigma_t) \neq \mathbf{x}_t$ ). // altered email
  - 5 The learner observes  $(\mathbf{r}_t(\alpha_t, \sigma_t), \hat{y}_t)$ , where  $\hat{y}_t \in \mathcal{Y}$  is the label of  $\mathbf{r}_t(\alpha_t, \sigma_t)$ , and incurs binary loss  $\ell(\alpha_t, \mathbf{r}_t(\alpha_t, \sigma_t), \hat{y}_t) = \mathbb{1}\{\text{sgn}(\hat{y}_t \cdot \langle \alpha_t, \mathbf{r}_t(\alpha_t, \sigma_t) \rangle) = -1\}$ . // loss on altered email
- 

**Agents' Behavior:  $\delta$ -Bounded Myopically Rational.** Drawing intuition from the email spam example, we focus on agents who can alter their feature vector *up to an extent* in order to make their email fool the classifier, and, if successful, they gain some value. The agent's reported feature vector

---

<sup>†</sup>This is wlog, as the normal vector of any hyperplane can be normalized to lie in  $[-1, 1]^{d+1}$ .

$\mathbf{r}_t(\alpha_t, \sigma_t)$  is the solution to the following constrained optimization problem<sup>†</sup>:

$$\mathbf{r}_t(\alpha_t, \sigma_t) = \arg \max_{\|\mathbf{z} - \mathbf{x}_t\| \leq \delta} u_t(\alpha_t, \mathbf{z}, \sigma_t)$$

where  $u_t(\cdot, \cdot, \cdot)$  is the agent's underlying utility function, *which is unknown to the learner*. In words, in choosing what to report, the agents are *myopic* (i.e., focus only on the current round  $t$ ), *rational* (i.e., maximize their utility), and *bounded* (i.e., misreport in a ball of radius  $\delta$  around  $\mathbf{x}_t$ ). In such settings (e.g., email spam example), the agents derive no value if, by altering their feature vector from  $\mathbf{x}_t$  to  $\mathbf{r}_t(\alpha_t)$ , they also change  $y_t$ . Indeed, a spammer ( $y_t = -1$ ) wishes to fool the learner's classifier, without actually having to change their email to be a non-spam one ( $\hat{y}_t = +1$ ). Since the agents are *rational* this means that the observed label by the learner is  $\hat{y}_t = y_t$  and we only use notation  $y_t$  for the rest of the chapter. We call such agents  $\delta$ -Bounded Myopically Rational ( $\delta$ -BMR). Note that if the agents were *adversarial*, they would report  $\mathbf{r}(\alpha_t) = \arg \max_{\|\mathbf{z} - \mathbf{x}_t\| \leq \delta} \ell(\alpha, \mathbf{z}, y_t)$ .

Note that  $\delta$ -BMR agents include (but are not limited to!) a broad class of rational agents for strategic classification, like for example agents whose utility is defined as:

$$u_t(\alpha_t, \mathbf{r}_t(\alpha_t, \sigma_t), \sigma_t) = \delta' \cdot \mathbb{1}\{\text{sgn}(\langle \alpha_t, \mathbf{r}_t(\alpha_t, \sigma_t) \rangle) = +1\} - \|\mathbf{x}_t - \mathbf{r}_t(\alpha_t, \sigma_t)\|_2 \quad (5.2.1)$$

where  $\text{sgn}(x) = +1$  if  $x \geq 0$  and  $\text{sgn}(x) = -1$  otherwise. According to the utility presented in Eq. (5.2.1), the agents get a value of  $\delta'$  if he gets labeled as  $+1$  and incurs a cost (i.e., time/resources spent for altering the original  $\mathbf{x}_t$ ) that is a metric. For this case, we have that  $\delta \leq \delta'$ .

### 5.3 STACKELBERG VERSUS EXTERNAL REGRET

For what follows, let  $\{\alpha_t\}_{t=1}^T$  be the sequence of the learner's actions in a repeated Stackelberg game. Appendices B.1.1 and B.1.2 include detailed discussions around external and Stackelberg regret for learning in Stackelberg games.

**Definition 5.1** (External).  $R(T) = \sum_{t \in [T]} \ell(\alpha_t, \mathbf{r}_t(\alpha_t), y_t) - \min_{\alpha_E^* \in \mathcal{A}} \sum_{t \in [T]} \ell(\alpha_E^*, \mathbf{r}_t(\alpha_t), y_t)$ .

The external regret compares the cumulative loss from  $\{\alpha_t\}_{t \in [T]}$  to the cumulative loss incurred

---

<sup>†</sup>For simplicity, we denote  $\mathbf{r}_t(\alpha_t) = \mathbf{r}_t(\alpha_t, \sigma_t)$  when clear from context.

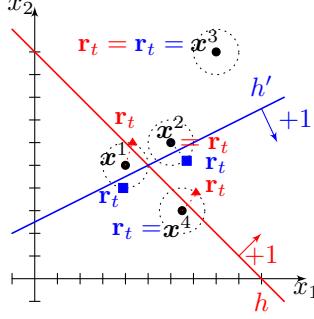


Figure 5.1: Sketch of strong incompatibility example. Black dots denote true feature vectors. Axes  $x_1, x_2$  correspond to the two features. Dotted circles correspond to the  $\delta$ -bounded interval inside which agents can misreport. Blue squares correspond to misreports against action  $h'$  and red triangles to misreports against action  $h$ .

by the best-fixed action in hindsight, had the learner *not* allowed the agents to best respond.

**Definition 5.2** (Stackelberg).  $\mathcal{R}(T) = \sum_{t \in [T]} \ell(\alpha_t, \mathbf{r}_t(\alpha_t), y_t) - \min_{\alpha^* \in \mathcal{A}} \sum_{t \in [T]} \ell(\alpha^*, \mathbf{r}_t(\alpha^*), y_t)$ .

Stackelberg regret (Balcan et al., 2015, Dong et al., 2018) compares the loss from  $\{\alpha_t\}_{t \in [T]}$  to the loss from the best-fixed action in hindsight, *had the learner allowed the agents to best respond*. Appendices B.1.1 and B.1.2 include detailed discussions around external and Stackelberg regret for learning in Stackelberg games.

### Theorem 5.1: External and Stackelberg Regret Strong Incompatibility

There exists a repeated strategic classification setting against a  $\delta$ -BMR agent, where *every* action sequence from the learner with *sublinear* external regret incurs *linear* Stackelberg regret, and vice versa, i.e., *every* action sequence for the learner with *sublinear* Stackelberg regret incurs *linear* external regret.

*Proof.* Let us define an action space  $\mathcal{A} = \{h, h'\}$  such that  $h = (1, 1, -1)$  and  $h' = (0.5, -1, 0.25)$ , and let  $\delta = 0.1$ . The environment draws feature vectors  $\mathbf{x}^1 = (0.4, 0.5), \mathbf{x}^2 = (0.6, 0.6), \mathbf{x}^3 = (0.8, 0.9), \mathbf{x}^4 = (0.65, 0.3)$  with probabilities  $p^1 = 0.05, p^2 = 0.15, p^3 = 0.05, p^4 = 0.75$  respectively, and with labels  $y^1 = -1, y^2 = -1, y^3 = +1, y^4 = +1$ . Figure 5.1 provides a pictorial depiction of the example, along with the best responses of the agents for each action. We first explain the values that the loss function takes according to the best-responses of the agents and the feature vectors drawn by nature.

- ( $\mathbb{E}[\ell(h, \mathbf{r}_t(h), y_t)]$ ) When the learner plays  $h$  against agent's responses  $\mathbf{r}_t(h)$ , she makes a mistake in her prediction every time that the environment drew  $x_1$  or  $x_2$  for the agent. This is because for  $x^1$ , the agent can misreport and fool  $h$ . For  $x^2$ , the agent does not need to misreport; hyperplane  $h$  classifies it as  $+1$  erroneously already. For  $x^4$  the agent can misreport and get correctly classified and for  $x^3$  the hyperplane is correct all by itself. Hence:

$$\mathbb{E}[\ell(h, \mathbf{r}_t(h), y_t)] = \Pr[\text{nature draws } x^1 \text{ or } x^2] = p^1 + p^2 = 0.2$$

- ( $\mathbb{E}[\ell(h', \mathbf{r}_t(h'), y_t)]$ ) When the learner plays  $h'$  against agent's responses  $\mathbf{r}_t(h')$ , she makes a mistake in her prediction every time that the environment drew  $x_1$  or  $x_2$  or  $x_3$  for the agent. This is because for both  $x^1$  and  $x^2$  the agent could misreport and fool the hyperplane and for  $x^3$  the hyperplane classifies it incorrectly, but there is nothing that the learner can do to change it (due to  $\delta$ -boundedness). For  $x^4$  the hyperplane classifies the point correctly, without the need of misreport from the agent. Hence:

$$\mathbb{E}[\ell(h', \mathbf{r}_t(h'), y_t)] = \Pr[\text{nature draws } x^1 \text{ or } x^2 \text{ or } x^3] = p^1 + p^2 + p^3 = 0.25$$

- ( $\mathbb{E}[\ell(h, \mathbf{r}_t(h'), y_t)]$ ) When the learner plays  $h$  against agent's responses  $\mathbf{r}_t(h')$ , she makes a mistake in her prediction every time that the environment drew  $x_2$  or  $x_4$  for the agent, i.e.,

$$\mathbb{E}[\ell(h, \mathbf{r}_t(h'), y_t)] = \Pr[\text{nature draws } x^2 \text{ or } x^4] = p^2 + p^4 = 0.9$$

- ( $\mathbb{E}[\ell(h', \mathbf{r}_t(h), y_t)]$ ) When the learner plays  $h'$  against agent's responses  $\mathbf{r}_t(h)$ , she makes a mistake in her prediction every time that the environment drew  $x_3$  for the agent, i.e.,

$$\mathbb{E}[\ell(h', \mathbf{r}_t(h), y_t)] = \Pr[\text{nature draws } x^3] = p^3 = 0.05$$

We now prove that *any* sequence with sublinear Stackelberg regret will have linear external regret. Observe that for the Stackelberg regret, the best fixed action in hindsight is action  $h$ , with cumulative loss  $0.2T$ . Therefore, any action sequence that yields sublinear Stackelberg regret must

have cumulative loss  $0.2T + o(T)$ , meaning that action  $h'$  is played at most  $o(T)$  times, while action  $h$  is played at least  $T - o(T)$  times. Given this, we proceed by identifying the best fixed action for the *external* regret in any action such sequence  $\{\alpha_t\}_{t=1}^T$ . For that, we compute the loss that any of the actions in  $\mathcal{A}$  would incur, had they been the fixed action for sequence  $\{\alpha_t\}_{t=1}^T$ .

Assume that action  $h$  was the fixed action in hindsight for the sequence  $\{\alpha_t\}_{t=1}^T$ . Then, the cumulative loss incurred by playing  $h$  constantly for  $T$  rounds, denoted by  $\sum_{t=1}^T \ell(h, \mathbf{r}_t(\alpha_t), y_t)$  is:

$$\underbrace{0.2(T - o(T))}_{\substack{\text{loss incurred when playing} \\ h \text{ against } \mathbf{r}_t(h)}} + \underbrace{0.9o(T)}_{\substack{\text{loss incurred when playing} \\ h \text{ against } \mathbf{r}_t(h')}}.$$

Assume that action  $h'$  was the fixed action in hindsight for the aforementioned action sequence. Then, the cumulative loss incurred by playing  $h'$ , denoted by  $\sum_{t=1}^T \ell(h', \mathbf{r}_t(\alpha_t), y_t)$  is equal to

$$\underbrace{0.05(T - o(T))}_{\substack{\text{loss incurred when playing} \\ h' \text{ against } \mathbf{r}_t(h)}} + \underbrace{0.25o(T)}_{\substack{\text{loss incurred when playing} \\ h' \text{ against } \mathbf{r}_t(h')}}$$

Hence, we have that the best fixed action in hindsight for the external regret for the sequence  $\{\alpha_t\}_{t=1}^T$  is action  $h'$ . This means, however, that for the sequence  $\{\alpha_t\}_{t=1}^T$ , which guaranteed sublinear Stackelberg regret, the external regret is *linear* in  $T$ :

$$R(T) \geq 0.2T - 0.05T \geq 0.15T$$

Moving forward, we prove that *any* action sequence with sublinear external regret will have linear Stackelberg regret. Since we previously proved that any action sequence  $\{\alpha_t\}_{t=1}^T$  with sublinear Stackelberg regret plays at least  $T - o(T)$  times action  $h$  and this resulted in having linear external regret, we only need to consider sequences where action  $h'$  is played  $T - o(T)$  times, while action  $h$  is played for  $o(T)$  times. For any such action sequence, it suffices to show that the external regret will be sublinear, since for any such sequence the Stackelberg regret will be linear:

$$\mathcal{R}(T) = 0.2o(T) + 0.25 \cdot (T - o(T)) - 0.2T \geq 0.05T.$$

Similarly to the analysis above, we distinguish the following cases. Assume that action  $h$  was the fixed action in hindsight for  $\{\alpha_t\}_{t=1}^T$ . Then, the cumulative loss incurred by playing  $h$  is

$$\sum_{t=1}^T \ell(h, \mathbf{r}_t(\alpha_t), y_t) = 0.2o(T) + 0.9(T - o(T)).$$

Assume that action  $h'$  was the fixed action in hindsight for the aforementioned action sequence. Then, the cumulative loss incurred by playing  $h'$  is

$$\sum_{t=1}^T \ell(h', \mathbf{r}_t(\alpha_t), y_t) = 0.05o(T) + 0.25(T - o(T)).$$

As a result, the best fixed action in hindsight for the Stackelberg regret would be action  $h'$ , yielding external regret  $o(T)$ , i.e., sublinear. This concludes our proof. ■

#### 5.4 THE GRINDER ALGORITHM

In this section, we present **GRINDER**, an algorithm that learns to adaptively partition the learner's action space according to the agent's responses. To help with the dense notation, we include notation tables in Appendix B.2.1. Formally, we prove the following *data-dependent* guarantee. Note that our actual bound is *tighter*, but harder to interpret without analyzing the algorithm.

##### Theorem 5.2: Regret of **GRINDER** Algorithm

Given a finite horizon  $T$  the Stackelberg regret incurred by **GRINDER** (Algo. 5.2) is:

$$\mathcal{R}(T) \leq \mathcal{O}\left(\sqrt{T \cdot \log\left(T \cdot \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})}\right) \cdot \log\left(\frac{\lambda(\mathcal{A})}{\lambda(\underline{p})}\right)}\right)$$

where by  $\lambda(A)$  we denote the Lebesgue measure of any measurable space  $A$ , and by  $\underline{p}$  we denote the polytope with the smallest Lebesgue measure induced by **GRINDER** after  $T$  rounds.

**Inferring  $\ell(\alpha, \mathbf{r}_t(\alpha), y_t)$  without Observing  $x_t$ .** We think of the learner's and the agent's spaces as dual ones (Figure 5.2), and focus on the agent's action space first. Since agents are  $\delta$ -BMR, then, for feature vector  $x_t$  the agent can only misreport within the ball of radius  $\delta$  centered around  $x_t$ ,

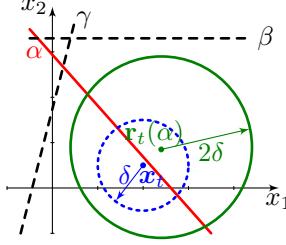


Figure 5.2: Agent’s action space in 2d.  $\mathbf{x}_t$  is the agent’s true feature vector, and  $\mathbf{r}_t$  is his misreport against  $\alpha$ .  $\alpha, \beta, \gamma$  are the learner’s potential actions.

denoted by  $\mathcal{B}_\delta(\mathbf{x}_t)$  (e.g., purple dotted circle in Figure 5.2). Since the learner observes  $\mathbf{r}_t(\alpha)$  and knows that the agent misreports in a ball of radius  $\delta$  around  $\mathbf{x}_t$  (which remains unknown to her), she knows that in the worst case the agent’s  $\mathbf{x}_t$  is found within the ball of radius  $\delta$  centered at  $\mathbf{r}_t(\alpha)$ . This means that the set of all of the agent’s possible misreports against any committed action  $\alpha'$  from the learner  $\mathbf{r}_t(\alpha')$  is the *augmented*  $2\delta$  ball (e.g., green solid circle). Since  $y_t$  is also observed by the learner, she can thus infer her loss  $\ell(\alpha', \mathbf{r}_t(\alpha'), y_t)$  for *any* action  $\alpha'$  that has  $\mathcal{B}_{2\delta}(\mathbf{r}_t(\alpha))$  *fully* in one of its halfspaces (e.g., actions  $\beta, \gamma$  in Figure 5.2).

In the learner’s action space, actions  $\alpha, \beta, \gamma$  are multidimensional *points*, and this has a nice mathematical translation. An action  $\gamma$  has  $\mathcal{B}_{2\delta}(\mathbf{r}_t(\alpha))$  fully in one of its halfspaces, if its distance from  $\mathbf{r}_t(\alpha)$  is *more than*  $2\delta$ . Alternatively for actions  $h \in \mathcal{A}$  such that:

$$\frac{|\langle h, \mathbf{r}_t(\alpha) \rangle|}{\|h\|_2} \leq 2\delta \Leftrightarrow |\langle h, \mathbf{r}_t(\alpha) \rangle| \leq 4\sqrt{d}\delta$$

where the last inequality comes from the fact that  $\mathcal{A} \subseteq [-1, 1]^{d+1}$ , the learner cannot infer  $\ell(h, \mathbf{r}_t(h))$ . But for all other actions  $\gamma$  in  $\mathcal{A}$ , the learner can compute her loss  $\ell(h, \mathbf{r}_t(h))$  *precisely*! From that, we derive that the learner can partition her action space into the following *polytopes*: upper polytopes  $\mathcal{P}_t^u$ , containing actions  $w \in \mathcal{A}$  such that  $\langle w, \mathbf{r}_t(\alpha) \rangle \geq 4\sqrt{d}\delta$  and lower polytopes  $\mathcal{P}_t^l$ , containing actions  $w' \in \mathcal{A}$  such that  $\langle w', \mathbf{r}_t(\alpha) \rangle \leq -4\sqrt{d}\delta$ . The distinction into the two sets is helpful as one of them always assigns label +1 to the agent’s best-response, and the other always assigns label -1. The sizes of  $\mathcal{P}_t^u$  and  $\mathcal{P}_t^l$  depend on  $\delta$  and  $\{\mathbf{x}_t\}_{t=1}^T$ , but we omit these for the ease of notation.

**Algorithm Overview.** At each round  $t$ , GRINDER (Algorithm 5.2) maintains a sequence of nested polytopes  $\mathcal{P}_t$ , with  $\mathcal{P}_1 = \{\mathcal{A}\}$  and decides which action  $\alpha_t$  to play according to a two-stage sam-

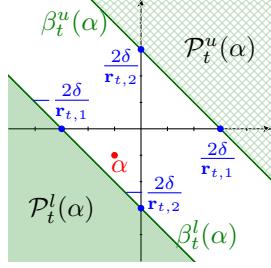


Figure 5.3: Polytope partitioning in 2d.  $\mathbf{r}_{t,1}, \mathbf{r}_{t,2}$  are the  $x_1$  and  $x_2$  coordinates of  $\mathbf{r}_t(\alpha)$ .

pling process. We denote the resulting distribution by  $\mathcal{D}_t$ , and by  $\Pr_t$  and  $f_{\mathcal{A}_t}(\alpha)$  the associated probability and probability density function.

After the learner observes  $\mathbf{r}_t(\alpha_t)$ , she computes two hyperplanes with the same normal vector  $(\mathbf{r}_t(\alpha_t))$  and symmetric intercepts  $(\pm 4\sqrt{d}\delta)$ . These *boundary* hyperplanes are defined as:

$$\beta_t^u(\alpha_t) : \forall \mathbf{w} \in \mathcal{A}, \langle \mathbf{w}, \mathbf{r}_t(\alpha_t) \rangle = 4\sqrt{d}\delta$$

$$\beta_t^l(\alpha_t) : \forall \mathbf{w} \in \mathcal{A}, \langle \mathbf{w}, \mathbf{r}_t(\alpha_t) \rangle = -4\sqrt{d}\delta$$

and they split the learner's action space into three *regions*; one for which  $\forall \mathbf{w} : \langle \mathbf{w}, \mathbf{r}_t(\alpha_t) \rangle \geq 4\sqrt{d}\delta$ , one for which  $\langle \mathbf{w}, \mathbf{r}_t(\alpha_t) \rangle \leq -4\sqrt{d}\delta$  and one for which  $|\langle \mathbf{w}, \mathbf{r}_t(\alpha_t) \rangle| \leq 4\sqrt{d}\delta$  (see Figure 5.3).

Let  $H^+(\beta), H^-(\beta)$  denote the closed positive and negative halfspaces defined by hyperplane  $\beta$  for intercept  $4\sqrt{d}\delta$  and  $-4\sqrt{d}\delta$  respectively<sup>§</sup>. Slightly abusing notation, we say that polytope  $p \subseteq H^+(\beta)$  if for all actions  $\alpha$  contained in  $p$  it holds that  $\alpha \in H^+(\beta)$ .

We define action  $\alpha_t$ 's *upper* and *lower* polytopes sets to be the sets of polytopes such that  $\mathcal{P}_t^u(\alpha_t) = \{p \subseteq \mathcal{P}_t, p \subseteq H^+(\beta_t^u(\alpha_t))\}$  and  $\mathcal{P}_t^l(\alpha_t) = \{p \subseteq \mathcal{P}_t, p \subseteq H^-(\beta_t^l(\alpha_t))\}$  respectively. Defining these sets is useful since they represent the subsets of the learner's action space for which she can infer  $\forall h : \ell(h, \mathbf{r}_t(h), y_t)$  despite never observing  $x_t$ ! To be more precise for  $h \in \mathcal{P}_t^u(\alpha_t) \cup \mathcal{P}_t^l(\alpha_t)$ :

$$\ell(h, \mathbf{r}_t(h), y_t) = \mathbb{1}\{y_t = -1\} \cdot \mathbb{1}\{h \in \mathcal{P}_t^u(\alpha_t)\} + \mathbb{1}\{y_t = +1\} \cdot \mathbb{1}\{h \in \mathcal{P}_t^l(\alpha_t)\}$$

The definition of the polytopes establishes that at each round the estimated loss within each polytope is *constant*. If a polytope has *not* been further "grinded" by the algorithm, then the estimated

<sup>§</sup>i.e.,  $\alpha \in H^+(\beta)$  if  $\alpha \in \mathcal{A}, \langle \beta, \alpha \rangle \geq 4\sqrt{d}\delta$  and similarly,  $\alpha \in H^-(\beta)$  if  $\alpha \in \mathcal{A}, \langle \beta, \alpha \rangle \leq -4\sqrt{d}\delta$

loss that was used to update the polytope has been the same within the actions of the polytope for each time step! This observation explains the way the weights of the polytopes are updated by scaling with the Lebesgue measure of each polytope. Due to the fact that the loss of all the points within a polytope is the same, we slightly abuse notation and we use  $\ell(p, \mathbf{r}_t(p), y_t)$  to denote the loss for any action  $\alpha \in p$  for round  $t$ , if the agent best-responded to it.

We next define the lower, upper, and middle  $\sigma_t$ -polytope induced sets as:

$$\mathcal{P}_{t,\sigma_t}^l = \{\alpha \in p, p \in \mathcal{P}_t : \mathcal{D}(\alpha, \mathbf{x}_t) \leq -2\delta\} \quad (\text{lower})$$

$$\mathcal{P}_{t,\sigma_t}^u = \{\alpha \in p, p \in \mathcal{P}_t : \mathcal{D}(\alpha, \mathbf{x}_t) \geq 2\delta\} \quad (\text{upper})$$

$$\mathcal{P}_{t,\sigma_t}^m = \{\alpha \in p, p \in \mathcal{P}_t : |\mathcal{D}(\alpha, \mathbf{x}_t)| < 2\delta\} \quad (\text{middle})$$

where  $\mathcal{D}(\alpha, \mathbf{x}_t) = \langle \alpha, \mathbf{x}_t \rangle / \|\alpha\|_2$ . Note that these can only be computed if one has access to the agent's true datapoint  $\sigma_t = (\mathbf{x}_t, y_t)$ . However, we only use them in our analysis, and GRINDER does not require access to them.

GRINDER uses access to what we call an *in-oracle* (Definition 5.3). Our main regret theorem is stated for an accurate oracle, but we show that our regret guarantees still hold for approximation oracles (Lemma B.5). Such oracles can be constructed in practice, as we show in Chapter 5.5.

**Definition 5.3** (In-Oracle). *We define the In-Oracle as a black-box algorithm, which takes as input a polytope (resp. action) and returns the total in-probability for this polytope (resp. action):*

$$\mathbb{P}_t^{\text{in}}[p] = \int_{\mathcal{A}} \mathbb{P}_t \left[ \left\{ p \subseteq H^+ (\beta_t^u(\alpha')) \right\} \bigcup \left\{ p \subseteq H^- (\beta_t^l(\alpha')) \right\} \right] d\alpha'$$

We also note that the algorithm can be turned into one that does not assume knowledge of  $T$  or  $\lambda(p)$  by using the standard *doubling trick* Auer et al. (1995).

The proof of Theorem 5.2 follows from a sequence of lemmas and claims presented below. By convention, we call a single point a *point-polytope*, and we use  $\bar{\mathcal{P}}$  for the set of all point-polytopes.

**Proposition 5.1.** *The two-stage sampling probability distribution  $\mathcal{D}_t$  is equivalent to a one-stage probability distribution of drawing directly an action from density  $d\pi_t(\cdot)$ .*

*Proof.* The one-stage sampling that draws an action from  $\pi_t$  is equivalent to choosing an action

---

**ALGORITHM 5.2: GRINDER Algorithm for Strategic Classification**


---

```

1 Initialize polytopes' set:  $\mathcal{P}_0 = \{\mathcal{A}\}$ .
2 Initialize polytope weights  $w_1(p) = \lambda(p), p \in \mathcal{P}_0$ .
3 Tune learning and exploration rates  $\eta = \gamma \leq 1/2$ , as specified in the analysis.
4 for  $t \leftarrow 1$  to  $T$  do
5   Compute  $\forall p \in \mathcal{P}_t : \pi_t(p) = (1 - \gamma)q_t(p) + \gamma \frac{\lambda(p)}{\lambda(\mathcal{A})}$ .           // distribution over polytopes
  /* Two-stage sampling: first, polytope, second, draw action from within. */
6   Select polytope  $p_t \sim \pi_t$  from which you draw action  $\alpha_t \sim \text{Unif}(p_t)$  and commit to  $\alpha_t$ .
7   Observe the agent's response  $(\mathbf{r}_t(\alpha_t), y_t)$  to committed  $\alpha_t$ .
  /* Space partitioning into smaller polytopes.  $\mathcal{P}_t$ : current polytopes set. */
8   Define a new set of polytopes  $\mathcal{P}_{t+1} = \mathcal{P}_{t+1}^u(\alpha_t) \cup \mathcal{P}_{t+1}^m(\alpha_t) \cup \mathcal{P}_{t+1}^l(\alpha_t)$ , where:
9   for each polytope  $p \in \mathcal{P}_t$  do
10    Add in  $\mathcal{P}_{t+1}^u(\alpha_t)$  the non-empty intersection  $p \cap H^+(\beta_t^u(\alpha_t))$  // upper polytopes set
11    Add in  $\mathcal{P}_{t+1}^l(\alpha_t)$  the non-empty intersection  $p \cap H^-(\beta_t^l(\alpha_t))$  // lower polytopes set
12    Add in  $\mathcal{P}_{t+1}^m(\alpha_t)$  the non-empty remainder of  $p$ .                      // middle polytopes set
13  Compute  $\hat{\ell}(\alpha_t, \mathbf{r}_t(\alpha_t), y_t) = \frac{\ell(\alpha_t, \mathbf{r}_t(\alpha_t), y_t)}{\mathbb{P}_t^{\text{in}}[\alpha_t]}$ .           // loss estimator for chosen action
14  for each polytope  $p \in \mathcal{P}_{t+1}$  do
    /* upper and lower polytopes get full information */
15    Compute  $\hat{\ell}(p, \mathbf{r}_t(p), y_t) = \frac{\ell(p, \mathbf{r}_t(p), y_t) \cdot \mathbb{1}\{p \subseteq \mathcal{P}_{t+1}^u(\alpha_t) \cup \mathcal{P}_{t+1}^l(\alpha_t)\}}{\mathbb{P}_t^{\text{in}}[p]}$ .
    /* weight scaling with the Lebesgue measure of the polytope */
16    Update  $w_{t+1}(p) = \lambda(p) \exp\left(-\eta \sum_{\tau=1}^t \hat{\ell}(p, \mathbf{r}_t(p), y_t)\right)$ ,  $q_{t+1}(p) = \frac{w_{t+1}(p)}{\sum_{p' \in \mathcal{P}_{t+1}} w_{t+1}(p')}$ .

```

---

$\alpha \in \mathcal{A}$  from probability density function:  $d\pi_t(\alpha) = (1 - \gamma)dq_t(\alpha) + \frac{\gamma}{\lambda(\mathcal{A})}$ . The two-stage sampling is:  $d\pi_{\mathcal{D}_t}(\alpha) = \frac{1}{\lambda(p)} \left( (1 - \gamma)q_t(p) + \frac{\gamma\lambda(p)}{\lambda(\mathcal{A})} \right)$ . Since  $q_t(p) = \lambda(p)dq_t(\alpha), \forall \alpha \in p$ , we get the result. ■

Moving forward we analyze the first and the second moment of the loss  $\hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)$  for each action  $\alpha$ , based on the induced probability distribution  $\mathcal{D}_t$ , assuming oracle access to  $\mathbb{P}_t^{\text{in}}[\alpha]$ .

**Lemma 5.1** (First Moment). *The estimated loss  $\hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)$  is an unbiased estimator of the true loss  $\ell(\alpha, \mathbf{r}_t(\alpha), y_t)$ , when actions are drawn from the induced probability distribution  $\mathcal{D}_t$ .*

*Proof.* For all the actions  $\alpha \in \mathcal{A}$ , given Proposition 5.1, it holds that:

$$\mathbb{E}_{\alpha_t \sim \mathcal{D}_t} [\hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)] = \int_{\mathcal{A}} f_{\mathcal{A}_t}(\alpha') \frac{\ell(\alpha, \mathbf{r}_t(\alpha), y_t) \mathbb{1}\{\alpha \in N^{\text{out}}(\alpha')\}}{\mathbb{P}_t^{\text{in}}[\alpha]} d\alpha' = \ell(\alpha, \mathbf{r}_t(\alpha), y_t)$$

■

**Lemma 5.2** (Second Moment). *For the second moment of the estimated loss  $\hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)$  with respect*

to the induced probability distribution  $\mathcal{D}_t$  it holds that:

$$\mathbb{E}_{\alpha_t \sim \mathcal{D}_t} [\widehat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)^2] = \frac{\ell(\alpha, \mathbf{r}_t(\alpha), y_t)^2}{\mathbb{P}_t^{\text{in}}[\alpha]} \leq \frac{1}{\mathbb{P}_t^{\text{in}}[\alpha]}$$

*Proof.* For all the actions  $\alpha \in \mathcal{A}$ , given Claim 5.1, it holds that:

$$\begin{aligned} \mathbb{E}_{\alpha_t \sim \mathcal{D}_t} [\widehat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)^2] &= \int_{\mathcal{A}} f_{\mathcal{A}_t}(\alpha') \frac{\ell(\alpha, \mathbf{r}_t(\alpha), y_t)^2 \mathbb{1}\{\alpha \in N^{out}(\alpha')\}}{\mathbb{P}_t^{\text{in}}[\alpha]^2} d\alpha' \\ &= \frac{\ell(\alpha, \mathbf{r}_t(\alpha), y_t)^2}{\mathbb{P}_t^{\text{in}}[\alpha]} \leq \frac{1}{\mathbb{P}_t^{\text{in}}[\alpha]} \end{aligned}$$

■

**Lemma 5.3.** Let  $\underline{p}(t) = \arg \min_{p \in \mathcal{P}_t \setminus \overline{\mathcal{P}}_t} \lambda(p)$  be the polytope with the smallest Lebesgue measure (excluding point-polytopes) after  $t$  rounds. Then, the following inequality holds:

$$\mathbb{E}_{\alpha_t \sim \mathcal{D}_t} \left[ \frac{1}{\mathbb{P}_t^{\text{in}}[\alpha_t]} \right] \leq 4 \log \left( \frac{4\lambda(\mathcal{A}) \cdot |\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l|}{\gamma \lambda(\underline{p}(t))} \right) + \lambda(\mathcal{P}_{t,\sigma_t}^m)$$

*Proof.* By definition, we expand the term:  $\mathbb{E}_{\alpha \sim \mathcal{D}_t} \left[ \frac{1}{\mathbb{P}_t^{\text{in}}[\alpha_t]} \right]$  as follows:

$$\begin{aligned} \mathbb{E}_{\alpha_t \sim \mathcal{D}_t} \left[ \frac{1}{\mathbb{P}_t^{\text{in}}[\alpha_t]} \right] &= \int_{\mathcal{A}} \frac{f_{\mathcal{A}_t}(\alpha)}{\mathbb{P}_t^{\text{in}}[\alpha]} d\alpha \\ &= \underbrace{\int_{\bigcup(\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l)} \frac{f_{\mathcal{A}_t}(\alpha)}{\mathbb{P}_t^{\text{in}}[\alpha]} d\alpha}_{Q_1} + \underbrace{\int_{\bigcup \mathcal{P}_{t,\sigma_t}^m} \frac{f_{\mathcal{A}_t}(\alpha)}{\mathbb{P}_t^{\text{in}}[\alpha]} d\alpha}_{Q_2} \end{aligned} \tag{5.4.1}$$

where by integrating over  $\bigcup \mathcal{P}$  we denote the integral over all *actions* that belong in some polytope from the set  $\mathcal{P}$ . In the right hand side of Equation (5.4.1), term  $Q_2$  is relatively easier to analyze. Due to the conservative estimates of the *true* middle space (i.e., the actions such that  $\mathcal{D}(\alpha, \mathbf{x}_t) \leq \delta$ ), the set of polytopes  $\mathcal{P}_{t,\sigma_t}^m$  contains *all* the actions that actually belong in the  $\sigma_t$ -induced middle space, plus some other actions for which the agent could not have misreported, due to their  $\delta$ -boundedness. Now, for all the actions that actually belong in the  $\sigma_t$ -induced middle space, it holds that they only get information (i.e., get updated) when they are chosen by the algorithm, while for the rest of the actions that have ended up in our middle space, they could have been updated by

other actions as well. Thus, it holds that:

$$\forall \alpha \in \bigcup \mathcal{P}_{t,\sigma_t}^m : \mathbb{P}_t^{\text{in}}[\alpha] \geq f_{\mathcal{A}_t}(\alpha)$$

As a result:

$$Q_2 = \int_{\bigcup \mathcal{P}_{t,\sigma_t}^m} \frac{f_{\mathcal{A}_t}(\alpha)}{\mathbb{P}_t^{\text{in}}[\alpha]} d\alpha \leq \int_{\bigcup \mathcal{P}_{t,\sigma_t}^m} \frac{f_{\mathcal{A}_t}(\alpha)}{f_{\mathcal{A}_t}(\alpha)} d\alpha = \lambda(\mathcal{P}_{t,\sigma_t}^m) \quad (5.4.2)$$

Moving forward, we turn our attention to term  $Q_1$ . Assume now that an action  $\alpha$  belongs in a polytope  $p_\alpha$ . Then, there are (weakly) more actions that can potentially update action  $\alpha$ , than the whole polytope in which it belongs,  $p_\alpha$ ; indeed, in order to update the polytope, one must make sure that every action within it is updateable. As a result,  $\mathbb{P}_t^{\text{in}}[\alpha] \geq \mathbb{P}_t^{\text{in}}[p_\alpha]$ . Using this in Equation (5.4.1) we get that the first term of the RHS of the variance is upper bounded by:

$$Q_1 \leq \sum_{p \in \mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l} \int_p \frac{f_{\mathcal{A}_t}(\alpha)}{\mathbb{P}_t^{\text{in}}[p]} d\alpha \quad (5.4.3)$$

Further, let  $\mathbb{P}_t^{\text{in}}[p]_{u,l}$  be the part of  $\mathbb{P}_t^{\text{in}}[p]$  that depends only in the updates that stem from actions in either the upper or the lower polytopes sets. As such:  $\mathbb{P}_t^{\text{in}}[p]_{u,l} \leq \mathbb{P}_t^{\text{in}}[p]$  and the term in Equation (5.4.3) can be upper bounded by:

$$Q_1 \leq \sum_{p \in \mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l} \frac{1}{\mathbb{P}_t^{\text{in}}[p]_{u,l}} \int_p f_{\mathcal{A}_t}(\alpha) d\alpha \quad (5.4.4)$$

where we have also used the fact that we gain oracle access to quantity  $\mathbb{P}_t^{\text{in}}[p]_{u,l}$  and therefore, we treat it as a constant in the integral. Observe now that the term  $\int_p f_{\mathcal{A}_t}(\alpha) d\alpha$  corresponds to the total probability that the action  $\alpha_t$ , which is chosen from the induced probability distribution  $\mathcal{D}_t$ , belongs to polytope  $p$ , i.e., it is equal to  $\pi_t(p)$ . Hence, the upper bound in Equation (5.4.4) can be relaxed to:

$$Q_1 \leq \sum_{p \in \mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l} \frac{\pi_t(p)}{\mathbb{P}_t^{\text{in}}[p]_{u,l}} \quad (5.4.5)$$

As we have explained before,  $\pi_t(p) = 0$ , for  $p \in \bar{\mathcal{P}}_t$  and as a result, we can disregard point-polytopes from our consideration for the rest of this proof. We now upper bound this term by using the graph-

theoretic lemma of [Alon et al. \(2015, Lemma 5\)](#), which we provide below for completeness.

**Lemma 5.4** ([\(Alon et al., 2015, Lemma 5\)](#)). *Let  $G = (V, E)$  be a directed graph with  $|V| = K$ , in which each node  $i \in V$  is assigned a positive weight  $w_i$  lower bounded by a positive scalar  $\varepsilon \in (0, 1/2)$ , i.e.,  $w_i \geq \varepsilon, \forall i \in V$ . If  $\sum_{i \in V} w_i \leq 1$  then, denoting by  $\alpha^G$  the independence number of  $G$  we have that:*

$$\sum_{i \in V} \frac{w_i}{w_i + \sum_{j \in N^{in}(i)} w_j} \leq 4\alpha^G \frac{4K}{\alpha^G \varepsilon}$$

Observe that all the actions within the  $\sigma_t$ -induced upper and the lower polytopes set form the following feedback graph: each node corresponds to a polytope from one of the sets  $\mathcal{P}_{t,\sigma_t}^u, \mathcal{P}_{t,\sigma_t}^l$ . So the total number of nodes is at most  $|\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l|$ , where by  $|S|$  we denote the cardinality of a set  $S$ . Each edge  $(i, j)$  corresponds to *information passing* from node  $i$  to node  $j$ , i.e., the directed edge  $(i, j)$  exists when the loss for actions of polytope  $j$  can be computed by just observing the loss for action from the polytope  $i$ . However, for each action belonging in a polytope among the  $\sigma_t$ -induced upper and lower polytopes sets, we know that the agent could not possibly misreport, due to him being myopically rational and  $\delta$ -bounded, and as a result, the loss for all the actions within the upper and the lower polytopes sets can be computed! As a result, the independence number of this feedback graph is  $\alpha^G = 1$ . Using the fact that each polytope  $p$  is chosen with probability at least  $\pi_t(p) \geq \gamma \frac{\lambda(p)}{\lambda(\mathcal{A})} \geq \gamma \frac{\lambda(p(t))}{\lambda(\mathcal{A})}$  we can apply Lemma 5.4 for  $\varepsilon = \gamma \frac{\lambda(p(t))}{\lambda(\mathcal{A})}$  and  $\alpha^G = 1$  and obtain:

$$Q_1 \leq 4 \log \left( \frac{4\lambda(\mathcal{A}) \cdot |\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l|}{\lambda(p(t)) \cdot \gamma} \right)$$

Summing up the upper bounds for  $Q_1$  and  $Q_2$  we get:

$$\mathbb{E}_{\alpha_t \sim \mathcal{D}_t} \left[ \frac{1}{\mathbb{P}_t^{\text{in}}[\alpha_t]} \right] \leq 4 \log \left( \frac{4\lambda(\mathcal{A}) \cdot |\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l|}{\lambda(p(t)) \cdot \gamma} \right) + \lambda(\mathcal{P}_{t,\sigma_t}^m)$$

■

**Lemma 5.5** (Second Order Regret Bound). *Let  $q_1, \dots, q_T$  be the probability distribution over the polytopes defined by in Step 15 of Algorithm 5.2 for the estimated losses  $\hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)$ ,  $t \in [T]$ . Then, the second*

order regret bound induced by GRINDER is:

$$\begin{aligned} & \sum_{t=1}^T \sum_{p \in \mathcal{P}_{t+1}} q_t(p) \hat{\ell}(p, \mathbf{r}_t(p), y_t) - \sum_{t=1}^T \hat{\ell}(\alpha^\star, \mathbf{r}_t(\alpha^\star), y_t) \\ & \leq \frac{\eta}{2} \sum_{t=1}^T \sum_{p \in \mathcal{P}_{t+1}} q_t(p) \hat{\ell}(p, \mathbf{r}_t(p), y_t)^2 + \frac{1}{\eta} \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) \end{aligned} \quad (5.4.6)$$

where  $\underline{p}$  is the polytope with the smallest Lebesgue measure in the finest partition of space  $\mathcal{A}$ :

$$\underline{p} = \arg \min_{p \in \mathcal{P}_T \setminus \bar{\mathcal{P}}_T} \lambda(p).$$

*Proof.* Let  $W_t = \sum_{p \in \mathcal{P}_t} w_t(p)$ . We upper and lower bound the quantity  $Q = \sum_{t=1}^T \log \frac{W_{t+1}}{W_t}$ . Then,

$$Q = \sum_{t=1}^T \log \left( \frac{W_{t+1}}{W_t} \right) = \log \left( \frac{W_T}{W_1} \right) \quad (5.4.7)$$

Observe now that in  $t = 1$  there only exists one polytope (the whole  $[-1, 1]^{d+1}$  space), with a total weight of  $\lambda(\mathcal{A})$  and a probability of 1. In other words, all the actions within this polytope have the same weight, which is equal to 1 (uniformly weighted). As a result,  $\log W_1 = \log \left( \sum_{p \in \mathcal{P}_1} \int_{\mathcal{A}} 1 d\alpha \right) = \log (\lambda(\mathcal{A}))$ . For term  $\log W_T$  we have:

$$\begin{aligned} \log W_T &= \log \left( \sum_{p \in \mathcal{P}_T} w_T(p) \right) = \log \left( \int_{\mathcal{A}} w_T(\alpha) d\alpha \right) \\ &= \log \left( \sum_{p \in \mathcal{P}_T \setminus \bar{\mathcal{P}}_T} \lambda(p) \exp \left( -\eta \sum_{t=1}^T \hat{\ell}(p, \mathbf{r}_t(p), y_t) \right) + \int_{\cup \bar{\mathcal{P}}_T} \exp \left( -\eta \sum_{t=1}^T \hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t) \right) d\alpha \right) \end{aligned} \quad (5.4.8)$$

where the last equality is due to the fact that not further grinded polytopes have maintained the *same* estimated loss,  $\hat{\ell}$ , for *all* their containing points at each round  $t$  and we denote by  $\bar{\mathcal{P}}_T$  the set of point-polytopes contained in  $\mathcal{P}_T$ .

Since the horizon  $T$  is finite, set  $\bar{\mathcal{P}}_T$  is essentially a set of points, and it has a Lebesgue measure

of 0. Hence,

$$\int_{\bigcup \bar{\mathcal{P}}_T} \exp \left( -\eta \sum_{t=1}^T \hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t) \right) d\alpha = 0$$

Let  $\alpha^* = \arg \min_{\alpha \in \mathcal{A}} \sum_{t=1}^T \hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)$  (i.e., the best fixed action in hindsight among the *all* actions after  $T$  rounds, irrespective of whether it belongs to  $\bigcup \bar{\mathcal{P}}_T$  or  $\bigcup \mathcal{P}_T \setminus \bar{\mathcal{P}}_T$ ) and  $\underline{p} \in \mathcal{P}_T \setminus \bar{\mathcal{P}}_T$  be the polytope with the smallest Lebesgue measure in  $\mathcal{P}_T \setminus \bar{\mathcal{P}}_T$  (i.e., excluding point-polytopes). Then, denoting by  $p^* \in \mathcal{P}_T$  the polytope where  $\alpha^*$  belongs to, among the set of active polytopes  $\mathcal{P}_T$ , Equation (5.4.8) becomes can be lower bounded as follows:

$$\begin{aligned} \log W_T &= \log \left( \sum_{p \in \mathcal{P}_T \setminus \bar{\mathcal{P}}_T} \lambda(p) \exp \left( -\eta \sum_{t=1}^T \hat{\ell}(p, \mathbf{r}_t(p), y_t) \right) \right) \\ &\geq \log \left( \lambda(\underline{p}) \sum_{p \in \mathcal{P}_T \setminus \bar{\mathcal{P}}_T} \exp \left( -\eta \sum_{t=1}^T \hat{\ell}(p, \mathbf{r}_t(p), y_t) \right) \right) \quad (\lambda(p) \geq \lambda(\underline{p}), \forall p \in \mathcal{P}_T \setminus \bar{\mathcal{P}}_T) \\ &\geq \log \left( \lambda(\underline{p}) \cdot \exp \left( -\eta \sum_{t=1}^T \hat{\ell}(p^*, \mathbf{r}_t(p^*), y_t) \right) \right) \quad (e^{-x} \geq 0, \forall x) \\ &= \log \left( \lambda(\underline{p}) \cdot \exp \left( -\eta \sum_{t=1}^T \hat{\ell}(\alpha^*, \mathbf{r}_t(\alpha^*), y_t) \right) \right) = \log(\lambda(\underline{p})) - \eta \sum_{t=1}^T \hat{\ell}(\alpha^*, \mathbf{r}_t(\alpha^*), y_t) \end{aligned} \tag{5.4.9}$$

As a result:

$$Q = \log W_T - \log W_1 \geq \log \left( \frac{\lambda(\underline{p})}{\lambda(\mathcal{A})} \right) - \eta \sum_{t=1}^T \hat{\ell}(\alpha^*, \mathbf{r}_t(\alpha^*), y_t) \tag{5.4.10}$$

We move on to the upper bound of  $Q$  now. Upper bounding quantity  $\log \frac{W_{t+1}}{W_t}$  we get:

$$\begin{aligned} \log \left( \frac{W_{t+1}}{W_t} \right) &= \log \left( \frac{\int_{\mathcal{A}} w_t(\alpha) \exp \left( -\eta \hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t) \right) d\alpha}{W_t} \right) \\ &= \log \left( \int_{\mathcal{A}} q_t(\alpha) \exp \left( -\eta \hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t) \right) d\alpha \right) \\ &\leq \log \left( \int_{\mathcal{A}} q_t(\alpha) \left( 1 - \eta \hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t) + \frac{\eta^2}{2} \hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)^2 \right) d\alpha \right) \quad (e^{-x} \leq 1 - x + \frac{x^2}{2}, x \in [0, 1]) \\ &\leq \log \left( 1 - \eta \int_{\mathcal{A}} q_t(\alpha) \hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t) d\alpha + \frac{\eta^2}{2} \int_{\mathcal{A}} q_t(\alpha) \hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)^2 d\alpha \right) \quad (\int_{\mathcal{A}} q_t(\alpha) d\alpha = 1) \\ &\leq -\eta \int_{\mathcal{A}} q_t(\alpha) \hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t) d\alpha + \frac{\eta^2}{2} \int_{\mathcal{A}} q_t(\alpha) \hat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)^2 d\alpha \quad (\log(1 - x) \leq x, x \leq 0) \end{aligned}$$

Summing up for the  $T$  rounds the latter becomes:

$$\sum_{t=1}^T \log \left( \frac{W_{t+1}}{W_t} \right) \leq - \sum_{t=1}^T \eta \int_{\mathcal{A}} q_t(\alpha) \widehat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t) d\alpha + \sum_{t=1}^T \frac{\eta^2}{2} \int_{\mathcal{A}} q_t(\alpha) \widehat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)^2 d\alpha \quad (5.4.11)$$

Combining the upper and lower bounds of Equations (5.4.10) and (5.4.11) we get that:

$$\begin{aligned} \sum_{t=1}^T \int_{\mathcal{A}} q_t(\alpha) \widehat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t) d\alpha - \sum_{t=1}^T \widehat{\ell}(\alpha^*, \mathbf{r}_t(\alpha^*), y_t) &\leq \\ \leq \frac{\eta}{2} \sum_{t=1}^T \int_{\mathcal{A}} q_t(\alpha) \widehat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)^2 d\alpha + \frac{1}{\eta} \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) \end{aligned}$$

■

We are now ready for the proof of Theorem 5.2.

*Proof of Theorem 5.2.* By taking the expectation with respect to distribution  $\mathcal{D}_t$  in Lemma 5.5:

$$\begin{aligned} \sum_{t=1}^T \int_{\mathcal{A}} q_t(\alpha) \mathbb{E}_{\mathcal{D}_t} [\widehat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)] d\alpha - \sum_{t=1}^T \mathbb{E}_{\mathcal{D}_t} [\widehat{\ell}(\alpha^*, \mathbf{r}_t(\alpha^*), y_t)] &\leq \\ \leq \frac{\eta}{2} \sum_{t=1}^T \int_{\mathcal{A}} q_t(\alpha) \mathbb{E}_{\mathcal{D}_t} [\widehat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)^2] d\alpha + \frac{1}{\eta} \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) \end{aligned}$$

Combining Lemmas 5.1, 5.2 with the latter we get:

$$\begin{aligned} \sum_{t=1}^T \int_{\mathcal{A}} q_t(\alpha) \ell(\alpha, \mathbf{r}_t(\alpha), y_t) d\alpha - \sum_{t=1}^T \ell(\alpha^*, \mathbf{r}_t(\alpha^*), y_t) & \\ \leq \sum_{t=1}^T \frac{\eta}{2} \int_{\mathcal{A}} \frac{q_t(\alpha)}{\mathbb{P}_t^{\text{in}}[\alpha]} d\alpha + \frac{1}{\eta} \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) & \\ \leq \sum_{t=1}^T \eta \int_{\mathcal{A}} \frac{\pi_t(\alpha)}{\mathbb{P}_t^{\text{in}}[\alpha]} d\alpha + \frac{1}{\eta} \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) & \quad (\pi_t(\alpha) \geq (1-\gamma)q_t(\alpha) \text{ and } \gamma \leq \frac{1}{2}) \\ \leq \sum_{t=1}^T \eta \left( 4 \log \left( \frac{4\lambda(\mathcal{A}) |\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l|}{\gamma \cdot \lambda(\underline{p}(t))} \right) + \lambda(\mathcal{P}_{t,\sigma_t}^m) \right) + \frac{1}{\eta} \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) & \quad (\text{Lemma 5.3}) \end{aligned}$$

Using the fact that  $\int_{\mathcal{A}} \pi_t(\alpha) d\alpha \leq \int_{\mathcal{A}} q_t(\alpha) d\alpha + \gamma$ , the latter becomes:

$$\mathcal{R}(T) \leq \gamma T + \eta \sum_{t=1}^T \left( 4 \log \left( \frac{4\lambda(\mathcal{A}) |\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l|}{\gamma \cdot \lambda(\underline{p}(t))} \right) + \lambda(\mathcal{P}_{t,\sigma_t}^m) \right) + \frac{1}{\eta} \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right)$$

Setting  $\gamma = \eta$ :

$$\mathcal{R}(T) \leq \eta \sum_{t=1}^T \left( 1 + 4 \log \left( \frac{4\lambda(\mathcal{A}) |\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l|}{\eta \cdot \lambda(\underline{p}(t))} \right) + \lambda(\mathcal{P}_{t,\sigma_t}^m) \right) + \frac{1}{\eta} \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right)$$

which can be relaxed to:

$$\begin{aligned} \mathcal{R}(T) &\leq \eta \sum_{t=1}^T \left( 1 + 4 \log \left( \frac{4\lambda(\mathcal{A}) |\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l|}{\lambda(\underline{p}(t))} \right) + \lambda(\mathcal{P}_{t,\sigma_t}^m) \right) + \frac{1}{\eta} \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) \\ &\leq \eta \sum_{t=1}^T \left( 1 + 4 \log \left( \frac{4\lambda(\mathcal{A}) |\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l| T}{\lambda(\underline{p})} \right) + \lambda(\mathcal{P}_{t,\sigma_t}^m) \right) + \frac{1}{\eta} \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) \quad (\lambda(\underline{p}(t)) \geq \lambda(\underline{p})) \\ &\leq \eta \cdot \max_{t \in [T]} \left\{ 1 + 4 \log \left( \frac{4\lambda(\mathcal{A}) |\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l| T}{\lambda(\underline{p})} \right) + \lambda(\mathcal{P}_{t,\sigma_t}^m) \right\} \cdot T + \frac{1}{\eta} \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) \end{aligned}$$

Tuning  $\eta$  to be

$$\eta = \sqrt{\frac{\log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right)}{\max_{t \in [T]} \left\{ 1 + 4 \log \left( \frac{4\lambda(\mathcal{A}) |\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l| T}{\lambda(\underline{p})} \right) + \lambda(\mathcal{P}_{t,\sigma_t}^m) \right\} \cdot T}}$$

we get that the Stackelberg regret is upper bounded by:

$$\mathcal{R}(T) \leq \mathcal{O} \left( \sqrt{\max_{t \in [T]} \left\{ \lambda(\mathcal{P}_{t,\sigma_t}^m) + 4 \log \left( \frac{4\lambda(\mathcal{A}) |\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l| T}{\lambda(\underline{p})} \right) + 1 \right\} \cdot \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) \cdot T} \right)$$

Since the actions that belong in  $\mathcal{P}_{t,\sigma_t}^m$  are a subset of all the actions in  $\mathcal{A}$ , then  $\lambda(\mathcal{P}_{t,\sigma_t}^m) \leq \lambda(\mathcal{A}) = 1$ .

The set of all polytopes is upper bounded by  $\frac{\lambda(\mathcal{A})}{\lambda(\underline{p})}$  and hence,  $|\mathcal{P}_{t,\sigma_t}^u \cup \mathcal{P}_{t,\sigma_t}^l| \leq \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})}$ . Hence, for the Stackelberg regret we have:

$$\begin{aligned} \mathcal{R}(T) &\leq \mathcal{O} \left( \sqrt{\log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} T \right) \cdot \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) T} \right) \\ &\leq \mathcal{O} \left( \sqrt{\left( \lambda(\mathcal{A}) + 1 + 4 \log \left( \frac{4\lambda(\mathcal{A})}{\lambda(\underline{p})} \cdot \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \cdot T \right) \right) \cdot \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) T} \right) \\ &\leq \mathcal{O} \left( \sqrt{\left( \lambda(\mathcal{A}) + 1 + 8 \log \left( \frac{2\lambda(\mathcal{A})}{\lambda(\underline{p})} \cdot T \right) \right) \cdot \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) T} \right) \end{aligned}$$

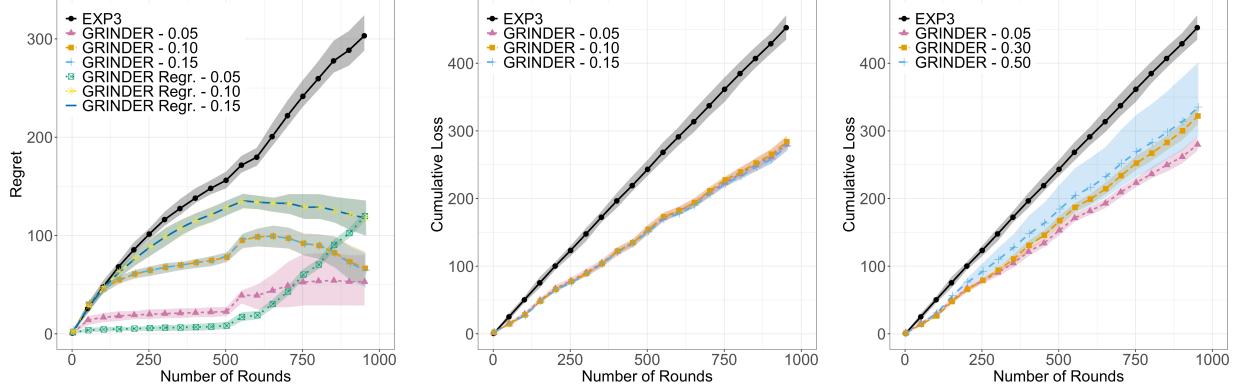


Figure 5.4: Performance of GRINDER vs. EXP3. In all cases, GRINDER outperforms EXP3. Solid lines correspond to average regret/loss, and opaque bands correspond to 10th and 90th percentile. **Left:** discrete action sets, accurate and regression oracle. **Middle:** Continuous action set for GRINDER with  $\delta = 0.05, 0.10, 0.15$ . **Right:** Continuous action set for GRINDER with  $\delta = 0.05, 0.3, 0.5$ .

where the  $\mathcal{O}(\cdot)$  notation hides constants with respect to the horizon  $T$ . The last inequality is upper bounded by  $\mathcal{O}\left(\sqrt{\log\left(\frac{\lambda(\mathcal{A})}{\lambda(p)}T\right) \cdot \log\left(\frac{\lambda(\mathcal{A})}{\lambda(p)}\right) T}\right)$ . ■

The regret guarantee of GRINDER is *preserved* if instead of an *accurate* in-oracle it is provided an  $\varepsilon$ -*approximate* one, where  $\varepsilon \leq 1/\sqrt{T}$  (Lemma B.5). As we also validate in Chapter 5.5, in settings where few points violate the margin between the +1 and the -1 labeled points such approximation oracles do exist and are relatively easy to construct.

Computing the volume of polytopes is a #P hard problem, so GRINDER should be viewed as an information-theoretic result. However, if GRINDER is provided access to an efficient black-box algorithm for computing the volume of a polytope, its runtime complexity is  $\mathcal{O}(T^d)$  (Lemma B.6).

## 5.5 SIMULATIONS

In this section, we present our simulation results. We build simulation datasets since in order to *evaluate* the performance of our algorithms one needs to know the original datapoints  $x_t$ . The results of our simulations are presented in Figure 5.4. Our code is publicly available here: <https://github.com/charapod/learn-strat-class>.

For the simulation, we run GRINDER against EXP3 for a horizon  $T = 1000$ , where each round was repeated for 30 repetitions. The  $\delta$ -BMR agents that we used are best-responding according to the

utility function of Eq. (5.2.1), and we studied 5 different values for  $\delta$ : 0.05, 0.1, 0.15, 0.3, 0.5. The +1 labeled points are drawn from Gaussian distribution as  $\mathbf{x}_t \sim (\mathcal{N}(0.7, 0.3), \mathcal{N}(0.7, 0.3))$  and the -1 labeled points are drawn from  $\mathbf{x}_t \sim (\mathcal{N}(0.4, 0.3), \mathcal{N}(0.4, 0.3))$ . Thus, we establish that for the majority of the points there is a clear “margin” but there are few points that violate it (i.e., there exists no perfect linear classifier).

`EXP3` is always run with a fixed set of actions and always suffers a dependence on the different actions (i.e., not  $\delta$ ). We then run `GRINDER` in the same fixed set of actions and with a continuous action set. For the discrete action set, we include the results for both the accurate and the regression-based approximate oracle. We remark that if the action set is discrete, then `GRINDER` becomes similar to standard online learning with feedback graph algorithms (see e.g., (Alon et al., 2015)), but the feedback graph is built according to  $\delta$ -BMR agents. In this case, the regret scales as  $\mathcal{O}(a(G) \log T)$ , where  $a(G)$  is the independence number of graph  $G$ .

For the continuous action set it is not possible to identify the best-fixed action in hindsight. As a result, we report the cumulative loss. In Appendix B.3, we include additional simulations for a different family of  $\delta$ -BMR agents, and different distributions of labels. Namely, their utility function is:  $u_t(\alpha_t, \mathbf{r}_t(\alpha_t), \sigma_t) = \delta' \cdot \langle \alpha_t, \mathbf{r}_t(\alpha_t) \rangle - \|\mathbf{x}_t - \mathbf{r}_t(\alpha_t)\|_2$ . In order to build the approximation oracle we used past data and we trained a logistic regression model for each polytope, learning the probability that it is updated. Our model has “recency bias” and gives more weight to more recent datapoints. We expect that for more accurate oracles, our results are strengthened, as proved by our theoretical bounds.

Validating our theoretical results, `GRINDER` outperforms the benchmark, *despite the fact that we use an approximation oracle*. We also see that in the discrete action set, where an accurate oracle can be constructed, `GRINDER` performs much better than the regression oracle. As expected, `GRINDER`’s performance becomes worse as the power of the adversary increases (i.e., as  $\delta$  grows larger).

**Why not compare GRINDER with a convex surrogate?** We explain here our decision to only compare GRINDER with EXP3.

In fact, *no standard convex surrogate can be used* for learning linear classifiers against  $\delta$ -BMR, since the learner does not know precisely the agent's response function  $\mathbf{r}_t(\alpha)$ . As a result, the learner cannot guarantee that  $\ell(\alpha, \mathbf{r}_t(\alpha))$  is *convex* in  $\alpha$ , even if  $\ell(\alpha, \mathbf{z})$  ( $\mathbf{z}$  being independent of  $\alpha$ ) is convex in  $\alpha$ ! Concretely, think about the following counterexample: let  $h = (1, 1, -1)$ ,  $h' = (0.5, -1, 0.25)$  be two hyperplanes, a point

$\mathbf{x} = (0.55, 0.4)$ ,  $y = +1$ ,  $\delta = 0.1$ , and let  $\ell(h, \mathbf{r}(h)) = \max\{0, 1 - y \cdot \langle h, \mathbf{r}(h) \rangle\}$  (i.e., hinge loss, which is convex). We show that when  $(\mathbf{x}, y)$  is a  $\delta$ -BMR agent,  $\ell(\alpha, \mathbf{r}(\alpha))$  is *no longer convex* in  $\alpha$ . Take  $b = 0.5$  and construct  $h_b = 0.5h + 0.5h' = (0.75, 0, -0.375) = (1, 0, -0.5)$ .  $(\mathbf{x}, y)$  only misreports to (say)  $(0.61, 0.4)$  when presented with  $h$  (as  $h_b$  and  $h'$  classify  $\mathbf{x}$  as  $+1$ ). Computing the loss:  $\ell(h_b, \mathbf{r}(h_b)) = 0.95$ ,  $\ell(h, \mathbf{r}(h)) = 0.99$  and  $\ell(h', \mathbf{r}(h')) = 0.875$ , so,  $\ell(h_b, \mathbf{r}(h_b)) > b\ell(h, \mathbf{r}(h)) + (1-b)\ell(h', \mathbf{r}(h'))$ . Since in general  $\ell(\alpha, \mathbf{r}(\alpha))$  is not convex, it is may seem unfair to compare Bandit Gradient Descent (BGD) with GRINDER but we include comparison results in Figure 5.5, where GRINDER greatly outperforms BGD, for completeness. Identifying surrogate losses that are convex against  $\delta$ -BMR agents remains a very interesting open question.

## 5.6 LOWER BOUND

In this section, we prove nearly matching lower bounds for learning a linear classifier against  $\delta$ -BMR agents. To do so, we use the geometry of the sequence of datapoints  $\sigma_t$  interpreted in the dual space.

### Theorem 5.3: Lower Bound for Learning Against $\delta$ -BMR Agents

For any strategy and any  $\delta$ , there exists a sequence of  $\{\sigma_t\}_{t=1}^T$  such that:

$$\mathbb{E} \left[ \sum_{t \in [T]} \ell(\alpha_t, \mathbf{r}_t(\alpha_t), y_t) \right] - \min_{\alpha^* \in \mathcal{A}} \mathbb{E} \left[ \sum_{t \in [T]} \ell(\alpha^*, \mathbf{r}_t(\alpha^*), y_t) \right] \geq \frac{1}{9\sqrt{2}} \sqrt{T \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\tilde{p}_\delta)} \right)} \quad (5.6.1)$$

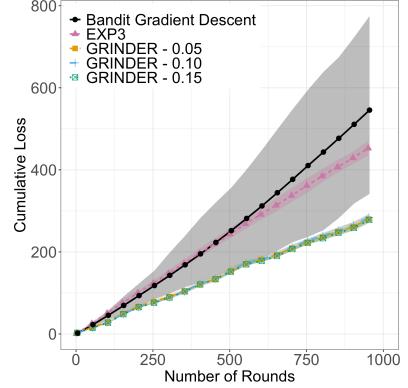


Figure 5.5: GRINDER vs. BGD

where  $\tilde{p}_\delta$  is the smallest  $\sigma_t$ -induced polytope from the sequence of  $\{\sigma_t\}_{t=1}^T$ .

We remark here that the  $\sigma_t$ -induced polytopes as defined in the previous section depend *only* on the sequence of agents that we face, and not on the properties of any algorithm.

**Lemma 5.6.** Fix a  $\mathbf{r} = \mathbf{x} = (u)^d$ , where by  $(u)^d$  we denote the  $d$ -dimensional vector with  $u \in [1/4, 3/4]$  in every dimension. There exists a utility model for the agents, and a pair of adversarial environments  $U$  and  $L$  such that  $\mathbf{r}_t(\alpha) = \mathbf{x}_t = \mathbf{x}, \forall \alpha \in \mathcal{A}, \forall t \in [T]$ , and the sequence of  $y_1, \dots, y_T$  is i.i.d. conditional on the choice of the adversary, such that:

$$\max_{\nu \in \{U, L\}} \min_{\alpha^* \in \mathcal{A}} \mathbb{E}_\nu \left[ \sum_{t \in [T]} \ell(\alpha_t, \mathbf{r}_t(\alpha_t), y_t) - \sum_{t \in [T]} \ell(\alpha^*, \mathbf{r}_t(\alpha^*), y_t) \right] \geq \frac{1}{9\sqrt{2}} \sqrt{T}$$

*Proof.* We are going to show this for the case where the agents  $\forall t \in [T]$  are *truthful*, i.e., they decide to report  $\mathbf{r}_t(\alpha) = \mathbf{x}_t, \forall \alpha \in \mathcal{A}, \forall t \in [T]$ . Of course, the learner does not know (and cannot infer) that, so fix a  $\delta > 0$  for the  $\delta$ -boundedness of the agents' utility function. We will prove the lemma only for deterministic strategies for the learner. As is customary, the claim for general strategies can be concluded by averaging over the learner's internal randomness and Fubini's theorem.

Fix an  $\varepsilon > 0$ , and a scalar  $u \in [1/4, 3/4]$ , and define the adversarial environments as follows:  $U$  is such that  $y_t = +1$  with probability  $1/2 + \varepsilon$  and  $y_t = -1$  with probability  $1/2 - \varepsilon$ , and  $L$  is such that  $y_t = -1$  with probability  $1/2 + \varepsilon$  and  $y_t = +1$  with probability  $1/2 - \varepsilon$ . This means that under  $U$ , the majority of times the label is  $+1$ , and under  $L$ , the majority of times the label is  $-1$ . Hence, under  $U$ , any action  $\alpha$  such that  $\langle \alpha, \mathbf{x} \rangle \geq 2\delta$  is *optimal* and under  $L$ , any action  $\alpha$  such that  $\langle \alpha, \mathbf{x} \rangle \leq -2\delta$  is *optimal*.

Take a sequence of actions  $\alpha_1, \dots, \alpha_T$  and let  $T_{\geq \delta}$  denote the *number* of rounds for which  $\langle \alpha_t, \mathbf{r} \rangle \geq \sqrt{d}\delta$ , and  $T_{\leq -\delta}$  the number of rounds for which  $\langle \alpha_t, \mathbf{r} \rangle \leq -\sqrt{d}\delta$ . Since  $T_{\leq -\delta} + T_{\geq \delta} \leq T$  we get that:

$$\begin{aligned} \mathbb{E}_U [\mathcal{R}(T)] &\geq \mathbb{E}_U [\mathcal{R}(T_{\leq -\delta})] \\ &\geq \sum_{t \in [T_{\leq -\delta}]} \left[ 1 \cdot \left( \frac{1}{2} + \varepsilon \right) - 1 \cdot \left( \frac{1}{2} - \varepsilon \right) \right] \\ &\geq 2\varepsilon \mathbb{E}_U [T_{\leq -\delta}] \end{aligned} \tag{5.6.2}$$

where the first inequality is due to the fact that  $\ell(\alpha_t, \mathbf{x}, y_t) = 0 = \ell(\alpha_U^*, \mathbf{x}, y_t), \forall t \in [T_{\geq \delta}]$  and any optimal action  $\alpha_U^*$  under  $U$  as we reasoned before. The second inequality uses the following two facts: first, that  $\ell(\alpha_U^*, \mathbf{x}, y_t) = 1, \forall t \in [T_{\leq -\delta}]$ , i.e., the best fixed action in hindsight when one encounters adversarial environment  $U$  is an action that estimates the label of  $\mathbf{x}$  to be 1. Second, that when playing against environment  $U$ , a learner incurs loss of 1 every time that she predicted the label of  $\mathbf{x}$  to be  $-1$  (which happens in at least all  $T_{\leq -\delta}$  rounds), and the actual label was 1 (which happens with probability  $1/2 + \varepsilon$ ). Similarly, we also see that

$$\mathbb{E}_L [\mathcal{R}(T)] \geq 2\varepsilon \mathbb{E}_L [T_{\geq \delta}] \quad (5.6.3)$$

Let  $\text{Pr}_U, \text{Pr}_L$  the distributions of  $T_{\leq -\delta}, T_{\geq \delta}$  for adversarial environments  $U, L$  respectively, and let  $\text{Pr}_m$  be the distribution of rounds when  $y_t = +1$  with probability  $1/2$ . From Pinsker's inequality, and denoting by  $\text{KL}(p, q)$  the KL-divergence between distributions  $p, q$ , we have the following:

$$\mathbb{E}_U [T_{\leq -\delta}] \geq \mathbb{E}_m [T_{\leq -\delta}] - T \sqrt{\frac{\text{KL}(\text{Pr}_U, \text{Pr}_m)}{2}} \quad (5.6.4)$$

$$\mathbb{E}_L [T_{\geq \delta}] \geq \mathbb{E}_m [T_{\geq \delta}] - T \sqrt{\frac{\text{KL}(\text{Pr}_L, \text{Pr}_m)}{2}} \quad (5.6.5)$$

Then, from the data processing inequality for the KL-divergence we get:

$$\text{KL}(\text{Pr}_U, \text{Pr}_m) \leq T \text{KL}\left(\text{Bern}\left(\frac{1}{2} + \varepsilon\right), \text{Bern}\left(\frac{1}{2}\right)\right) \leq 4T\varepsilon^2 \quad (5.6.6)$$

$$\text{KL}(\text{Pr}_L, \text{Pr}_m) \leq T \text{KL}\left(\text{Bern}\left(\frac{1}{2} + \varepsilon\right), \text{Bern}\left(\frac{1}{2}\right)\right) \leq 4T\varepsilon^2 \quad (5.6.7)$$

Plugging in Equations (5.6.6) and (5.6.7) in Equations (5.6.4) and (5.6.5) we get:

$$\mathbb{E}_U [T_{\leq -\delta}] \geq \mathbb{E}_m [T_{\leq -\delta}] - T\varepsilon\sqrt{2T}$$

$$\mathbb{E}_L [T_{\geq \delta}] \geq \mathbb{E}_m [T_{\geq \delta}] - T\varepsilon\sqrt{2T}$$

Finally, averaging Equations (5.6.2) and (5.6.3) and using the latter two Equations we get:

$$\max_{\nu \in \{U, L\}} \mathbb{E}_\nu [\mathcal{R}(T)] \geq \frac{\mathbb{E}_U [\mathcal{R}(T)] + \mathbb{E}_L [\mathcal{R}(T)]}{2} \geq \varepsilon \cdot \left(T - 2\varepsilon T \sqrt{2T}\right) \quad (5.6.8)$$

Tuning  $\varepsilon = \frac{1}{3\sqrt{2T}}$  gives the result. ■

## 5.7 DISCUSSION AND OPEN QUESTIONS

In this chapter, we have studied online learning against  $\delta$ -bounded, strategic agents in classification settings, for which we provided the GRINDER algorithm. We complemented our theoretical analysis with simulations, showing first the benefits of our algorithm with a precise oracle, and second, that there are approximation oracles that perform comparably well.

There is a number of interesting questions stemming from the results of this chapter. On a technical level, an interesting research direction is to provide both theoretical results regarding the Stackelberg regret experienced by the learner for the case that she uses *approximation* oracles (e.g., similar to the ones used in the simulations) and experimental ones for different variants of the agents' best-response oracles. In many natural settings (e.g., [Bechavod et al. \(2019\)](#)), the learner can only observe the labels of *some* of the agents. This partial observability imposes an extra challenge, and cannot be handled by our framework currently. Moreover, our current analysis is based on the fact that the learner knows  $\delta$  a priori, but no studies exist currently about an estimation of  $\delta$ . We believe that this can only be achieved through experiments with human subjects in as realistic conditions as possible. Finally, we do believe that the agents' real feature vectors are neither fully stochastic, nor fully adversarial. This model formulation is similar to the one considered in the recent line of works in the online learning literature on the best-of-both-worlds ([Bubeck and Slivkins, 2012](#), [Seldin and Slivkins, 2014](#), [Auer and Chiang, 2016](#), [Seldin and Lugosi, 2017](#)) regret bounds. It would be very interesting to provide theoretical bounds for the Stackelberg regret in such strategic classification settings.

# 6

## Adaptive Discretization for Adversarial Lipschitz Bandits

In Chapter 5, we used multi-armed bandits as a modeling tool for providing learning algorithms that adapt to the agents' strategizing. As a reminder, the adaptive discretization technique that we used for GRINDER was the key technical tool that allowed us to derive data-dependent regret bounds that depended on the sophistication of the strategic agents we encountered. Our adaptive discretization algorithm was tailored (and thus, limited) to the strategic classification setting, since as we saw, in general such settings the loss for the learner is not smooth enough to use standard algorithms from online learning.

In this chapter, we focus on a more fundamental bandits problem which is related to the adap-

tive discretization techniques discussed in Chapter 5. Specifically, we focus on *Lipschitz bandits*, a prominent version of the MAB problem that studies large, structured action spaces such as the  $[0, 1]$  interval. Similar actions have similar rewards, as per Lipschitz-continuity or a similar condition. In applications, actions can correspond to items with feature vectors, such as products, documents or webpages; or to offered prices for buying, selling or hiring; or to different tunings of a complex system such as a datacenter or an ad auction. *Dynamic pricing*, a version where the algorithm is a seller and arms are prices, has attracted much attention on its own.

Extensive literature on Lipschitz bandits centers on two key themes. One is *adaptive discretization* of the action space which gradually “zooms in” on the more promising regions thereof (Kleinberg et al., 2008b, Bubeck et al., 2008, Slivkins et al., 2013, Slivkins, 2014, Munos, 2011, Slivkins, 2011, Valko et al., 2013, Minsker, 2013, Bull, 2015, Ho et al., 2016, Grill et al., 2015). This approach takes advantage of “nice” problem instances – ones in which near-optimal arms are confined to a relatively small region of the action space – while retaining near-optimal worst-case performance. Another theme is relaxing and mitigating the Lipschitz assumptions (Kleinberg et al., 2008b, 2019, Bubeck et al., 2008, 2011a,b, Munos, 2011, Slivkins, 2011, Valko et al., 2013, Minsker, 2013, Bull, 2015, Ho et al., 2016, Grill et al., 2015, Krishnamurthy et al., 2020). The point of departure for all this literature is *uniform discretization* (Kleinberg and Leighton, 2003, Kleinberg, 2004, Kleinberg et al., 2008b, 2019, Bubeck et al., 2008), a simple algorithm which discretizes the action space uniformly and obtains worst-case optimal regret bounds.

All these developments concern the *stochastic* version, in which the rewards of each action are drawn from the same, albeit unknown, distribution in each round. A more general version allows the rewards to be adversarially chosen. Known as *adversarial bandits*, this version is also widely studied in the literature (starting from Auer et al., 2002b), and tends to be much more challenging. The adversarial version of Lipschitz bandits is not understood beyond uniform discretization.

In this chapter, we tackle adversarial Lipschitz bandits, touching upon both themes outlined above, i.e., adaptive discretization and relaxing Lipschitzness. We provide the first algorithm for adaptive discretization for the adversarial case, and obtain instance-dependent regret bounds (i.e., bounds that depend on the properties of the problem instance that are not known initially). Our regret bounds are optimal in the worst case, and improve dramatically when the near-optimal arms

comprise a small region of the action space. In particular, we recover the instance-dependent regret bound for the stochastic version of the problem (Kleinberg et al., 2008b, Bubeck et al., 2008). On the second theme, we find that our analysis does not require the full power of Lipschitz-continuity. In particular, the application to dynamic pricing works without additional Lipschitz assumptions.

## 6.1 CHAPTER OVERVIEW

We are given a set  $\mathcal{A}$  of actions (a.k.a. *arms*), the time horizon  $T$ , and a metric space  $(\mathcal{A}, \mathcal{D})$ , also called the *action space*. The adversary chooses randomized reward functions  $g_1, \dots, g_T : \mathcal{A} \rightarrow [0, 1]$ . In each round  $t$ , the algorithm chooses an arm  $x_t \in \mathcal{A}$  and observes reward  $g_t(x_t) \in [0, 1]$  and nothing else. We focus on the *oblivious adversary*: all reward functions are chosen before round 1. The adversary is restricted in that the expected rewards  $\mathbb{E}[g_t(\cdot)]$  satisfy the Lipschitz condition:<sup>\*</sup>

$$\mathbb{E}[g_t(x) - g_t(y)] \leq \mathcal{D}(x, y) \quad \forall x, y \in \mathcal{A}, t \in [T]. \quad (6.1.1)$$

The algorithm's goal is to minimize *regret*, defined as:

$$R(T) := \sup_{x \in \mathcal{A}} \sum_{t \in [T]} g_t(x) - g_t(x_t). \quad (6.1.2)$$

A problem instance consists of action space  $(\mathcal{A}, \mathcal{D})$  and reward functions  $g_1, \dots, g_T$ . The stochastic version of the problem (*stochastic rewards*) posits that each  $g_t$  is drawn independently from some fixed but unknown distribution  $\mathcal{G}$ . A problem instance is then the tuple  $(\mathcal{A}, \mathcal{D}, \mathcal{G}, T)$ .

The canonical examples are a  $d$ -dimensional unit cube  $(\mathcal{A}, \mathcal{D}) = ([0, 1]^d, \ell_p)$ ,  $p \geq 1$  (where  $\ell_p(x, y) = \|x - y\|_p$  is the  $p$ -norm), and the *exponential tree metric*, where  $\mathcal{A}$  is a leaf set of a rooted infinite tree, and the distance between any two leaves is exponential in the height of their least common ancestor. Our results are equally meaningful for large but finite action sets.

---

<sup>\*</sup>The expectation in (6.1.1) is over the randomness in the reward functions. While adversarial bandits are often defined with deterministic reward functions, it is also common to allow randomness therein, e.g., to include stochastic bandits as a special case. The said randomness is essential to include *stochastic* Lipschitz bandits as a special case. Indeed, for stochastic rewards, (6.1.1) specializes to the Lipschitz condition from prior work on stochastic Lipschitz bandits. A stronger Lipschitz condition  $g_t(x) - g_t(y) \leq \mathcal{D}(x, y)$  is unreasonable for many applications; e.g., if the rewards correspond to user's clicks or other discrete signals, we can only assume Lipschitzness "on average".

Our results are most naturally stated without an explicit Lipschitz constant  $L$ . The latter is implicitly “baked into” the metric  $\mathcal{D}$ , e.g., if the action set is  $[0, 1]$  one can take  $\mathcal{D}(x, y) = L|x - y|$ . However, we investigate the dependence on  $L$  in corollaries. Absent  $L$ , one can take  $\mathcal{D} \leq 1$  w.l.o.g.

**Our Results.** We present `ADVERSARIALZOOMING`, an algorithm for adaptive discretization of the action space. Our main result is a regret bound of the form

$$\mathbb{E}[R(T)] \leq \tilde{\mathcal{O}}(T^{(z+1)/(z+2)}), \quad (6.1.3)$$

where  $z = \text{AdvZoomDim} \geq 0$  is a new quantity called the *adversarial zooming dimension*.<sup>†</sup> This quantity, determined by the problem instance, measures how wide-spread the near-optimal arms are in the action space. In fact, we achieve this regret bound with high probability.

The meaning of this result is best seen via corollaries:

- We recover the optimal *worst-case* regret bounds for the adversarial version. Prior work ([Kleinberg, 2004](#), [Kleinberg et al., 2008b](#), [Bubeck et al., 2008](#)) obtains Equation (6.1.3) for the  $d$ -dimensional unit cube, and more generally Equation (6.1.3) with  $z = \text{CovDim}$ , the covering dimension of the action space. The latter bound is the best possible for any given action space. We recover it in the sense that  $\text{AdvZoomDim} \leq \text{CovDim}$ . Moreover, we match the worst-case optimal regret  $\tilde{\mathcal{O}}(\sqrt{KT})$  for instances with  $K < \infty$  arms and any metric space.
- We recover the optimal *instance-dependent* regret bound from prior work on the stochastic version ([Kleinberg et al., 2008b](#), [Bubeck et al., 2008](#)). This bound is Equation (6.1.3) with  $z = \text{ZoomDim}$ , an instance-dependent quantity called the *zooming dimension*, and it is the best possible for any given action space and any given value of  $\text{ZoomDim}$  ([Slivkins, 2014](#)).  $\text{ZoomDim}$  can be anywhere between 0 and  $\text{CovDim}$ , depending on the problem instance. We prove that, essentially,  $\text{AdvZoomDim} = \text{ZoomDim}$  for stochastic rewards.
- Our regret bound can similarly improve over the worst case even for adversarial rewards. In particular, we may have  $\text{AdvZoomDim} = 0$  for arbitrarily large  $\text{CovDim}$ , even if the reward functions change substantially. Then we obtain  $\tilde{\mathcal{O}}(\sqrt{T})$  regret, as if there were only two arms.

---

<sup>†</sup>As usual, the  $\tilde{\mathcal{O}}(\cdot)$  and  $\tilde{\Omega}(\cdot)$  notation hides polylog( $T$ ) factors.

Adaptive discretization algorithms from prior work (Kleinberg et al., 2008b, Bubeck et al., 2008) do not extend to the adversarial version. For example, specializing to  $K$ -armed bandits with uniform metric  $\mathcal{D} \equiv 1$ , these algorithms reduce to a standard algorithm for stochastic bandits (UCB1, Auer et al., 2002a), which fails badly for many simple instances of adversarial rewards.

The per-round running time of our algorithm is  $\tilde{O}(T^{d/(d+2)})$ , where  $d = \text{CovDim}$ , matching the running time of uniform discretization (with EXP3 (Auer et al., 2002b), a standard algorithm for adversarial bandits). In fact, we obtain a better running time when  $\text{AdvZoomDim} < \text{CovDim}$ .

**Adversarial Zooming Dimension.** The new notion of  $\text{AdvZoomDim}$  can be defined in a common framework with  $\text{CovDim}$  and  $\text{ZoomDim}$  from prior work. All three notions are determined by the problem instance, and talk about *set covers* in the action space. Each notion specifies particular subset(s) of arms to be covered, denoted  $\mathcal{A}_\varepsilon \subset \mathcal{A}$ ,  $\varepsilon > 0$ , and counts how many “small” subsets are needed to cover each  $\mathcal{A}_\varepsilon$ . For a parameter  $\gamma > 0$  called the *multiplier*, the respective “dimension” is

$$\inf \left\{ d \geq 0 : \mathcal{A}_\varepsilon \text{ can be covered with } \gamma \cdot \varepsilon^{-d} \text{ sets of diameter at most } \varepsilon, \quad \forall \varepsilon > 0 \right\}. \quad (6.1.4)$$

Generally, a small “dimension” quantifies the simplicity of a problem instance.

The covering dimension  $\text{CovDim}$  has  $\mathcal{A}_\varepsilon \equiv \mathcal{A}$ . The intuition comes from the  $d$ -dimensional cube, for which  $\text{CovDim} = d$ . <sup>†</sup> Thus, we are looking for the covering property enjoyed by the unit cube. Note that  $\text{CovDim}$  is determined by the action space alone, and is therefore known to the algorithm.

Both  $\text{ZoomDim}$  and  $\text{AdvZoomDim}$  are about covering near-optimal arms. Each subset  $\mathcal{A}_\varepsilon$  comprises all arms that are, in some sense, within  $\varepsilon$  from being optimal. These subsets may be easier to cover compared to  $\mathcal{A}$ ; this may reduce Equation (6.1.4) compared to the worst case of  $\text{CovDim}$ .

The zooming dimension  $\text{ZoomDim}$  is only defined for stochastic rewards. It focuses on the standard notion of *stochastic gap* of an arm  $x$  compared to the best arm:  $\text{Gap}(x) := \max_{y \in \mathcal{A}} \mathbb{E}[g_t(y)] - \mathbb{E}[g_t(x)]$ . Each subset  $\mathcal{A}_\varepsilon$  is defined as the set of all arms  $x \in \mathcal{A}$  with  $\text{Gap}(x) \leq O(\varepsilon)$ .

$\text{AdvZoomDim}$  extends  $\text{ZoomDim}$  as follows. The *adversarial gap* of a given arm  $x$  measures this arm’s

---

<sup>†</sup>More formally, the covering dimension of  $([0, 1]^d, \ell_p)$ ,  $p \geq 1$  is  $d$ , with multiplier  $\gamma = \text{poly}(d, p)$ .

suboptimality compared to the best arm on a given time-interval  $[0, t]$ . Specifically,

$$\text{AdvGap}_t(x) := \frac{1}{t} \max_{y \in \mathcal{A}} \sum_{\tau \in [t]} g_\tau(y) - g_\tau(x). \quad (6.1.5)$$

Given  $\varepsilon > 0$ , an arm  $x$  is called inclusively  $\varepsilon$ -optimal if  $\text{AdvGap}_t(x) < \mathcal{O}(\varepsilon \ln^{3/2} T)$  for some end-time  $t > \Omega(\varepsilon^{-2})$ ; the precise definition is spelled out in Equation (6.3.1). In words, we include all arms whose adversarial gap is sufficiently small at some point in time. It suffices to restrict our attention to a *representative set* of arms  $\mathcal{A}_{\text{repr}} \subset \mathcal{A}$  with  $|\mathcal{A}_{\text{repr}}| \leq O(T^{1+\text{CovDim}})$ , specified in the analysis.<sup>§</sup> Thus, the subset  $\mathcal{A}_\varepsilon$  is defined as the set of all arms  $x \in \mathcal{A}_{\text{repr}}$  that are inclusively  $\varepsilon$ -optimal.

By construction,  $\text{AdvZoomDim} \leq \text{CovDim}$  for any given multiplier  $\gamma > 0$ . For stochastic rewards,  $\text{AdvZoomDim}$  coincides with  $\text{ZoomDim}$  up to a polylog  $(T, |\mathcal{A}_{\text{repr}}|)$  multiplicative change in  $\gamma$ .

The definition of  $\text{AdvZoomDim}$  is quite flexible. First, we achieve the stated regret bound for all  $\gamma > 0$  at once, with a multiplicative  $\gamma^{1/(z+2)}$  dependence thereon. Second, we could relax (6.1.4) to hold only for  $\varepsilon$  smaller than some threshold  $\theta$ ; the regret bound increases by  $+\tilde{O}(\sqrt{T \theta^{-\text{CovDim}}})$ .

**Examples.** We provide a flexible family of examples with small  $\text{AdvZoomDim}$ . Fix an arbitrary action space  $(\mathcal{A}, \mathcal{D})$  and time horizon  $T$ . Consider  $M$  problem instances with stochastic rewards, each with  $\text{ZoomDim} \leq d$ . Construct an instance with adversarial rewards, where each round is assigned in advance to one of these stochastic instances. This assignment can be completely arbitrary: *e.g.*, the stochastic instances can appear consecutively in “phases” of arbitrary duration, or they can be interleaved arbitrarily. Then  $\text{AdvZoomDim} \leq d$  for constant  $M, d$  under some assumptions.

In particular, we allow arbitrary disjoint subsets  $S_1, \dots, S_M \subset \mathcal{A}$  such that each stochastic instance  $i \in [M]$  can behave arbitrarily on  $S_i$  as long as the spread between the largest and smallest mean rewards exceeds a constant. All arms outside  $S_i$  receive the same “baseline” mean reward, which does not exceed the mean rewards inside  $S_i$ . The analysis of this example is somewhat non-trivial, and separate from the main regret bound (6.1.3).

**Application: Adversarial Dynamic Pricing.** Here, an algorithm is a seller with unlimited supply of identical items (*e.g.*, digital items such as songs, movies or software, which can be replicated at

---

<sup>§</sup>Essentially,  $\mathcal{A}_{\text{repr}}$  contains a uniform discretization for scale  $1/T$  and also the local optima for such discretization.

no cost). Each round  $t$  proceeds as follows. A new customer arrives, with *private value*  $v_t \in [0, 1]$  that is not known to the algorithm. The algorithm chooses a price  $x_t \in [0, 1]$  and offers one item for sale at this price. The customer buys if and only if  $x_t \leq v_t$ . The algorithm maximizes revenue, *i.e.*, its reward is  $g_t(x_t) = x_t \cdot \mathbb{1}\{x_t \leq v_t\}$ . The private values  $v_1, \dots, v_T$  are fixed in advance, possibly using randomization. In the stochastic version, they come from the same distribution (which is not known to the algorithm). Prior work on algorithms for this problem is limited to uniform discretization ([Kleinberg and Leighton, 2003](#)), which obtains a worst-case optimal regret rate. However, adaptive discretization was only known for the stochastic version.

If the dynamic pricing problem satisfied the Lipschitz assumption (Eq. (6.1.1)), then it would be a special case of adversarial Lipschitz bandits, for which ADVERSARIALZOOMING incurs regret given by Equation (6.1.3). In fact, this bound holds even without any additional assumptions such as Equation (6.1.1). This is due to the fact that the dynamic pricing problem inherently satisfies a “one-sided” Lipschitz condition (see Equation (6.3.7)), which suffices for our analysis. AdvZoomDim can improve from the worst case of CovDim = 1, *e.g.*, in the family of examples described above. We obtain regret  $\tilde{O}(T^{2/3})$  in the worst case, which is optimal even for stochastic dynamic pricing ([Kleinberg and Leighton, 2003](#)).

**Challenges and Techniques.** We build on the high-level idea of *zooming* from prior work on the stochastic version ([Kleinberg et al., 2008b](#), [Bubeck et al., 2008](#)), but provide a very different implementation of this idea. At each round, we maintain a partition of the action space into “active regions”, and refine this partition adaptively over time. We “zoom in” on a given region by partitioning it into several “children” of half the diameter; we do it only if the sampling uncertainty goes below the region’s diameter. In each round, we select an active region according to (a variant of) a standard algorithm for bandits with a fixed action set, and then sample an arm from the selected region according to a fixed, data-independent rule. The standard algorithm we use is EXP3.P ([Auer et al., 2002b](#)); prior work on stochastic rewards used UCB1 ([Auer et al., 2002a](#)).

Adversarial rewards bring about several challenges compared to the stochastic version. First, the technique in EXP3.P does not easily extend to variable number of arms (when the action set is increased via “zooming”), whereas the technique in UCB1 does, for stochastic rewards. Second, the sampling uncertainty is not directly related to the total probability mass allocated to a given region.

In contrast, this relation is straightforward and crucial for the stochastic version. Third, the adversarial gap is much more difficult to work with. Indeed, the analysis for stochastic rewards relies on two easy but crucial steps — bounding the gap for regions with small sampling uncertainty, and bounding the “total damage” inflicted by all small-gap arms — which break adversarial rewards.

These challenges prompt substantial complications in the algorithm and the analysis. For example, to incorporate “children” into the multiplicative weights analysis, we split the latter into two steps: first we update the weights, then we add the children. To enable the second step, we partition the parent’s weight equally among the children. Effectively, we endow each child with a copy of the parent’s data, and argue that the latter is eventually diluted by the child’s own data.

Another example of the challenges faced is that to argue that we only “zoom in” if the parent has small adversarial gap, we need to enhance the “zoom-in rule”: in addition to the “aggregate” rule (the sampling uncertainty must be sufficiently small), we need the “instantaneous” one: the current sampling probability must be sufficiently large, and it needs to be formulated in just the right way. Then, we need to be much more careful about deriving the “zooming invariant”, a crucial property of the partition enforced by the “zoom-in rule”. In turn, this derivation necessitates the algorithm’s parameters to change from round to round, which further complicates the multiplicative weights analysis.

An important part of our contribution is formalizing what we mean by “nice” problem instances, and boiling the analysis down to an easily interpretable notion such as `AdvZoomDim`.

**Remarks.** We obtain an *anytime* version, with similar regret bounds for all time horizons  $T$  at once, using the standard *doubling trick*: in each phase  $i \in \mathbb{N}$ , we restart the algorithm with time horizon  $T = 2^i$ . The only change is that the definition of `AdvZoomDim` redefines  $\mathcal{A}_\varepsilon$  to be the set of all arms that are inclusively  $\varepsilon$ -optimal within some phase.

Our regret bound depends sublinearly on the *doubling constant*  $C_{\text{dbl}}$ : the smallest  $C \in \mathbb{N}$  such that any ball can be covered with  $C$  sets of at most half the diameter. Note that  $C_{\text{dbl}} = 2^d$  for a  $d$ -dimensional unit cube, or any subset thereof. The doubling constant has been widely used in theoretical computer science, e.g., see Kleinberg et al. (2009) for references.

### 6.1.1 RELATED WORK

Lipschitz bandits are introduced in (Agrawal, 1995) for action space  $[0, 1]$ , and optimally solved in the worst case via uniform discretization in (Kleinberg, 2004). Adaptive discretization is introduced in (Kleinberg et al., 2008b, Bubeck et al., 2008), and subsequently extended to contextual bandits (Slivkins, 2014), ranked bandits (Slivkins et al., 2013), and contract design for crowdsourcing (Ho et al., 2016). The terms “zooming algorithm/dimension” trace back to Kleinberg et al. (2008b). Kleinberg et al. (2008b, 2019) consider regret rates with instance-dependent constant (e.g.,  $\log(t)$  for finitely many arms), and build on adaptive discretization to characterize worst-case optimal regret rates for any given metric space. Pre-dating the work on adaptive discretization, Kocsis and Szepesvari (2006), Pandey et al. (2007), Munos and Coquelin (2007) allow a “taxonomy” on arms without any numerical information (and without any non-trivial regret bounds).

Several papers recover adaptive discretization guarantees under mitigated Lipschitz conditions: when Lipschitzness only holds near the best arm  $x^*$  or when one of the two arms is  $x^*$  (Kleinberg et al., 2008b, Bubeck et al., 2008); when the algorithm is only given a taxonomy of arms, but not the metric (Slivkins, 2011, Bull, 2015); when the actions correspond to contracts offered to workers, and no Lipschitzness is assumed (Ho et al., 2016), and when expected rewards are Hölder-smooth with an unknown exponent (Locatelli and Carpentier, 2018).

In other work on mitigating Lipschitzness, Bubeck et al. (2011b) recover the optimal worst-case bound with unknown Lipschitz constant. Munos (2011), Valko et al. (2013), Grill et al. (2015) consider adaptive discretization in the “pure exploration” version, and allow for a parameterized class of metrics with unknown parameter. Krishnamurthy et al. (2020) posit a weaker, “smoothed” benchmark and get adaptive discretization-like regret bounds without any Lipschitz assumptions.

All work discussed above assumes stochastic rewards. Adaptive discretization is extended to expected rewards with bounded change over time (Slivkins, 2014), and to a version with ergodicity and mixing assumptions (Azar et al., 2014). For Lipschitz bandits with adversarial rewards, the uniform discretization approach easily extends (Kleinberg, 2004), and *nothing else is known*.<sup>¶</sup>

Dynamic pricing has a long history in operations research (survey: Boer, 2015). The regret-

---

<sup>¶</sup>We note that Maillard and Munos (2010) achieve  $O(\sqrt{T})$  regret for the full-feedback version.

minimizing version was optimally solved in the worst case in (Kleinberg and Leighton, 2003), via uniform discretization and an intricate lower bound. Extensions to limited supply were studied in (Besbes and Zeevi, 2009, 2011, Babaioff et al., 2015, Wang et al., 2014, Badanidiyuru et al., 2018). Departing from the stochastic version, Besbes and Zeevi (2011), Keskin and Zeevi (2017a) allow bounded change over time. Cohen et al. (2020), Lobel et al. (2018), Paes Leme and Schneider (2018), Liu et al. (2021), Krishnamurthy et al. (2021) study the contextual version of the problem, where the value for an item is linearly dependent on a publicly observed context, and a common ground-truth value-vector which is the same for all the agents. The interested reader can find an extensive discussion on these settings in Part III.

*Learning to bid* in adversarial repeated auctions (discussed extensively in Chapter 10) is related as an online learning problem with a “continuous” action set. However, prior work on regret-minimizing formulations of learning-to-bid is incomparable with ours, as it assumes more-than-bandit feedback. Specifically, Weed et al. (2016) posit full feedback whenever the algorithm wins the second-price auction, while Feng et al. (2018) do so for the generalized second price auction. Han et al. (2020) assume full feedback in a learning setting for first-price auctions, which essentially obviates the need for adaptive discretization. Instead, in our model the algorithm only observes whether a purchase happened.

Cesa-Bianchi et al. (2017) obtain improved regret rates for some other adversarial continuum-armed bandit problems. Their results, which use different techniques than ours, require more-than-bandit feedback of a specific shape; this holds e.g., for learning reserve prices in contextual second-price auctions, but not for dynamic pricing.

While Lipschitz bandits only capture “local” similarity between arms, other structural models such as convex bandits (e.g., Flaxman et al., 2005, Agarwal et al., 2013, Bubeck et al., 2017) and linear bandits (e.g., Dani et al., 2008, Abernethy et al., 2008, Abbasi-Yadkori et al., 2011) allow for *long-range inferences*: by observing some arms, an algorithm learns something about other arms that are far away. This is why  $\tilde{O}(\sqrt{T})$  regret rates are achievable in adversarial versions of these problems, via different techniques.

## 6.2 OUR ALGORITHM: ADVERSARIAL ZOOMING

For ease of presentation, we present the algorithm for the special case of  $d$ -dimensional unit cube,  $(\mathcal{A}, \mathcal{D}) = ([0, 1]^d, \ell_\infty)$ . Our algorithm partitions the action space into axis-parallel hypercubes. More specifically, we consider a rooted directed tree, called the *zooming tree*, whose nodes correspond to axis-parallel hypercubes in the action space. The root is  $\mathcal{A}$ , and each node  $u$  has  $2^d$  children that correspond to its quadrants. For notation,  $\mathcal{U}$  is the set of all tree nodes,  $\mathcal{C}(u)$  is the set of all children of node  $u$ , and  $L(u) = \max_{x,y \in u} \mathcal{D}(x, y)$  is its diameter in the metric space; w.l.o.g.  $L(\cdot) \leq 1$ .

On a high level, the algorithm operates as follows. We maintain a set  $A_t \subset \mathcal{U}$  of tree nodes in each round  $t$ , called *active nodes*, which partition the action space. We start with a single active node, the root. After each round  $t$ , we may choose some node(s)  $u$  to “zoom in” on according to the *zoom-in rule*, in which case we de-activate  $u$  and activate its children. We denote this decision with  $z_t(u) = \mathbb{1}\{\text{zoom in on } u \text{ at round } t\}$ . In each round, we choose an active node  $U_t$  according to the *selection rule*. Then, we choose a representative arm  $x_t = \text{repr}_t(U_t) \in U_t$  to play in this round. The latter choice can depend on  $t$ , but not on the algorithm’s observations; the choice could be randomized, e.g., we could choose uniformly at random from  $U_t$ .

The main novelty of our algorithm is in the zoom-in rule. However, presenting it requires some scaffolding: we need to present the rest of the algorithm first. The selection rule builds on EXP3 ([Auer et al., 2002b](#)), a standard algorithm for adversarial bandits. We focus on EXP3.P, a variant that uses “optimistic” reward estimates, the inverse propensity score (IPS) plus a “confidence term” (see Equation (6.2.1)). This is because we need a similar “confidence term” from the zooming rule to “play nicely” with the EXP3 machinery. If we never zoomed in *and* used  $\eta = \eta_t$  for multiplicative updates in each round, then our algorithm would essentially coincide with EXP3.P. Specifically, we maintain weights  $w_{t,\eta}(u)$  for each active node  $u$  and round  $t$ , and update them multiplicatively, as per Equation (6.2.2). In each round  $t$ , we define a probability distribution  $p_t$  on the active nodes, proportional to the weights  $w_{t,\eta_t}$ . We sample from this distribution, mixing in some low-probability uniform exploration.

We are ready to present the pseudocode (Algorithm 6.1). The algorithm has parameters  $\beta_t, \gamma_t, \eta_t \in$

---

**ALGORITHM 6.1: ADVERSARIALZOOMING**


---

```

1 Parameters:  $\beta_t, \gamma_t, \eta_t \in (0, 1/2]$  for each round  $t$ .
2 Variables: active nodes  $A_t \subset \mathcal{U}$ , weights  $w_{t,\eta} : \mathcal{U} \rightarrow (0, \infty]$   $\forall$  round  $t$ ,  $\eta \in (0, 1/2]$ 
3 Initialization:  $w_1(\cdot) = 1$  and  $A_1 = \{\text{root}\}$  and  $\beta_1 = \gamma_1 = \eta_1 = 1/2$ .
4 for  $t = 1, \dots, T$  do
5    $p_t \leftarrow \{\text{distribution } p_t \text{ over } A_t, \text{ proportional to weights } w_{t,\eta_t}\}$ .
6   Add uniform exploration: distribution  $\pi_t(\cdot) \leftarrow (1 - \gamma_t) p_t(\cdot) + \gamma_t / |A_t|$  over  $A_t$ .
7   Select a node  $U_t \sim \pi_t(\cdot)$ , and then its representative:  $x_t = \text{repr}(U_t)$ . // selection rule
8   Observe the reward  $g_t(x_t) \in [0, 1]$ .
9   for  $u \in A_t$  do
10    |
11    |    $\widehat{g}_t(u) = \frac{g_t(x_t) \cdot \mathbb{1}\{u = U_t\}}{\pi_t(u)} + \frac{(1 + 4 \log T) \beta_t}{\pi_t(u)}$  // IPS + "conf term" (6.2.1)
12    |    $w_{t+1,\eta}(u) = w_{t,\eta}(u) \cdot \exp(\eta \cdot \widehat{g}_t(u))$ ,  $\forall \eta \in (0, 1/2]$  // MW update (6.2.2)
13    |   if  $z_t(u) = 1$  then // zoom-in rule
14    |   |    $A_{t+1} \leftarrow A_t \cup \mathcal{C}(u) \setminus \{u\}$  // activate children of  $u$ , deactivate  $u$ 
15    |   |    $w_{t+1}(v) = w_{t+1}(u) / |\mathcal{C}(u)|$  for all  $v \in \mathcal{C}(u)$ . // split the weight

```

---

$(0, 1/2]$  for each round  $t$ ; we fix them later in the analysis as a function of  $t$  and  $|A_t|$ . Their meaning is that  $\beta_t$  drives the “confidence term”,  $\gamma_t$  is the probability of uniform exploration, and  $\eta_t$  parameterizes the multiplicative update. To handle the changing parameters  $\eta_t$ , we use a trick from (Bubeck, 2010, Bubeck and Cesa-Bianchi, 2012): conceptually, we maintain the weights  $w_{t,\eta}$  for all values of  $\eta$  simultaneously, and plug in  $\eta = \eta_t$  only when we compute distribution  $p_t$ . Explicitly maintaining all these weights is cleaner and mathematically well-defined, so this is what our pseudocode does.

For the subsequent developments, we need to carefully account for the ancestors of the currently active nodes. Suppose node  $u$  is active in round  $t$ , and we are interested in some earlier round  $s \leq t$ . Exactly one ancestor of  $u$  in the zooming tree has been active then; we call it the *active ancestor* of  $u$  and denote  $\text{act}_s(u)$ . If  $u$  itself was active in round  $s$ , we write  $\text{act}_s(u) = u$ .

For computational efficiency, we do not explicitly perform the multiplicative update of Equation (6.2.2). Instead, we recompute the weights  $w_{t,\eta_t}$  from scratch in each round  $t$ , using the following characterization:

**Lemma 6.1.** Let  $\mathcal{C}_{\text{prod}}(u) = \prod_v |\mathcal{C}(v)|$ , where  $v$  ranges over all ancestors of node  $u$  in the zooming tree (not

including  $u$  itself). Then for all nodes  $u \in A_t$ , rounds  $t$ , and parameter values  $\eta \in (0, 1/2]$ ,

$$w_{t+1,\eta}(u) = \mathcal{C}_{\text{prod}}^{-1}(u) \cdot \exp \left( \eta \sum_{\tau \in [t]} \hat{g}_\tau(\text{act}_\tau(u)) \right). \quad (6.2.3)$$

*Proof.* We prove the lemma by using induction on the zooming tree at round  $t + 1$ , specifically on the path from node  $\text{root}$  from  $u$ . For the base case, if the set of active nodes at round  $t + 1$  included only node  $\text{root}$  then from Equation (6.2.2):

$$w_{t+1,\eta}(\text{root}) = w_{t,\eta}(\text{root}) \exp (\eta \hat{g}_t(\text{root})) = \exp \left( \eta \sum_{\tau \in [t]} \hat{g}_\tau(\text{root}) \right).$$

So Equation (6.2.3) holds, since  $\mathcal{C}_{\text{prod}}(\text{root}) = 1$ , as  $\text{root}$  is the root node of the tree (i.e., it has no ancestors). We next assume that the active node at round  $t + 1$  is  $\xi$  and that Equation (6.2.3) holds for the zooming tree until node  $\xi$ , such that  $\xi = \text{parent}(u)$ , i.e.:

$$w_{t+1,\eta}(\xi) = \frac{1}{\mathcal{C}_{\text{prod}}(\xi)} \cdot \exp \left( \eta \sum_{\tau \in [t]} \hat{g}_\tau(\text{act}_\tau(\xi)) \right). \quad (6.2.4)$$

Next assume that the active node at round  $t + 1$  is node  $u$ . Then, from Equation (6.2.3)

$$w_{t+1,\eta}(u) = w_{t,\eta}(u) \exp (\eta \hat{g}_t(u)) = w_{t,\eta}(\text{act}_t(u)) \exp (\eta \hat{g}_t(\text{act}_t(u)))$$

since by definition for all the rounds  $\tau \leq t + 1$  during which  $u$  is active  $\text{act}_\tau(u) = u$ . By definition, from round  $\tau_0(u)$  until round  $t + 1$  node  $u$  has not been zoomed-in. Hence, by the weight-update rule for nodes that are not further zoomed-in we have that:

$$\begin{aligned} w_{t+1,\eta}(u) &= w_{\tau_0(u)-1,\eta}(u) \cdot \prod_{\tau=\tau_0(u)}^t \exp (\eta \hat{g}_\tau(\text{act}_\tau(u))) \\ &= w_{\tau_0(u)-1,\eta}(u) \cdot \exp \left( \eta \sum_{\tau=\tau_0(u)}^t \hat{g}_\tau(\text{act}_\tau(u)) \right). \end{aligned} \quad (6.2.5)$$

Since  $\tau_0(u) - 1$  is the last round of  $\xi$ 's lifetime, it is the round that  $\xi$  got zoomed-in, and the weight

of  $u$  was initialized to be  $1/|\mathcal{C}(\xi)|$  the weight of  $\xi$ . Hence, Equation (6.2.5) becomes:

$$w_{t+1,\eta}(u) = \frac{1}{|\mathcal{C}(\xi)|} w_{\tau_1(\xi),\eta}(\xi) \cdot \exp \left( \eta \sum_{\tau=\tau_0(u)}^t \hat{g}_\tau(\text{act}_\tau(u)) \right)$$

But for the rounds where node  $\xi$  was active, Equation (6.2.4) was true for node  $\xi$ . Since  $\xi = \text{parent}(u)$ , then  $\text{act}_\tau(\xi) = \text{act}_\tau(u), \forall \tau \leq \tau_0(u)$ . Hence, using Equation (6.2.4) in the latter, we obtain:

$$\begin{aligned} w_{t+1,\eta}(u) &= \frac{1}{|\mathcal{C}(\xi)| \cdot \mathcal{C}_{\text{prod}}(\xi)} \cdot \exp \left( \eta \sum_{\tau \in [\tau_1(\xi)]} \hat{g}_\tau(\text{act}_\tau(u)) \right) \cdot \exp \left( \eta \sum_{\tau=\tau_0(u)}^t \hat{g}_\tau(\text{act}_\tau(u)) \right) \\ &= \frac{1}{\mathcal{C}_{\text{prod}}(u)} \cdot \exp \left( \eta \sum_{\tau \in [t]} \hat{g}_\tau(\text{act}_\tau(u)) \right). \end{aligned}$$

where for the penultimate equation, we used the definition of  $\mathcal{C}_{\text{prod}}(u)$ . ■

**Remarks.** We make no restriction on how many nodes  $u$  can be “zoomed-in” in any given round. However, our analysis implies that we cannot immediately zoom in on any “newborn children”.

When we zoom in on a given node, we split its weight equally among its children. Maintaining the total weight allows the multiplicative weights analysis to go through, and the equal split allows us to conveniently represent the weights in Lemma 6.1 (which is essential in the multiplicative weights analysis, too). An undesirable consequence is that we effectively endow each child with a copy of the parent’s data; we deal with it in the analysis via Equation (6.4.4).

We now explain the confidence term in Equation (6.2.1). Define the *total confidence term*

$$\text{conf}_t^{\text{tot}}(u) := 1/\beta_t + \sum_{\tau \in [t]} \beta_\tau / \pi_\tau(\text{act}_\tau(u)). \quad (6.2.6)$$

Essentially, we upper-bound the cumulative gain from node  $u$  up to time  $t$  using

$$\text{conf}_t^{\text{tot}}(u) + \sum_{\tau \in [t]} \text{IPS}_t(\text{act}_\tau(u)), \quad \text{where} \quad \text{IPS}_t(u) := g_t(x_t) \cdot \mathbb{1}\{u = U_t\} / \pi_t(u). \quad (6.2.7)$$

The  $+4 \log T$  term in Equation (6.2.1) is needed to account for the ancestors later in the analysis; it

would be redundant if there were no zooming and the active node set were fixed throughout.

**The Zoom-In Rule.** Intuitively, we want to zoom in on a given node  $u$  when its per-round sampling uncertainty gets smaller than its diameter  $L(u)$ , in which case exploration at the level of  $u$  is no longer productive. A natural way to express this is  $\text{conf}_t^{\text{tot}}(u) \leq t \cdot L(u)$ , which we call the *aggregate* zoom-in rule. However, it does not suffice: we also need an *instantaneous* version which asserts that the current sampling probability is large enough. Making this precise is somewhat subtle. Essentially, we lower-bound  $\text{conf}_t^{\text{tot}}(u)$  as a sum of “instantaneous confidence terms”

$$\text{conf}_{\tau}^{\text{inst}}(u) := \tilde{\beta}_{\tau} + \frac{\beta_{\tau}}{\pi_{\tau}(\text{act}_{\tau}(u))}, \quad \tau \in [t], \quad (6.2.8)$$

where  $\tilde{\beta}_{\tau} \in (0, 1/2]$  are new parameters. We require each such term to be at most  $L(u)$ . In fact, we require a stronger upper bound  $e^{L(u)} - 1$ , which plugs in nicely into the multiplicative weights argument, and implies an upper bound of  $L(u)$ . Thus, the zoom-in rule is as follows:

$$z_t(u) := \mathbb{1} \left\{ \text{conf}_t^{\text{inst}}(u) \leq e^{L(u)} - 1 \right\} \cdot \mathbb{1} \left\{ \text{conf}_t^{\text{tot}}(u) \leq t \cdot L(u) \right\} \quad (6.2.9)$$

Parameters  $\tilde{\beta}_{\tau}$  must be well-defined for all  $\tau \in [0, T]$  and satisfy the following, for any rounds  $t < t'$ :

$$\left\{ \begin{array}{l} \tilde{\beta}_{\tau} \text{ decreases in } \tau \\ \tilde{\beta}_t \geq \beta_t \end{array} \right\} \text{ and } \int_t^{t'} \tilde{\beta}_{\tau} d\tau \leq \frac{1}{\beta_{t'}} - \frac{1}{\beta_t}. \quad (6.2.10)$$

We cannot obtain the third condition of Equation (6.2.10) with equality because parameters  $\beta_t$  and  $\tilde{\beta}_t$  depend on  $|A_t|$ , and the latter is not related to  $t$  with a closed form solution.

### 6.3 ALGORITHM'S GUARANTEES

**Running Time.** The per-round running time of the algorithm is  $\tilde{O}(T^{d/(d+2)})$ , where  $d = \text{CovDim}$ . Indeed, given Lemma 6.1, in each round  $t$  of the algorithm we only need to compute the weight  $w_{t,\eta}(\cdot)$  for all active nodes and one specific  $\eta = \eta_t$ . This takes only  $O(1)$  time per node (since we can maintain the total estimated reward  $\sum_{\tau \in [t]} \hat{g}_{\tau}(\text{act}_{\tau}(u))$  separately). So, the per-round running time is  $O(|A_T|)$ , which is at most  $\tilde{O}(T^{d/(d+2)})$ , as we prove in Lemma 6.17. Moreover, we obtain

an improved bound on  $|A_T|$  (and hence on the running time) when  $\text{AdvZoomDim} < \text{CovDim}$  and the doubling constant  $C_{\text{dbl}}$  is  $\text{polylog}(T)$ , see Equation (6.5.63).

**Regret Bounds.** Our regret bounds are broken into three steps. First, we state the “raw” regret bound in terms of the algorithm’s parameters, with explicit assumptions thereon. Second, we tune the parameters and derive the “intermediate” regret bound of the form  $\tilde{O}(\sqrt{T|A_T|})$ . Third, we derive the “final” regret bound, upper-bounding  $|A_T|$  in terms of  $\text{AdvZoomDim}$ . For ease of presentation, we use failure probability  $\delta = T^{-2}$ ; for any known  $\delta > 0$ , regret scales as  $\log^{1/\delta}$ . The covering dimension is denoted  $d$ , for some constant multiplier  $\gamma_0 > 0$  (we omit the  $\log(\gamma_0)$  dependence). Precisely, an inclusively  $\varepsilon$ -optimal arm in the definition of  $\text{AdvZoomDim}$  has:

$$\text{AdvGap}_t(\cdot) < 30\varepsilon \ln(T) \sqrt{d \ln(C_{\text{dbl}} \cdot T)} \quad \text{for some end-time } t > \varepsilon^{-2}/9. \quad (6.3.1)$$

### Theorem 6.1: ADVERSARIALZOOMING Regret

Assume the sequences  $\{\eta_t\}$  and  $\{\beta_t\}$  are decreasing in  $t$ , and satisfy

$$\eta_t \leq \beta_t \leq \gamma_t / |A_t| \quad \text{and} \quad \eta_t (1 + \beta_t(1 + 4 \log T)) \leq \gamma_t / |A_t|. \quad (6.3.2)$$

With probability at least  $1 - T^{-2}$ , ADVERSARIALZOOMING satisfies

$$R(T) \leq \mathcal{O}(\ln T) \left( \sqrt{dT} + \frac{1}{\beta_T} + \frac{\ln(C_{\text{dbl}} \cdot |A_T|)}{\eta_T} + \sum_{t \in [T]} \beta_t + \gamma_t \ln T \right) \quad (6.3.3)$$

$$\leq \mathcal{O}\left(\sqrt{T|A_T|}\right) \cdot \ln^2(T) \sqrt{d \ln(T|A_T|) \ln(C_{\text{dbl}}|A_T|)} \quad (\text{parameter tuning}) \quad (6.3.4)$$

$$\leq \mathcal{O}\left(T^{\frac{z+1}{z+2}}\right) \cdot \left(d^{1/2} (\gamma C_{\text{dbl}})^{1/(z+2)} \ln^5 T\right), \quad (6.3.5)$$

where  $d = \text{CovDim}$  and  $z = \text{AdvZoomDim}$  with multiplier  $\gamma > 0$ . The parameters in (6.3.4) are:

$$\begin{aligned} \beta_t &= \tilde{\beta}_t = \eta_t = \sqrt{2 \ln(|A_t| \cdot T^3) \ln(C_{\text{dbl}} \cdot |A_t|)} / \sqrt{t |A_t| d \cdot \ln^2 T}, \\ \gamma_t &= (2 + 4 \log T) |A_t| \cdot \beta_t. \end{aligned} \quad (6.3.6)$$

**Remark 6.1.** We can relax the definition of  $\text{AdvZoomDim}$  so that (6.1.4) needs to hold only for scales  $\varepsilon$

smaller than some threshold  $\theta$ . Then we obtain the regret bound in (6.3.5) plus  $\tilde{O}\left(\sqrt{T\theta^{-\text{CovDim}}}\right)$ .

**Application to Adversarial Dynamic Pricing.** According to the problem statement, we have a bandit problem with action set  $\mathcal{A} = [0, 1]$  and a one-sided Lipschitz condition

$$g_t(x) - g_t(x') \leq x - x' \quad \text{for any prices } x, x' \in [0, 1] \text{ with } x > x'. \quad (6.3.7)$$

This condition holds because selling at a given price  $x$  implies selling at any lower price  $x'$ . We specialize the algorithm slightly: the representative arm  $\text{repr}_t(u)$  is defined as the lowest point of the price interval that node  $u$  corresponds to. We observe that our analysis uses the Lipschitz condition only through a specific corollary (6.4.9), and derive this corollary from (6.3.7). We obtain:

**Corollary 6.1.** *Adversarial dynamic pricing satisfies Theorem 6.1 without any additional assumptions such as the two-sided Lipschitz condition (6.1.1).*

**Special cases.** First, we argue that for stochastic rewards  $\text{AdvZoomDim}$  coincides with the zooming dimension  $\text{ZoomDim}$  from prior work, up to a small change in the multiplier  $\gamma$ . We specify the latter by putting it in the subscript. The key is to relate each arm's stochastic gap to its adversarial gap.

**Lemma 6.2.** *Consider an instance of Lipschitz bandits with stochastic rewards. For any  $\gamma > 0$ , with probability at least  $1 - 1/T$  it holds that:*

$$\text{ZoomDim}_{\gamma, f} \leq \text{AdvZoomDim}_{\gamma, f} \leq \text{ZoomDim}_\gamma, \quad \text{where } f = (O(\text{poly}(d) \ln^3 T))^{\log(C_{\text{dbl}}) - \text{ZoomDim}_\gamma}.$$

This lemma holds for any representative set  $\mathcal{A}_{\text{repr}}$ . Then the base in factor  $f$  scales with  $\ln(|\mathcal{A}_{\text{repr}}|)$ .

Second, for problem instances with  $K < \infty$  arms, we recover the standard  $\tilde{O}(\sqrt{KT})$  regret bound by observing that any problem instance has  $\text{AdvZoomDim} = 0$  with multiplier  $\gamma = K$  and  $\mathcal{A}_{\text{repr}} = [K]$ . **ADVERSARIALZOOMING** satisfies  $R(T) \leq \mathcal{O}(\sqrt{KT} \cdot \sqrt{C_{\text{dbl}}} \cdot \ln^5 T)$  w.h.p.

Third, we analyze the dependence on the Lipschitz constant. Fix a problem instance, and multiply the metric by some  $L > 1$ . The Lipschitz condition (6.1.1) still holds, and the definition of  $\text{AdvZoomDim}$  implies that regret scales as  $L^{z/(z+2)}$ . This is optimal in the worst case by prior work.<sup>||</sup>

---

<sup>||</sup>For a formal statement, consider the unit cube with  $\mathcal{A} = [0, 1]^d$  and metric  $D(x, y) = L \cdot \|x - y\|_p$ , for

**Corollary 6.2.** Fix a problem instance and a multiplier  $\gamma > 0$ , and let  $R_\gamma(T)$  denote the right-hand side of (6.3.5). Consider a modified problem instance with metric  $\mathcal{D}' = L \cdot \mathcal{D}$ , for some  $L \geq 1$ . Then ADVERSARIALZOOMING satisfies  $R(T) \leq L^{z/(z+2)} \cdot R_\gamma(T)$ , with probability at least  $1 - T^{-2}$ .

## 6.4 REGRET ANALYSIS OUTLINE

We outline the key steps and the proof structure; the lengthy details are in the next section. For ease of presentation, we focus on the  $d$ -dimensional unit cube  $(\mathcal{A}, \mathcal{D}) = ([0, 1]^d, \ell_\infty)$ .

We start with some formalities. First, we posit a *representative arm*  $\text{repr}_t(u) \in u$  for each tree node  $u$  and each round  $t$ , so that  $x_t = \text{repr}_t(U_t)$ . W.l.o.g., all representative arms are chosen before  $t = 1$ . Thus, we can endow  $u$  with rewards  $g_t(u) := g_t(\text{repr}_t(u))$ . Second, let  $\text{OPT}_S(u) \in \arg \max_{x \in u} \sum_{t \in S} g_t(x)$  be the best arm in  $u$  over the set  $S$  of rounds (ties broken arbitrarily). Let  $\text{OPT}_S = \text{OPT}_S(\mathcal{A})$  be the best arm over  $S$ . Let  $u_t^*$  be the active node at  $t$  which contains  $\text{OPT}_{[t]}$ .

The representative set  $\mathcal{A}_{\text{repr}} \subset \mathcal{A}$  (used in the definition of AdvZoomDim) consists of arms  $\text{repr}(u), x_{[t]}^*(u)$  for all tree nodes of height at most  $1 + \log T$  and all rounds  $t$ . Only these arms are invoked by the algorithm or the analysis. This enables us to transition to deterministic rewards that satisfy a certain “per-realization” Lipschitz property (Equation (C.1.1) in the appendix).

**Part I: Properties of the Zoom-In Rule.** This part depends on the zoom-in rule, but not on the selection rule, *i.e.*, it works no matter how distribution  $\pi_t$  is chosen. First, the zoom-in rule ensures that all active nodes satisfy the following property, called the *zooming invariant*:

$$\text{conf}_t^{\text{tot}}(u) \geq (t - 1) \cdot L(u) \quad \text{if node } u \text{ is active in round } t \quad (6.4.1)$$

It is proved by induction on  $t$ , using the fact that when a node does *not* get zoomed-in, this is because either instantaneous or the aggregate zoom-in rule does not apply.

Let us characterize the *lifespan* of node  $u$ : the time interval  $[\tau_0(u), \tau_1(u)]$  during which the node some constants  $d \in \mathbb{N}$  and  $p \geq 1$ . Then the worst-case optimal regret rate is  $\tilde{\mathcal{O}}(L^{d/(d+2)} \cdot T^{(d+1)/(d+2)})$ . The proof for  $d = 1$  can be found in (Slivkins, 2019, Chapter 4.1); the proof for  $d > 1$  can be derived similarly.

is active. We lower-bound the deactivation time, using the instantaneous zoom-in rule:

$$\text{node } u \text{ is zoomed-in} \Rightarrow \tau_1(u) \geq 1/L(u). \quad (6.4.2)$$

It follows that only nodes of diameter  $L(\cdot) \geq 1/2T$  can be activated. Next, we show that a node's deactivation time is (approx.) at least twice as the parent's:

$$\text{node } u \text{ is zoomed-in} \Rightarrow \tau_1(u) \geq 2\tau_1(\text{parent}(u)) - 2. \quad (6.4.3)$$

We use this to argue that a node's own datapoints eventually drown out those inherited from the parent when the node was activated. Specifically:

$$\text{node } u \text{ is active at time } t \Rightarrow \frac{1}{t} \sum_{\tau \in [t]} L(\text{act}_\tau(u)) \leq 4 \log(T) \cdot L(u). \quad (6.4.4)$$

Next, we prove that the total probability mass spent on a zoomed-in node must be large:

$$\text{node } u \text{ is zoomed-in} \Rightarrow \mathcal{M}(u) := \sum_{\tau=\tau_0(u)}^{\tau_1(u)} \pi_\tau(u) \geq \frac{1}{9L^2(u)} \quad (6.4.5)$$

This statement is essential for bounding the number of active nodes in Part IV. To prove it, we apply both the zooming invariant (6.4.1) and the (aggregate) zooming rule. Finally, the instantaneous zoom-in rule implies that the zoomed-in node is chosen with large probability:

$$\text{node } u \text{ is zoomed-in at round } t \Rightarrow \pi_t(u)/\pi_t(u_t^*) \geq \beta_t^2/e^{L(u)}. \quad (6.4.6)$$

**Part II: Multiplicative Weights.** This part depends on the selection rule, but not on the zooming rule: it works regardless of how  $z_t(u)$  is defined. We analyze the following potential function:  $\Phi_t(\eta) = \left( \frac{1}{|A_t|} \sum_{u \in A_t} w_{t+1,\eta}(u) \right)^{1/\eta}$ , where  $w_{t+1,\eta}(u)$  is given by Eq. (6.2.3), with  $\Phi_0(\cdot) = 1$ .

We upper- and the lower-bound the telescoping product

$$Q := \ln \left( \frac{\Phi_T(\eta_T)}{\Phi_0(\eta_0)} \right) = \ln \left( \prod_{t=1}^T \frac{\Phi_t(\eta_t)}{\Phi_{t-1}(\eta_{t-1})} \right) = \sum_{t \in [T]} Q_t, \text{ where } Q_t = \ln \left( \frac{\Phi_t(\eta_t)}{\Phi_{t-1}(\eta_{t-1})} \right).$$

We lower-bound  $Q$  in terms of the “best node”  $u_T^*$ , accounting for the ancestors via  $\mathcal{C}_{\text{prod}}(\cdot)$ :

$$Q \geq \sum_{t \in [T]} \widehat{g}_t(\mathbf{act}_t(u_T^*)) - \ln(|A_T| \cdot \mathcal{C}_{\text{prod}}(u_T^*)) / \eta_T. \quad (6.4.7)$$

For the upper bound, we focus on the  $Q_t$  terms. We transition from potential  $\Phi_{t-1}(\eta_t)$  to  $\Phi_t(\eta_t)$  in two steps: first, the weights of all currently active nodes get updated, and then we zoom-in on the appropriate nodes. The former is handled using standard techniques, and the latter relies on the fact that the weights are preserved. We obtain:

$$Q \leq \sum_{t \in [T]} g_t(x_t) + \sum_{t \in [T]} O(\ln T) (\gamma_t + \beta_t \sum_{u \in A_t} \widehat{g}_t(u)). \quad (6.4.8)$$

**Part III: from Estimated to Realized Rewards.** We argue about realized rewards, with probability (sat) at least  $1 - 1/T$ . We bring in two more pieces of the overall puzzle: a Lipschitz property and a concentration bound for IPS estimators. If node  $u$  is active at time  $t$ , then

$$\sum_{\tau \in [t]} g_\tau(\text{OPT}_{[t]}(u)) - \sum_{\tau \in [t]} L(\mathbf{act}_\tau(u)) - 4\sqrt{td} \ln T \leq \sum_{\tau \in [t]} g_\tau(\mathbf{act}_\tau(u)). \quad (6.4.9)$$

(We only use Lipschitzness through (6.4.9).) For any subsets  $A'_\tau \subseteq A_\tau$ ,  $\tau \in [T]$  it holds that:

$$\left| \sum_{\tau \in [t], u \in A'_\tau} g_\tau(u) - \text{IPS}_\tau(u) \right| \leq O(\ln T) / \beta_t + \sum_{\tau \in [t], u \in A'_\tau} \beta_\tau / \pi_\tau(u). \quad (6.4.10)$$

The analysis of EXP3.P derives a special case of Eq. (6.4.10) with  $A'_\tau = \{u\}$  in all rounds  $\tau$ . The stronger version relies on *negative association* between random variables  $\widehat{g}_\tau(u)$ ,  $u \in A'_\tau$ .

Putting these two properties together, we relate estimates  $\widehat{g}_t(u)$  with the actual gains  $g_t(u)$ . First, we argue that we do not *over*-estimate by too much: fixing round  $t$ ,

$$\sum_{\tau \in [t], u \in A'_\tau} \beta_\tau (\widehat{g}_\tau(u) - g_\tau(u)) \leq O(\ln T) \left( 1 + \sum_{\tau \in [t]} \beta_\tau |A'_\tau| \right). \quad (6.4.11)$$

This holds for any subsets  $A'_\tau \subset A_\tau$ ,  $\tau \in [t]$  which only contain ancestors of the nodes in  $A'_t$ .

Second, we need a stronger version for a singleton node  $u$ , one with  $L(u)$  on the right-hand side.

If node  $u$  is zoomed-in in round  $t$ , then for each arm  $y \in u$  we have:

$$\sum_{\tau \in [t]} \hat{g}_\tau(\text{act}_\tau(u)) - g_\tau(y) \leq O \left( L(u) \cdot t \ln(T) + \sqrt{t d} \ln T + (\ln T)/\beta_t \right). \quad (6.4.12)$$

Third, we argue that the estimates  $\hat{g}_t(u)$  form an approximate upper bound. We only need this property for singleton nodes: for each node  $u$  which is active at round  $t$ , we have

$$\sum_{\tau \in [t]} \hat{g}_\tau(\text{act}_\tau(u)) - g_\tau(\text{OPT}_{[t]}(u)) \geq -O \left( \sqrt{t d} \ln T + (\ln T)/\beta_t \right). \quad (6.4.13)$$

To prove Eq. (6.4.13), we also use the zooming invariant Eq. (6.4.1) and the bound from Eq. (6.4.4) on inherited diameters.

Using these lemmas in conjunction with the upper/lower bounds of  $Q$  we can derive the “raw” regret bound (6.3.3), and subsequently the “tuned” version (6.3.4).

**Part IV: the Final Regret Bound.** We bound  $|A_T|$  to derive the final regret bound in Theorem 6.1. First, use the probability mass bound (6.4.5) to bound  $|A_T|$  in the worst case. We use an “adversarial activation” argument: given the rewards, what would an adversary do to activate as many nodes as possible, if it were only constrained by (6.4.5)? The adversary would go through the nodes in the order of decreasing diameter  $L(\cdot)$ , and activate them until the total probability mass exceeds  $T$ . The number of active nodes with diameter  $L(u) \in [\varepsilon, 2\varepsilon]$ , denoted  $N^{\text{act}}(\varepsilon)$ , is bounded via CovDim.

Second, we bound  $\text{AdvGap}_t(\cdot)$ . Plugging probabilities  $\pi_t$  into (6.4.6), bound the “estimated gap”,

$$\sum_{\tau \in [t]} \hat{g}_\tau(\text{act}_\tau(u_t^*)) - \sum_{\tau \in [t]} \hat{g}_\tau(\text{act}_\tau(u)) \leq \ln \left( \frac{g_{\mathcal{C}_{\text{prod}}}(u_t^*)}{\mathcal{C}_{\text{prod}}(u) \cdot \beta_t^2} \right) / \eta_t. \quad (6.4.14)$$

for a node  $u$  which is zoomed-in at round  $t$ . To translate this to the actual  $\text{AdvGap}_t(\cdot)$ , we bring in the machinery from Part III and the worst-case bound on  $|A_T|$  derived above.

$$\text{AdvGap}_t(\text{repr}(u)) \leq L(u) \cdot \mathcal{O} \left( \ln(T) \sqrt{d \ln(C_{\text{db1}} \cdot T)} \right). \quad (6.4.15)$$

We can now upper-bound  $N^{\text{act}}(\varepsilon)$  via  $\text{AdvZoomDim}$  rather than  $\text{CovDim}$ . With this, we run another

“adversarial activation” argument to upper-bound  $|A_T|$  in terms of AdvZoomDim.

## 6.5 REGRET ANALYSIS (DETAILS)

We analyze ADVERSARIALZOOMING and prove the main theorem (Theorem 6.1). We follow the proof sketch in Section 6.4, with subsections corresponding to the “parts” of the proof sketch. While Theorem 6.1 makes one blanket assumption on the parameters, we explicitly spell out which assumptions are needed for which lemma.

In what follows, we use  $d$  to denote the covering dimension, for some constant multiplier  $\gamma_0 > 0$ . The dependence on  $\gamma_0$  is only logarithmic, we suppress it for clarity.

**From randomized to deterministic rewards.** Define the *representative set* of arms as

$$\mathcal{A}_{\text{repr}} := \left\{ \text{repr}(u), x_{[t]}^*(u) : \text{tree nodes } u \text{ with } h(u) \leq 1 + \log T, \text{ rounds } t \in [T] \right\}. \quad (6.5.1)$$

Only tree nodes of height at most  $1 + \log(T)$  can be activated by the algorithm (as per Lemma 6.4), so  $\mathcal{A}_{\text{repr}}$  contains all arms that can possibly be pulled. Note that  $|\mathcal{A}_{\text{repr}}| \leq (T+1)^T$ .

A *canonical arm-sequence* is a sequence of arms  $y_1, \dots, y_t \in \mathcal{A}_{\text{repr}}$ , for some round  $t$ , which contains at most  $\log T$  switches. As it happens, we will only invoke Lemma C.3 on canonical arm-sequences. Since there are at most  $|\mathcal{A}_{\text{repr}}|^{\log T}$  such sequences, we can take a Union Bound over all of them. Formally, we define the *clean event* for rewards, denoted  $\mathcal{E}_{\text{clean}}^{\text{rew}}$ , which asserts that Equation (C.1.1) in Lemma C.3 holds with  $\delta = T^{-2-d \log T}$  for all rounds  $t$  and all canonical arm-sequences  $(y_1, \dots, y_t), (y'_1, \dots, y'_t)$ . Lemma C.3 implies that  $\Pr[\mathcal{E}_{\text{clean}}^{\text{rew}}] \geq 1 - 1/T$ .

From here on, we condition on  $\mathcal{E}_{\text{clean}}^{\text{rew}}$  without further mention. Put differently, we assume that rewards are deterministic and satisfy  $\mathcal{E}_{\text{clean}}^{\text{rew}}$ . Any remaining randomness is due to the algorithm’s random seed.

### 6.5.1 PROPERTIES OF THE ZOOM-IN RULE

This part of the analysis depends on the zoom-in rule, but not on the selection rule, *i.e.*, it works no matter how distribution  $\pi_t$  is chosen.

We start by defining the *zooming invariant* which holds for *all* active nodes. The zooming invariant is a property of the confidence that we have on the currently active nodes, and it is proved inductively using the fact that when a node does not get zoomed-in, then either the instantaneous or the aggregate rules are not satisfied (Equation (6.2.9) in Section 6.2).

**Lemma 6.3** (Zooming Invariant). *If node  $u$  is active at round  $t$ , then:  $\text{conf}_t^{\text{tot}}(u) \geq (t - 1) \cdot L(u)$ .*

*Proof.* Since  $u$  is active at round  $t$  then  $t \in [\tau_0(u), \tau_1(u)]$ . We first focus on rounds where  $z_\tau(\cdot) = 0$ . Since for all rounds  $\tau \in [\tau_0(u), \tau_1(u)]$  node  $u$  was *not* zoomed-in, it must have been because either the instantaneous or the aggregate rules (Equation (6.2.9)) were not true. Assume first that the aggregate rule was not satisfied for rounds  $\tau \in [\tau_0(u), t_1]$  such that  $t_1 \leq t$ . In other words, from Equation (6.2.9) we have that:

$$\sum_{\tau=1}^{t_1} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \frac{1}{\beta_{t_1}} \geq t_1 \cdot L(u) \quad (6.5.2)$$

Note that if  $t_1 = t$ , then the lemma follows directly. Let  $t_2 \geq t_1 + 1$  be the round such that for all rounds  $\tau \in [t_1 + 1, t_2]$  the aggregate rule *does* hold, but the instantaneous *does not*. Hence, for all such rounds it holds that:

$$\frac{\beta_\tau}{\pi_\tau(u)} + \tilde{\beta}_\tau \geq e^{L(u)} - 1 \geq L(u) \quad (6.5.3)$$

where the last inequality is due to the property that  $e^x - 1 \geq x$ . Summing up both sides of Equation (6.5.3) for all rounds  $\tau \in [t_1 + 1, t_2]$  we get that:

$$\sum_{\tau=t_1+1}^{t_2} \frac{\beta_\tau}{\pi_\tau(u)} + \sum_{\tau=t_1+1}^{t_2} \tilde{\beta}_\tau \geq \sum_{\tau=t_1+1}^{t_2} L(u) \quad (6.5.4)$$

Since  $\tilde{\beta}_t$  is positive:  $\sum_{\tau=t_1+1}^{t_2} \tilde{\beta}_t \leq \sum_{\tau=t_1}^{t_2} \tilde{\beta}_t$ , and by the assumption on  $\tilde{\beta}_t$  (Equation (6.2.10) in Section 6.2) we can relax the left hand side of Equation (6.5.4) and obtain:

$$\sum_{\tau=t_1+1}^{t_2} \frac{\beta_\tau}{\pi_\tau(u)} + \frac{1}{\beta_{t_2}} - \frac{1}{\beta_{t_1}} \geq (t_2 - t_1)L(u) \quad (6.5.5)$$

Note that for all rounds  $\tau \in [\tau_0(u), \tau_1(u)]$ :  $\text{act}_\tau(u) = u$ . As a result, summing up Equation (6.5.2)

and Equation (6.5.5) we have that:

$$\sum_{\tau=1}^{t_2} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \frac{1}{\beta_{t_2}} \geq t_2 \cdot L(u)$$

Applying the same arguments for all rounds  $\tau \in [1, t]$  we have that:

$$\sum_{\tau=1}^t \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \frac{1}{\beta_{\tau_1(u)-1}} \geq tL(u) \quad (6.5.6)$$

which completes our proof if  $t \leq \tau_1(u) - 1$ . In order to complete the proof we show what happens for the case that  $t = \tau_1(u)$ . Note that when node  $u$  gets zoomed-in at round  $\tau_1(u)$ , both the instantaneous and the aggregate rules hold (and  $z_\tau(u) = 1$ ). So we have that:

$$\begin{aligned} \sum_{\tau=1}^{\tau_1(u)} \frac{\beta_\tau}{\pi_\tau(u)} + \frac{1}{\beta_{\tau_1(u)}} &\geq \sum_{\tau=1}^{\tau_1(u)-1} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \frac{1}{\beta_{\tau_1(u)}} && (\beta_\tau, \pi_\tau(\cdot) > 0) \\ &\geq \sum_{\tau=1}^{\tau_1(u)-1} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \frac{1}{\beta_{\tau_1(u)-1}} && (\beta_\tau \leq \beta_{\tau-1}) \\ &\geq (\tau_1(u) - 1) L(u) && \text{(Equation (6.5.6))} \end{aligned}$$

This concludes our proof. ■

We define the inherited  $\tau$ -diameter as:  $b(u) = \tau_1(u) \cdot L(u)$ , i.e., inherited  $\tau$ -diameter is the upper bound on the “total bias” suffered when we are at node  $u$ . The next lemma shows that the zooming tree cannot grow arbitrarily large as time goes by, by bounding the de-activation time of a node  $u$  with the inverse of its diameter (Equation (6.4.2)); in fact, we show that the height of the zooming tree at any node  $u$ , denoted by  $h(u)$ , cannot be larger than  $\log T$ .

**Lemma 6.4** (Bound on Height of Node). *Assume that  $\tilde{\beta}_\tau \geq \beta_\tau, \forall \tau$  and  $\beta_\tau \geq 1/\tau$ . Then, if node  $u$  gets de-activated at round  $\tau_1(u)$ , it holds that  $b(u) \geq 1$  and as a result,  $h(u) \leq \log(\tau_1(u)) \leq \log T$ .*

*Proof.* From the instantaneous rule in Equation (6.2.9) for  $z_t(u)$  we have that if we zoom-in at a node  $u$  at round  $\tau_1(u)$  the following must be true:

$$\frac{\beta_{\tau_1(u)}}{\pi_t(u)} + \tilde{\beta}_{\tau_1(u)} \leq e^{L(u)} - 1 \Leftrightarrow \pi_{\tau_1(u)}(u) \geq \frac{\beta_{\tau_1(u)}}{e^{L(u)} - 1 - \tilde{\beta}_{\tau_1(u)}} \quad (6.5.7)$$

Because of the fact that  $\pi_\tau(\cdot)$  is a valid probability distribution:  $\pi_\tau(v) \leq 1, \forall v \in A_t$ . This imposes the following restriction on the right hand side of Equation (6.5.21):

$$\begin{aligned} L(u) &\geq \ln \left( \beta_{\tau_1(u)} + \tilde{\beta}_{\tau_1(u)} + 1 \right) \geq \ln \left( 2\beta_{\tau_1(u)} + 1 \right) \\ &\geq \frac{2\beta_{\tau_1(u)}}{2\beta_{\tau_1(u)} + 1} = \frac{1}{1 + \frac{1}{2\beta_{\tau_1(u)}}} \quad (\ln(1+x) \geq \frac{x}{x+1}, x \geq -1) \\ &\geq \frac{1}{\tau_1(u)} \end{aligned} \tag{6.5.8}$$

where the last inequality is due to the fact that  $\beta_\tau \geq 1/\tau, \forall \tau$  according to the assumptions of the lemma. Thus,  $b(u) \geq 1$ . Since every time that a node gets zoomed-in its diameter gets halved, we have that if node  $u$  is found at height  $h(u)$ , then:  $L(u) = L(u_0)2^{-h(u)} = 2^{-h(u)}$  because  $L(u_0) = 1$ . Substituting this to Eq. (6.5.8) and taking logarithms, we get that  $h(u) \leq \log(\tau_1(u)) \leq \log T$ . ■

Adding up to the point made earlier (i.e., that the action tree cannot grow arbitrarily large), in the next lemma we show that the lifespan of any node is *strongly* correlated with the lifespan of its ancestors. To be more precise, we show that the de-activation time of a node  $u$  is (approximately) at least *twice* larger than its activation (Equation (6.4.3)). An important implication of this is that once a node  $u$  gets zoomed in at a round  $t$  then it will not be possible to immediately zoom-in on any of its children, as we remarked on Section 6.2.

**Lemma 6.5** (Lifespan of Node Compared to Ancestors). *Let  $\{\beta_t\}_{t=1}^T$  be a non-increasing sequence. Then, if  $u$  gets de-activated at round  $\tau_1(u)$ , it holds that  $\tau_1(u) \geq 2\tau_1(\xi) - 2$ , where  $\xi$  is  $u$ 's parent.*

*Proof.* Since  $\xi$  is  $u$ 's parent, then the two nodes share the same ancestry tree. Hence,  $\forall \tau \in [1, \tau_1(\xi)] : \text{act}_\tau(\xi) = \text{act}_\tau(u)$  and  $\forall \tau \in [\tau_0(u), \tau_1(u)] : \text{act}_\tau(u) = u$ . For all rounds  $\tau \in [\tau_0(\xi), \tau_1(\xi)]$  the zooming invariant holds for node  $\xi$ . Hence, for  $t = \tau_1(\xi)$  and using the notation  $b(\xi) = \tau_1(\xi)L(\xi)$ :

$$\sum_{\tau=1}^{\tau_1(\xi)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(\xi))} + \frac{1}{\beta_{\tau_1(\xi)}} \geq b(\xi) - L(\xi)$$

Since the sequence  $\{\beta_t\}_{t=1}^T$  is (by assumption) non-increasing and  $\tau_1(u) \geq \tau_1(\xi)$  we can relax the

left hand side of the above inequality and get:

$$\sum_{\tau=1}^{\tau_1(\xi)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(\xi))} + \frac{1}{\beta_{\tau_1(u)}} \geq b(\xi) - L(\xi) \quad (6.5.9)$$

By definition, on round  $\tau_1(u)$  we decided to zoom-in, hence, the aggregate component of the zoom-in rule was true:

$$\begin{aligned} b(u) &\geq \sum_{\tau=1}^{\tau_1(u)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \frac{1}{\beta_{\tau_1(u)}} \\ &= \sum_{\tau=1}^{\tau_1(\xi)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \sum_{\tau=\tau_0(u)}^{\tau_1(u)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \frac{1}{\beta_{\tau_1(u)}} \\ &= \sum_{\tau=1}^{\tau_1(\xi)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(\xi))} + \sum_{\tau=\tau_0(u)}^{\tau_1(u)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \frac{1}{\beta_{\tau_1(u)}} \quad (\text{act}_\tau(\xi) = \text{act}_\tau(u), \forall \tau \leq \tau_1(\xi)) \\ &\geq b(\xi) - L(\xi) - \frac{1}{\beta_{\tau_1(\xi)}} + \sum_{\tau=\tau_0(u)}^{\tau_1(u)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \frac{1}{\beta_{\tau_1(u)}} \quad (\text{zooming invariant for node } \xi) \\ &\geq b(\xi) - L(\xi) + \sum_{\tau=\tau_0(u)}^{\tau_1(u)} \tilde{\beta}_\tau + \sum_{\tau=\tau_0(u)}^{\tau_1(u)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} \quad (\text{assumption on } \tilde{\beta}_t) \\ &\geq b(\xi) - L(\xi) + \sum_{\tau=\tau_0(u)}^{\tau_1(u)} \tilde{\beta}_\tau + \sum_{\tau=\tau_0(u)}^{\tau_1(u)} \beta_\tau \quad (\pi_t(\cdot) \leq 1) \\ &\geq b(\xi) - L(\xi) \end{aligned} \quad (6.5.10)$$

where the last inequality is due to the fact that  $\beta_\tau > 0, \forall \tau$ . To complete the proof, we use the definition of  $b(\cdot)$  along with the fact that  $L(\xi) = 2L(u)$  (since every time that a node gets de-activated its diameter gets halved) in Equation (6.5.10). ■

Next, we analyze the *total inherited diameter* of a node  $u$  in round  $t$ , defined as  $\sum_{\tau \in [t]} L(\text{act}_\tau(u))$ . We relate it to the “total bias”, as expressed by the node’s diameter, that the Lipschitz condition would have imposed on this node had it been active from round 1 (this is Equation (6.4.4) in Section 6.4).

**Lemma 6.6** (Inherited Bias Bound). *For a node  $u$  that is active at round  $t$ , the total inherited diameter that it has suffered is upper bounded by:  $\sum_{\tau \in [t]} L(\text{act}_\tau(u)) \leq 4t \log(T)L(u)$ .*

*Proof.* We first prove that for any node  $u$  it holds that

$$\tau_1(u) \geq 2^{h(u)} (\tau_1(u) - 2) \quad (6.5.11)$$

where  $u'$  is any node in the path from `root` to node  $u$ . Indeed, from Lemma 6.5 and denoting by  $v_i, i = \{1, \dots, h(u)\}, v_0 = \text{root}$  the path from `root` to  $u$  we have that:

$$\begin{aligned} \tau_1(u) &\geq 2(\tau_1(v_{h(u)-1}) - 1) && \text{(Lemma 6.5 for node } u\text{)} \\ &\geq 2(2(\tau_1(v_{h(u)-2}) - 1) - 1) && \text{(Lemma 6.5 for node } v_{h(u)-1}\text{)} \\ &\geq 2(2(2(\tau_1(v_{h(u)-3}) - 1) - 1) - 1) && \text{(Lemma 6.5 for node } v_{h(u)-2}\text{)} \\ &= 2^{h(u)-h(v_j)} \tau_1(v_j) - (1 + 2 + 4 + \dots) \\ &= 2^{h(u)-h(v_j)} \tau_1(v_j) - 2^{h(u)-h(v_j)+1} \\ &= 2^{h(u)-h(u')} (\tau_1(u') - 2) \end{aligned}$$

For the ease of notation we denote node  $v_{h(u)-1}$  as node  $\xi$ . Then, by the definition of total inherited diameter it holds that:

$$\begin{aligned} \sum_{\tau=1}^t L(\text{act}_\tau(u)) &= \tau_1(v_0)L(v_0) + (\tau_1(v_1) - \tau_0(v_1) + 1)L(v_1) + \dots + (t - \tau_0(u) + 1) \cdot L(u) \\ &= \tau_1(v_0)L(v_0) + (\tau_1(v_1) - \tau_1(v_0))L(v_1) + \dots + (t - \tau_1(\xi)) \cdot L(u) \\ &= [\tau_1(v_0)(L(v_0) - L(v_1)) + \tau_1(v_1)(L(v_1) - L(v_2)) + \dots] + tL(u) \\ &= \left[ \tau_1(v_0) \frac{L(v_0)}{2} + \tau_1(v_1) \frac{L(v_1)}{2} + \dots \right] + tL(u) \\ &= \frac{1}{2} [\tau_1(v_0)L(v_0) + \tau_1(v_1)L(v_1) + \dots] + tL(u) \\ &= \frac{1}{2} [\tau_1(v_0) \cdot 2^{h(\xi)} \cdot L(\xi) + \tau_1(v_1) \cdot 2^{h(\xi)-1} \cdot L(\xi) + \dots] + tL(u) \\ &\leq \frac{1}{2} [\tau_1(\xi) \cdot L(\xi) \cdot h(\xi) + L(\xi)(2 + 4 + 8 + \dots)] + tL(u) && \text{(Equation (6.5.11))} \\ &\leq tL(u)h(u) + 2L(u) \cdot 2^{h(\xi)+1} + tL(u) && (L(\xi) = 2L(u) \text{ and geometric series}) \\ &\leq 2tL(u)h(u) + 2L(u) \cdot 2^{h(u)} && (6.5.12) \end{aligned}$$

where the first equality is due to the fact that  $\tau_1(v_j) + 1 = \tau_0(v_{j+1})$  and the fourth and fifth equalities is due to the fact that every time that we zoom-in the diameter of the parent node gets halved. From Lemma (6.4) we have that  $h(u) \leq \log T$ , and hence Equation (6.5.12) becomes:  $\sum_{\tau=1}^t L(\text{act}_\tau(u)) \leq 4tL(u)\log T$ . ■

The next lemma shows that the probability mass that has been spent on a node from the round it gets activated until the round it gets de-activated is inversely proportional to the square of the diameter of the node (Equation (6.4.5)). This property will be very important for arguing about the adversarial zooming dimension.

**Lemma 6.7** (Probability Mass Spent on A Node). *For a node  $u$  that gets de-activated at round  $\tau_1(u)$ , the probability mass spent on it from its activation time until its de-activation,  $\mathcal{M}(u)$ , is:*

$$\mathcal{M}(u) = \sum_{\tau=\tau_0(u)}^{\tau_1(u)} \pi_\tau(u) \geq \frac{1}{9L^2(u)}$$

*Proof.* Let node  $\xi$  be node  $u$ 's parent. Since at round  $\tau = \tau_1(u)$  we zoom-in on node  $u$  then, from the aggregate zoom-in rule it holds that:

$$\sum_{\tau=1}^{\tau_1(\xi)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \sum_{\tau=\tau_0(u)}^{\tau_1(u)} \frac{\beta_\tau}{\pi_\tau(u)} + \frac{1}{\beta_{\tau_1(u)}} = \sum_{\tau=1}^{\tau_1(u)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \frac{1}{\beta_{\tau_1(u)}} \leq b(u) \quad (6.5.13)$$

where the first equality is due to the fact that for rounds  $\tau \in [\tau_0(u), \tau_1(u)]$ :  $\text{act}_\tau(u) = u$ . Since nodes  $u$  and  $\xi$  share the same ancestors, for all rounds  $\tau \leq \tau_1(\xi)$  it holds that:  $\text{act}_\tau(u) = \text{act}_\tau(\xi)$ , and Equation (6.5.13) can be rewritten as:

$$\sum_{\tau=1}^{\tau_1(\xi)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(\xi))} + \sum_{\tau=\tau_0(u)}^{\tau_1(u)} \frac{\beta_\tau}{\pi_\tau(u)} + \frac{1}{\beta_{\tau_1(u)}} \leq b(u) \quad (6.5.14)$$

From the zooming invariant (Lemma 6.3) for round  $\tau_1(\xi)$  we have that:

$$\sum_{\tau=1}^{\tau_1(\xi)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(\xi))} \geq b(\xi) - L(\xi) - \frac{1}{\beta_{\tau_1(\xi)}}$$

Substituting the latter to Equation (6.5.14) we get that:

$$\sum_{\tau=\tau_0(u)}^{\tau_1(u)} \frac{\beta_\tau}{\pi_\tau(u)} + \frac{1}{\beta_{\tau_1(u)}} - \frac{1}{\beta_{\tau_1(\xi)}} \leq b(u) - b(\xi) + L(\xi) \quad (6.5.15)$$

Since the sequence of  $\beta_\tau$ 's is non-increasing and  $\tau_1(\xi) \leq \tau_1(u)$ , then  $1/\beta_{\tau_1(u)} - 1/\beta_{\tau_1(\xi)} \geq 0$  and also  $\beta_\tau \geq \beta_{\tau_1(u)}$ ,  $\forall \tau \leq \tau_1(u)$ , so Equation (6.5.15) becomes:

$$\beta_{\tau_1(u)} \cdot \sum_{\tau=\tau_0(u)}^{\tau_1(u)} \frac{1}{\pi_\tau(u)} \leq b(u) - b(\xi) + L(\xi) \quad (6.5.16)$$

From the properties of the harmonic mean it holds that  $\frac{n}{\sum_{i=1}^t x_i^{-1}} \leq \frac{1}{n} \sum_{i=1}^n x_i$  and the above can be relaxed to:

$$\beta_{\tau_1(u)} \cdot (\tau_1(u) - \tau_1(\xi))^2 \cdot \frac{1}{\mathcal{M}(u)} \leq b(u) - b(\xi) + L(\xi) \quad (6.5.17)$$

At round  $\tau_1(\xi)$ , the aggregate rule holds for node  $\xi$ , so:

$$\sum_{\tau=1}^{\tau_1(\xi)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \frac{1}{\beta_{\tau_1(\xi)}} \leq b(\xi) \quad (6.5.18)$$

Since  $\sum_{\tau=1}^{\tau_1(\xi)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} \geq 0$  the left hand side of the above becomes:  $\frac{1}{\beta_{\tau_1(\xi)}} \leq b(\xi)$ . Since the sequence of  $\beta_\tau$ 's is non-increasing:  $\beta_\tau \leq \beta_1 \leq 1/2$  thus we have:  $b(\xi) \geq 2 \geq L(\xi), \forall \xi$ . This implies that the right hand side of Equation (6.5.17) can be relaxed and so:

$$\beta_{\tau_1(u)} \cdot (\tau_1(u) - \tau_1(\xi))^2 \cdot \frac{1}{\mathcal{M}(u)} \leq b(u) \quad (6.5.19)$$

From Lemma (6.5), we have that  $\tau_1(u) \geq 2(\tau_1(\xi) - 1)$  and it also holds that  $\tau_1(u)/2 - 1 \geq \tau_1(u)/3$ .

Combining this with Equation (6.5.19) we have that:

$$\frac{\beta_{\tau_1(u)} \cdot \tau_1(u)}{9 \cdot \mathcal{M}(u)} \leq L(u) \Leftrightarrow \mathcal{M}(u) \geq \frac{\beta_{\tau_1(u)} \cdot \tau_1(u)}{9L(u)} \quad (6.5.20)$$

In the next step, we will show that  $\tau_1(u) \cdot \beta_{\tau_1(u)} \geq 1/L(u)$ . At round  $\tau_1(u)$  node  $u$  gets zoomed-in,

so the aggregate zoom-in rule holds for node  $u$ :

$$\begin{aligned} b(u) &= \tau_1(u) \cdot L(u) \geq \sum_{\tau=1}^{\tau_1(u)} \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \frac{1}{\beta_{\tau_1(u)}} \\ &\geq \frac{1}{\beta_{\tau_1(u)}} \end{aligned} \quad (\beta_\tau, \pi_\tau(\cdot) > 0)$$

As a result,  $\tau_1(u) \cdot \beta_{\tau_1(u)} \geq 1/L(u)$  and Equation (6.5.20) becomes:

$$\mathcal{M}(u) \geq \frac{1}{9L^2(u)}$$

This concludes our proof. ■

Another important property of the zoom-in rule (and to be more precise, of its *instantaneous* component) is that when `ADVERSARIALZOOMING` zooms-in on a node  $u$ , then, the probability  $\pi_t(u)$  on this node is large (Equation (6.4.6)). Formally, this is stated below.

**Lemma 6.8** (Probability of Zoomed-In Node). *If node  $u$  gets zoomed-in at round  $t$ , then:*

$$\frac{\beta_t}{\pi_t(u)} + \tilde{\beta}_t \leq e^{L(u)} - 1 \Leftrightarrow \pi_t(u) \geq \frac{\beta_t}{e^{L(u)}} \quad (6.5.21)$$

Note here that relaxing the right hand side of Equation (6.5.21) using the two facts that  $\beta_t \geq \beta_t^2$  and  $1 \geq \pi_t(u_t^*)$  gives the stated form of Equation (6.4.6).

*Proof.* From the instantaneous zoom-in rule for node  $u$  we have that if we zoom-in at a node  $u$  at round  $t$  the following must be true:

$$\frac{\beta_t}{\pi_t(u)} + \tilde{\beta}_t \leq e^{L(u)} - 1 \Leftrightarrow \pi_t(u) \geq \frac{\beta_t}{e^{L(u)} - 1 - \tilde{\beta}_t} \geq \frac{\beta_t}{e^{L(u)}} \quad \blacksquare$$

## 6.5.2 PROPERTIES OF THE SELECTION RULE

This part of the analysis depends on the selection rule in the algorithm, but not on the zoom-in rule. Specifically, it works regardless of how  $z_t(u)$  is defined in Line 10 of the algorithm. We consider

the multiplicative weights update, as defined in (6.2.2) and (6.2.3), and derive a lemma which corresponds to Equation (6.4.7) and Equation (6.4.8) in Section 6.4. The proof encompasses the standard multiplicative-weights arguments, with several key modifications due to zooming.

**Lemma 6.9.** *Assume the sequences  $\{\eta_t\}$  and  $\{\beta_t\}$  are decreasing in  $t$ , and satisfy*

$$\eta_t \leq \beta_t \leq \gamma_t / |A_t| \quad \text{and} \quad \eta_t (1 + \beta_t(1 + 4 \log T)) \leq \gamma_t / |A_t|. \quad (6.5.22)$$

*Then, the following inequality holds:*

$$\begin{aligned} \sum_{t \in [T]} \widehat{g}_t(\mathbf{act}_t(u_T^*)) - \sum_{t \in [T]} g_t(x_t) &\leq \frac{\ln(|A_T| \cdot \mathcal{C}_{\text{prod}}(u_T^*))}{\eta_T} + 4(1 + \log T) \sum_{t \in [T]} \gamma_t + \\ &+ 2 \sum_{t \in [T]} \eta_t (1 + (1 + 4 \log T) \beta_t) \sum_{u \in A_t} \widehat{g}_t(u) \end{aligned}$$

*Proof.* We use the following potential function:

$$\Phi_t(\eta) = \left( \frac{1}{|A_t|} \sum_{u \in A_t} \frac{1}{\mathcal{C}_{\text{prod}}(u)} \cdot \exp \left( \eta \sum_{\tau=1}^t \widehat{g}_\tau(\mathbf{act}_\tau(u)) \right) \right)^{1/\eta}, \quad \Phi_0(\eta) = 1, \forall \eta \quad (6.5.23)$$

where  $\Phi_0(\cdot) = 1$  since at round 0 there is only one active node (the root with  $h(\text{root}) = 0$ ) and the estimator of the cumulative reward is initialized to 0. We next upper and lower bound the quantity

$$Q = \ln \left( \frac{\Phi_T(\eta_T)}{\Phi_0(\eta_0)} \right) = \ln \left( \prod_{t=1}^T \frac{\Phi_t(\eta_t)}{\Phi_{t-1}(\eta_{t-1})} \right) = \sum_{t=1}^T \underbrace{\ln \left( \frac{\Phi_t(\eta_t)}{\Phi_{t-1}(\eta_{t-1})} \right)}_{Q_t} \quad (6.5.24)$$

Expanding  $Q$  we get:

$$\begin{aligned}
\ln \left( \frac{\Phi_T(\eta_T)}{\Phi_0(\eta_0)} \right) &= \frac{1}{\eta_T} \ln \left( \frac{1}{|A_T|} \sum_{u \in A_T} \frac{1}{C_{\text{prod}}(u)} \cdot \exp \left( \eta_T \sum_{t=1}^T \hat{g}_t(\text{act}_t(u)) \right) \right) - \ln (\Phi_0(\eta_0)) \\
&= \frac{1}{\eta_T} \ln \left( \frac{1}{|A_T|} \sum_{u \in A_T} \frac{1}{C_{\text{prod}}(u)} \cdot \exp \left( \eta_T \sum_{t=1}^T \hat{g}_t(\text{act}_t(u)) \right) \right) \quad (\Phi_0(\cdot) = 1) \\
&\geq \frac{1}{\eta_T} \ln \left( \frac{1}{|A_T|} \cdot \frac{1}{C_{\text{prod}}(u)} \cdot \exp \left( \eta_T \sum_{t=1}^T \hat{g}_t(\text{act}_t(u_T^*)) \right) \right) \quad (e^x > 0) \\
&= \frac{1}{\eta_T} \ln \left( \exp \left( \eta_T \sum_{t=1}^T \hat{g}_t(\text{act}_t(u_T^*)) \right) \right) - \frac{\ln(|A_T| \cdot C_{\text{prod}}(u_T^*))}{\eta_T} \\
&= \sum_{t=1}^T \hat{g}_t(\text{act}_t(u_T^*)) - \frac{\ln(|A_T| \cdot C_{\text{prod}}(u_T^*))}{\eta_T} \tag{6.5.25}
\end{aligned}$$

For the upper bound we first focus on quantity  $Q_t$  from Equation (6.5.23), and we start by breaking  $Q_t$  into the following parts:

$$\begin{aligned}
Q_t &= \ln \left( \frac{\Phi_t(\eta_t)}{\Phi_{t-1}(\eta_{t-1})} \right) = \ln \left( \frac{\Phi_t(\eta_t)}{\Phi_{t-1}(\eta_t)} \cdot \frac{\Phi_{t-1}(\eta_t)}{\Phi_{t-1}(\eta_{t-1})} \right) \\
&= \underbrace{\ln \left( \frac{\Phi_t(\eta_t)}{\Phi_{t-1}(\eta_t)} \right)}_{\tilde{Q}_t} + \underbrace{\ln \left( \frac{\Phi_{t-1}(\eta_t)}{\Phi_{t-1}(\eta_{t-1})} \right)}_{\tilde{Q}_t} \tag{6.5.26}
\end{aligned}$$

We define the auxiliary function  $f(\eta) = \ln(\Phi_{t-1}(\eta))$  and prove that  $f'(\eta) \geq 0$ , hence the function is *increasing* in  $\eta$ . Since  $\eta_{t-1} \geq \eta_t$ , this implies that quantity  $\tilde{Q}_t$  is *negative* for all  $t \in [T]$ . For the ease of notation of this part we denote by  $\hat{G}_t(u)$  the quantity  $\sum_{\tau=1}^t g_\tau(\text{act}_\tau(u))$ . For the derivative

of function  $f(\eta)$  we have:

$$\begin{aligned}
f'(\eta) &= -\frac{1}{\eta^2} \ln \left( \frac{1}{|A_{t-1}|} \sum_{u \in A_{t-1}} \frac{1}{C_{\text{prod}}(u)} \exp \left( \eta \hat{G}_t(u) \right) \right) \\
&\quad + \frac{1}{\eta} \frac{\sum_{u \in A_{t-1}} \frac{1}{C_{\text{prod}}(u)} \hat{G}_t(u) \exp \left( \eta \hat{G}_t(u) \right)}{\sum_{u \in A_{t-1}} \frac{1}{C_{\text{prod}}(u)} \exp \left( \eta \hat{G}_t(u) \right)} \\
&= \frac{1}{\eta^2} \frac{1}{\sum_{u \in A_{t-1}} \frac{1}{C_{\text{prod}}(u)} \exp \left( \eta \hat{G}_t(u) \right)} \sum_{u \in A_{t-1}} \frac{1}{C_{\text{prod}}(u)} \exp \left( \eta \hat{G}_t(u) \right) \cdot \\
&\quad \cdot \left[ \eta \hat{G}_t(u) - \ln \left( \frac{1}{|A_{t-1}|} \sum_{u \in A_{t-1}} \frac{1}{C_{\text{prod}}(u)} \exp \left( \eta \hat{G}_t(u) \right) \right) \right] \\
&\geq \frac{1}{\eta^2} \frac{1}{\sum_{u \in A_{t-1}} \frac{1}{C_{\text{prod}}(u)} \exp \left( \eta \hat{G}_t(u) \right)} \sum_{u \in A_{t-1}} \frac{1}{C_{\text{prod}}(u)} \exp \left( \eta \hat{G}_t(u) \right) \cdot \\
&\quad \cdot \left[ \eta \hat{G}_t(u) - \ln \left( \frac{C_{\text{prod}}(u)}{|A_{t-1}|} \sum_{u \in A_{t-1}} \frac{1}{C_{\text{prod}}(u)} \exp \left( \eta \hat{G}_t(u) \right) \right) \right] \tag{6.5.27}
\end{aligned}$$

where the last inequality is due to the fact that  $C_{\text{prod}}(u) \geq 1, \forall u$ . We define now the following two probability distributions:

$$\begin{aligned}
D_1(u) &= \frac{\frac{1}{C_{\text{prod}}(u)} \exp \left( \eta \hat{G}_t(u) \right)}{\sum_{u' \in A_{t-1}} \frac{1}{C_{\text{prod}}(u')} \exp \left( \eta \hat{G}_t(u') \right)} \\
D_2(u) &= \frac{1}{|A_{t-1}|}
\end{aligned}$$

Then, the right hand side of Equation (6.5.27) is the KL-divergence from  $D_2$  to  $D_1$ . Because the KL-divergence is a non-negative quantity,  $f'(\eta) \geq 0$ , as desired.

We turn our attention to term  $\hat{Q}_t$  now and we break the process of transitioning from potential  $\Phi_{t-1}(\eta_t)$  to  $\Phi_t(\eta_t)$  into two steps. In the first step, the potential gets updated to be  $\Phi_t^I(\eta_t)$ <sup>\*\*</sup>, where the weights of all active nodes  $u \in A_{t-1}$  get updated according to the  $\hat{g}_t(u)$  estimator. In the second step, the zoom-in happens and the potential transitions from  $\Phi_t^I(\eta_t)$  to  $\Phi_t(\eta_t)$ . Note that since  $|A_{t-1}| \leq |A_t|$  and for all children  $v$  of  $u$ :  $\sum_{v \in C(u)} w_{t+1}(v) = w_{t+1}(u)$  (and similarly for the probability of the children nodes) we have that, irrespective of whether we zoom-in on node  $u$  or not,

---

<sup>\*\*</sup>I stands for “intermediate”

$\Phi_{t+1}^I(\eta_t) = \Phi_{t+1}(\eta_t)$ . Hence,

$$\begin{aligned}
\widehat{Q}_t &= \ln \left( \frac{\Phi_t(\eta_t)}{\Phi_t^I(\eta_t)} \cdot \frac{\Phi_t^I(\eta_t)}{\Phi_{t-1}(\eta_t)} \right) = \ln \left( \frac{\Phi_t(\eta_t)}{\Phi_t^I(\eta_t)} \right) + \ln \left( \frac{\Phi_t^I(\eta_t)}{\Phi_{t-1}(\eta_t)} \right) \\
&= \ln \left( \frac{\Phi_t^I(\eta_t)}{\Phi_{t-1}(\eta_t)} \right) \quad (\Phi_t(\eta_t) = \Phi_t^I(\eta_t)) \\
&= \frac{1}{\eta_t} \cdot \ln \left( \frac{\frac{1}{|A_t|} \cdot \sum_{u \in A_t} w_t(\text{act}_t(u), \eta_t) \cdot \exp(\eta_t \widehat{g}_t(\text{act}_t(u)))}{\frac{1}{|A_t|} W_t(\eta_t)} \right) \quad (\text{Lemma 6.1}) \\
&= \frac{1}{\eta_t} \ln \left( \sum_{u \in A_t} p_t(\text{act}_t(u)) \cdot \exp(\eta_t \widehat{g}_t(\text{act}_t(u))) \right) = \frac{1}{\eta_t} \ln \left( \sum_{u \in A_t} p_t(u) \cdot \exp(\eta_t \widehat{g}_t(u)) \right) \quad (6.5.28)
\end{aligned}$$

where the last equality comes from the fact that by definition  $\text{act}_t(u) = u$  for all the rounds  $t$  that  $u$  was active.

Next we show that choosing  $\beta_t, \gamma_t$  and  $\eta_t$  according to the assumptions of the lemma, we have that  $\eta_t \widehat{g}_t(\text{act}_t(u)) \leq 1$ . Indeed,

$$\begin{aligned}
\eta_t \widehat{g}_t(u) &= \eta_t \cdot \frac{g_t(x_t) \mathbb{1}\{u = U_t\} + (1 + 4 \log T) \beta_t}{\pi_t(u)} \quad (\text{by definition of } \widehat{g}_t(\cdot)) \\
&\leq \eta_t \cdot \frac{1 + (1 + 4 \log T) \beta_t}{\pi_t(u)} \quad (g_t(\cdot) \in [0, 1]) \\
&\leq \frac{\eta_t (1 + (1 + 4 \log T) \beta_t) \cdot |A_t|}{\gamma_t} \quad (\pi_t(u) \geq \gamma_t / |A_t|) \\
&\leq 1 \quad (\text{assumptions of Lemma})
\end{aligned}$$

Since  $\eta_t \widehat{g}_t(u) \leq 1$ , then we can use inequality  $e^x \leq 1 + x + x^2$  for  $x \leq 1$  in Equation (6.5.28) and we have that:

$$\begin{aligned}
\widehat{Q}_t &\leq \frac{1}{\eta_t} \left( \ln \left( \sum_{u \in A_t} p_t(u) (1 + \eta_t \widehat{g}_t(u) + \eta_t^2 \widehat{g}_t^2(u)) \right) \right) \\
&= \frac{1}{\eta_t} \left( \ln \left( 1 + \eta_t \sum_{u \in A_t} p_t(u) \widehat{g}_t(u) + \eta_t^2 \sum_{u \in A_t} p_t(u) \widehat{g}_t^2(u) \right) \right) \quad (\sum_{u \in A_t} p_t(u) = 1) \\
&\leq \sum_{u \in A_t} p_t(u) \widehat{g}_t(u) + \eta_t \sum_{u \in A_t} p_t(u) \widehat{g}_t^2(u) \quad (\ln(1 + x) \leq x, x \geq 0) \\
&\leq \frac{1}{1 - \gamma_t} \sum_{u \in A_t} \pi_t(u) \widehat{g}_t(u) + \frac{\eta_t}{1 - \gamma_t} \sum_{u \in A_t} \pi_t(u) \widehat{g}_t^2(u) \quad (6.5.29)
\end{aligned}$$

where the last inequality uses the fact that since for any node  $u$ :  $\pi_t(u) = (1 - \gamma_t)p_t(u) + \gamma_t/|A_t|$  then  $p_t(u) \leq \pi_t(u)/(1 - \gamma_t)$ . We next analyze term  $\sum_{u \in A_t} \pi_t(u)\hat{g}_t(u)$ :

$$\begin{aligned}\sum_{u \in A_t} \pi_t(u)\hat{g}_t(u) &= \sum_{u \in A_t} \pi_t(u) \left( \frac{g(x_t)\mathbb{1}\{u = U_t\}}{\pi_t(u)} + \frac{(1 + 4 \log T)\beta_t}{\pi_t(u)} \right) \\ &= \sum_{u \in A_t} \pi_t(u) \left( \frac{g_t(x_t)\mathbb{1}\{u = I_t\}}{\pi_t(u)} + \frac{(1 + 4 \log T)\beta_t}{\pi_t(u)} \right) \\ &= g_t(x_t) + (1 + 4 \log T)\beta_t|A_t|\end{aligned}\tag{6.5.30}$$

Next, we analyze term  $\sum_{u \in A_t} \pi_t(u)\hat{g}_t^2(u)$ :

$$\begin{aligned}\sum_{u \in A_t} \pi_t(u)\hat{g}_t^2(u) &= \sum_{u \in A_t} (\pi_t(u)\hat{g}_t(u)) \cdot \hat{g}_t(u) \\ &= \sum_{u \in A_t} \left( \pi_t(u) \frac{g_t(x_t)\mathbb{1}\{u = U_t\} + (1 + 4 \log T)\beta_t}{\pi_t(u)} \right) \cdot \hat{g}_t(u) \\ &\leq \sum_{u \in A_t} (1 + (1 + 4 \log T)\beta_t) \cdot \hat{g}_t(u)\end{aligned}\tag{6.5.31}$$

where the last inequality is due to the fact that  $g_t(\cdot) \in [0, 1]$ . Using Equation (6.5.30) and Equation (6.5.31) in Equation (6.5.29), we get that:

$$\begin{aligned}\hat{Q}_t &\leq \frac{1}{1 - \gamma_t} \cdot \left[ g_t(x_t) + (1 + 4 \log T)\beta_t|A_t| + \eta_t(1 + (1 + 4 \log T)\beta_t) \sum_{u \in A_t} \hat{g}_t(u) \right] \\ &\leq (1 + 2\gamma_t) \left[ g_t(x_t) + (1 + 4 \log T)\beta_t|A_t| + \eta_t(1 + (1 + 4 \log T)\beta_t) \sum_{u \in A_t} \hat{g}_t(u) \right]\end{aligned}\tag{6.5.32}$$

where the last inequality comes from the assumption that  $\gamma_t \leq 1/2$ . Summing up both sides of the above for rounds  $t = 1, \dots, T$  we have that:

$$\sum_{t=1}^T \hat{Q}_t \leq \sum_{t=1}^T g_t(x_t) + 4(1 + \log T) \sum_{t=1}^T \gamma_t + 2 \sum_{t=1}^T \eta_t(1 + (1 + 4 \log T)\beta_t) \sum_{u \in A_t} \hat{g}_t(u)$$

From the assumption that  $\eta_t \leq \beta_t$  the latter becomes:

$$\sum_{t=1}^T \hat{Q}_t \leq \sum_{t=1}^T g_t(x_t) + 2 \sum_{t=1}^T \gamma_t + 2(1 + 4 \log T) \sum_{t=1}^T \gamma_t + 4(1 + 2 \log T) \sum_{t=1}^T \beta_t \sum_{u \in A_t} \hat{g}_t(u)\tag{6.5.33}$$

Combining and re-arranging Equations (6.5.25) and (6.5.33) concludes the proof. ■

### 6.5.3 FROM ESTIMATED TO REALIZED REWARDS.

In this part of the analysis, we go from properties of the estimated rewards to those of realized rewards. Recall that we posit deterministic rewards, conditioning on the clean event  $\mathcal{E}_{\text{clean}}^{\text{rew}}$ . Essentially, per-realization Lipschitzness (C.1.1) holds for all rounds  $t$  and all canonical arm-sequences. We prove several high-probability statements about estimated rewards; they hold with probability at least  $1 - O(\delta)$  over the algorithm's random seed, for a given  $\delta > 0$ .

The confidence bounds are somewhat more general than those presented in the proof sketch in Section 6.4: essentially, the sums over rounds  $\tau$  are weighted by time-dependent multipliers  $\hat{\beta}_\tau \leq \beta_\tau$ . This generality does not require substantive new ideas, but it is essential for the intended applications.

We revisit per-realization Lipschitzness and derive a corollary for inherited rewards. This is the statement and the proof of Equation (6.4.9).

**Lemma 6.10** (One-Sided-Lipschitz). *For any node  $u$  that is active at round  $t$  it holds that:*

$$\sum_{\tau \in [t]} g_\tau (\text{OPT}_{[t]}(u)) - \sum_{\tau \in [t]} L(\text{act}_\tau(u)) - 4\sqrt{td} \ln T \leq \sum_{\tau \in [t]} g_\tau(\text{act}_\tau(u)) \quad (6.5.34)$$

*Proof.* We use (C.1.1) with two sequences of arms:  $y_\tau = \text{repr}(\text{act}_\tau(u))$ ,  $\tau \in [t]$  and  $y'_\tau \equiv \text{OPT}_{[t]}(u)$ . By Lemma 6.4, node  $u$  has height at most  $1 + \log T$ , so  $(y_1, \dots, y_t)$  is a canonical arm-sequence. The other sequence,  $(y'_1, \dots, y'_t)$  is a canonical arm-sequence since  $\text{OPT}_{[t]}(u) \in \mathcal{A}_{\text{repr}}$  by (6.5.1). So, the Lemma follows by definition of the clean event  $\mathcal{E}_{\text{clean}}^{\text{rew}}$ . ■

We prove a series of lemmas, which heavily rely on the properties of the zoom-in rule derived in Section 6.5.1. The next lemma formally states the high probability confidence bound (Equation (6.4.10)).

**Lemma 6.11** (High Probability Confidence Bounds). *For any round  $t \in [T]$ , any  $\hat{\beta}_\tau \in (0, 1]$  such that*

$\widehat{\beta}_\tau \leq \beta_\tau$ ,  $\delta > 0$  and any subset  $A'_\tau \subseteq A_\tau$ , with probability at least  $1 - T^{-2}$  it holds that:

$$\left| \sum_{\tau \in [t]} \widehat{\beta}_\tau \sum_{u \in A'_\tau} g_\tau(u) - \sum_{\tau \in [t]} \widehat{\beta}_\tau \sum_{u \in A'_\tau} \text{IPS}_\tau(u) \right| \leq \sum_{\tau \in [t]} \widehat{\beta}_\tau \sum_{u \in A'_\tau} \frac{\beta_\tau}{\pi_\tau(u)} + 2 \ln T$$

*Proof.* To simplify notation in the next proof, we define two quantities which when aggregated over time will correspond to the upper and lower confidence bounds of the true cumulative reward, for each node  $u \in A_t$ :

$$\tilde{g}_t^+(u) = \text{IPS}_t(u) + \frac{\beta_t}{\pi_t(u)}, \text{ and } \sum_{s=1}^t \tilde{g}_s^+(u) = \sum_{s=1}^t \left( \text{IPS}_s(\text{act}_s(u)) + \frac{\beta_s}{\pi_s(\text{act}_s(u))} \right) \quad (6.5.35)$$

$$\tilde{g}_t^-(u) = \text{IPS}_t(u) - \frac{\beta_t}{\pi_t(u)}, \text{ and } \sum_{s=1}^t \tilde{g}_s^-(u) = \sum_{s=1}^t \left( \text{IPS}_s(\text{act}_s(u)) - \frac{\beta_s}{\pi_s(\text{act}_s(u))} \right) \quad (6.5.36)$$

It is easy to see that for all  $u$  we can express  $\widehat{g}_t(u)$  in terms of  $\tilde{g}_t^+(u)$  and  $\tilde{g}_t^-(u)$  as follows:

$$\widehat{g}_t(u) = \tilde{g}_t^+(u) + \frac{4 \log(T) \beta_t}{\pi_t(u)}, \text{ and } \sum_{s=1}^t \widehat{g}_s(u) = \sum_{s=1}^t \left( \tilde{g}_s^+(\text{act}_s(u)) + \frac{4 \log(T) \beta_s}{\pi_s(\text{act}_s(u))} \right) \quad (6.5.37)$$

$$\widehat{g}_t(u) = \tilde{g}_t^-(u) + \frac{(3 + 4 \log(T)) \beta_t}{\pi_t(u)}, \text{ and } \sum_{s=1}^t \widehat{g}_s(u) = \sum_{s=1}^t \left( \tilde{g}_s^-(\text{act}_s(u)) + \frac{(3 + 4 \log(T)) \beta_s}{\pi_s(\text{act}_s(u))} \right) \quad (6.5.38)$$

We prove the lemma in two steps. First, we show that for any  $\delta > 0$ :

$$\sum_{\tau \in [t]} \widehat{\beta}_\tau \sum_{u \in A'_\tau} g_\tau(u) \leq \sum_{\tau \in [t]} \widehat{\beta}_\tau \sum_{u \in A'_\tau} \widehat{g}_\tau^+(u) + \ln(1/\delta) \quad (6.5.39)$$

We denote by  $\mathbb{E}_\tau$  the expectation conditioned on the draws of nodes until round  $\tau$ , i.e., conditioned on  $U_1, \dots, U_{\tau-1}$ . Assume node  $v$  was active on round  $\tau$ . We will upper bound the quantity

$$\begin{aligned}
& \mathbb{E}_\tau \left[ \exp \left( \widehat{\beta}_\tau g_\tau(v) - \widehat{\beta}_\tau \widetilde{g}_\tau^+(v) \right) \right] : \\
& \mathbb{E}_\tau \left[ \exp \left( \widehat{\beta}_\tau g_\tau(v) - \widehat{\beta}_\tau \widetilde{g}_\tau^+(v) \right) \right] = \mathbb{E}_\tau \left[ \exp \left( \widehat{\beta}_\tau g_\tau(v) - \widehat{\beta}_\tau \cdot \frac{g_\tau(x_\tau) \mathbf{1}\{v = U_\tau\} + \beta_\tau}{\pi_\tau(v)} \right) \right] \\
& \leq \mathbb{E}_\tau \left[ 1 + \widehat{\beta}_\tau g_\tau(v) - \widehat{\beta}_\tau \frac{g_\tau(x_\tau) \mathbf{1}\{v = U_\tau\}}{\pi_\tau(v)} + \left( \widehat{\beta}_\tau g_\tau(v) - \widehat{\beta}_\tau \frac{g_\tau(x_\tau) \mathbf{1}\{v = U_\tau\}}{\pi_\tau(v)} \right)^2 \right] \\
& \quad \cdot \exp \left( - \frac{\widehat{\beta}_\tau \cdot \beta_\tau}{\pi_\tau(v)} \right) \\
& = \left( 1 + \mathbb{E}_\tau \left[ \widehat{\beta}_\tau g_\tau(v) - \widehat{\beta}_\tau \frac{g_\tau(x_\tau) \mathbf{1}\{v = U_\tau\}}{\pi_\tau(v)} \right] + \mathbb{E}_\tau \left[ \left( \widehat{\beta}_\tau g_\tau(v) - \widehat{\beta}_\tau \frac{g_\tau(x_\tau) \mathbf{1}\{v = U_\tau\}}{\pi_\tau(v)} \right)^2 \right] \right) \\
& \quad \cdot \exp \left( - \frac{\widehat{\beta}_\tau \cdot \beta_\tau}{\pi_\tau(v)} \right) \quad (\text{linearity of expectation}) \\
& \leq \left( 1 + \widehat{\beta}_\tau^2 \cdot \frac{g_\tau^2(x_\tau)}{\pi_\tau(v)} \right) \cdot \exp \left( - \frac{\widehat{\beta}_\tau \cdot \beta_\tau}{\pi_\tau(v)} \right) \leq 1 \tag{6.5.40}
\end{aligned}$$

where the first inequality is due to the fact that  $e^x \leq 1 + x + x^2$ ,  $x \leq 1$  and the last inequality is due to  $1 + x \leq e^x$  and  $\widehat{\beta}_\tau \leq \beta_\tau$ .

Let  $X_u$  be a *binary* random variable associated with node  $u$  which takes the value 1 if node  $u$  was chosen at this round by the algorithm (i.e.,  $u = U_\tau$ ) and 0 otherwise. Clearly,  $\{X_u\}_{u \in A_\tau}$  are *not* independent. However, since  $\sum_{u \in A_\tau} X_u = 1$  (i.e., at every round we play only one arm) then from [Dubhashi and Ranjan \(1996, Lemma 8\)](#), they are *negatively associated*. As a result, from [Joag-Dev and Proschan \(1983\)](#) for any non-increasing functions  $f_u(\cdot)$ ,  $\forall u \in A_\tau$  it holds that:

$$\mathbb{E} \left[ \prod_{u \in A_\tau} f_u(X_u) \right] \leq \prod_{u \in A_\tau} \mathbb{E}[f_u(X_u)]$$

In our case, the functions  $f_u$ ,  $\forall u \in A'_\tau \subseteq A_\tau$  are defined as:

$$f_u(X_u) = \exp \left( \widehat{\beta}_\tau g_\tau(u) - \widehat{\beta}_\tau \cdot \frac{g_\tau(x_\tau) \cdot X_u + \beta_\tau}{\pi_\tau(u)} \right), \quad \text{and} \quad \forall u \notin A'_\tau : f_u(X_u) = 1$$

which are non-increasing for each node. Multiplying both sides of Equation (6.5.40) for all nodes

$v \in A'_\tau$  and using the above stated properties we obtain:

$$\prod_{u \in A'_\tau} \mathbb{E}_\tau \left[ \exp \left( \widehat{\beta}_\tau g_\tau(u) - \widehat{\beta}_\tau \tilde{g}_\tau^+(u) \right) \right] \leq 1 \Leftrightarrow \mathbb{E}_\tau \left[ \prod_{u \in A'_\tau} \exp \left( \widehat{\beta}_\tau g_\tau(u) - \widehat{\beta}_\tau \tilde{g}_\tau^+(u) \right) \right] \leq 1$$

Since both sides of Equation (6.5.40) are positive and at each round we take the expectation conditional on the previous rounds, we have that:

$$\mathbb{E} \left[ \exp \left( \sum_{\tau=1}^t \widehat{\beta}_\tau \sum_{u \in A'_\tau} g_\tau(u) - \sum_{\tau=1}^t \widehat{\beta}_\tau \sum_{u \in A'_\tau} \tilde{g}_\tau^+(u) \right) \right] \leq 1 \quad (6.5.41)$$

From Markov's inequality, we have that  $\Pr [X > \ln(1/\delta)] \leq \delta \mathbb{E}[e^X]$  for any  $\delta > 0$ . Hence, from Equation (6.5.41) we have that with probability at least  $1 - \delta$ :

$$\sum_{\tau=1}^t \widehat{\beta}_\tau \sum_{u \in A'_\tau} g_\tau(u) - \sum_{\tau=1}^t \widehat{\beta}_\tau \sum_{u \in A'_\tau} \tilde{g}_\tau^+(u) \leq \ln(1/\delta)$$

Next, we show that:

$$\sum_{\tau \in [t]} \widehat{\beta}_\tau \sum_{u \in A'_\tau} g_\tau(u) \geq \sum_{\tau \in [t]} \widehat{\beta}_\tau \sum_{u \in A'_\tau} \tilde{g}_\tau^- - \ln(1/\delta) \quad (6.5.42)$$

Using exactly the same techniques we can show that:

$$\mathbb{E} \left[ \exp \left( \sum_{\tau=1}^t \widehat{\beta}_\tau \sum_{u \in A'_\tau} \tilde{g}_\tau^-(u) - \sum_{\tau=1}^t \widehat{\beta}_\tau \sum_{u \in A'_\tau} g_\tau(u) \right) \right] \leq 1$$

and ultimately:

$$\sum_{\tau=1}^t \widehat{\beta}_\tau \sum_{u \in A'_\tau} \tilde{g}_\tau^-(u) - \sum_{\tau=1}^t \widehat{\beta}_\tau \sum_{u \in A'_\tau} g_\tau(u) \leq \ln(1/\delta).$$

Substituting  $\delta = T^{-2}$  we get the result. ■

**Corollary 6.3.** Fix round  $t$  and tree node  $u$ . With probability at least  $1 - T^{-2}$ , we have:

$$\sum_{\tau=1}^t g_\tau(\text{act}_\tau(u)) \leq \sum_{\tau=1}^t \tilde{g}_\tau^+(\text{act}_\tau(u)) + \frac{2 \ln T}{\beta_t} \quad (6.5.43)$$

$$\sum_{\tau=1}^t g_\tau(\text{act}_\tau(u)) \geq \sum_{\tau=1}^t \tilde{g}_\tau^-(\text{act}_\tau(u)) - \frac{2 \ln T}{\beta_t} \quad (6.5.44)$$

*Proof.* Apply Lemma 6.11 for  $\hat{\beta}_\tau = \beta_t$  and singleton subsets  $A'_\tau = \text{act}_\tau(u), \forall \tau \leq t$ .  $\blacksquare$

The next lemma relates the estimates  $\hat{g}_t(u)$  with the optimal values  $g_t(\text{OPT}_t(u))$  (Equation (6.4.11)).

**Lemma 6.12.** Fix round  $t > 0$  and a non-increasing sequence of scalars  $\hat{\beta}_\tau$  such that  $\hat{\beta}_\tau \leq \beta_\tau, \forall \tau \leq t$ , and  $\beta_\tau \cdot |A_\tau| \leq \gamma_\tau$ . Then, with probability at least  $1 - T^{-2}$  it holds that:

$$\sum_{\tau=1}^t \hat{\beta}_\tau \sum_{u \in A_\tau} \hat{g}_\tau(u) - \sum_{\tau=1}^t \hat{\beta}_\tau \sum_{u \in A_\tau} g_\tau(u) \leq 5 \log(T) \sum_{\tau=1}^t \hat{\beta}_\tau |A_\tau| + 2 \ln T$$

*Proof.* From Equation (6.5.38), we can relate function  $\hat{g}(\cdot)$  with function  $\tilde{g}^-(\cdot)$  as follows:

$$\begin{aligned} \sum_{\tau=1}^t \hat{\beta}_\tau \sum_{u \in A_\tau} \hat{g}_\tau(u) &= \sum_{\tau=1}^t \hat{\beta}_\tau \sum_{u \in A_\tau} \tilde{g}_\tau^-(u) + \sum_{\tau=1}^t \hat{\beta}_\tau \sum_{u \in A_\tau} \frac{\beta_\tau (3 + 4 \log T)}{\pi_\tau(u)} \\ &\leq \sum_{\tau=1}^t \hat{\beta}_\tau \sum_{u \in A_\tau} g_\tau(u) + \sum_{\tau=1}^t \hat{\beta}_\tau \sum_{u \in A_\tau} \frac{\beta_\tau (3 + 4 \log T)}{\pi_\tau(u)} + 2 \ln T \end{aligned} \quad (6.5.45)$$

where the last inequality comes from Equation (6.5.42) (Lemma 6.11). Re-arranging, for Equation (6.5.45) we have:

$$\sum_{\tau=1}^t \hat{\beta}_\tau \sum_{u \in A_\tau} \hat{g}_\tau(u) - \sum_{\tau=1}^t \hat{\beta}_\tau \sum_{u \in A_\tau} g_\tau(u) \leq \underbrace{\sum_{\tau=1}^t \hat{\beta}_\tau \sum_{u \in A_\tau} \frac{\beta_\tau (3 + 4 \log(T))}{\pi_\tau(u)}}_{\Gamma_3} + 2 \ln T \quad (6.5.46)$$

Next, we focus on term  $\Gamma_3$ :

$$\begin{aligned} \Gamma_3 &= \sum_{\tau=1}^t \hat{\beta}_\tau \sum_{u \in A_\tau} \frac{\beta_\tau (3 + 4 \log(T))}{\pi_\tau(u)} \leq \sum_{\tau=1}^t \hat{\beta}_\tau \sum_{u \in A_\tau} \frac{\beta_\tau \cdot |A_\tau| (3 + 4 \log(T))}{\gamma_\tau} \quad (\pi_\tau(\cdot) \geq \gamma_\tau / |A_\tau|) \\ &\leq (3 + 4 \log T) \sum_{\tau=1}^t \hat{\beta}_\tau |A_\tau| \end{aligned} \quad (6.5.47)$$

where the last inequality comes from the fact that by assumption  $\beta_\tau |A_\tau| \leq \gamma_\tau$ . Substituting Equation (6.5.47) in Equation (6.5.46) we have that:

$$\sum_{\tau=1}^t \widehat{\beta}_\tau \sum_{u \in A_\tau} \widehat{g}_\tau(u) - \sum_{\tau=1}^t \widehat{\beta}_\tau \sum_{u \in A_\tau} g_\tau(u) \leq 5 \log(T) \sum_{\tau=1}^t \widehat{\beta}_\tau |A_\tau| + 2 \ln T$$

■

The next lemma formalizes Equation (6.4.12) and Equation (6.4.13).

**Lemma 6.13** (Singleton sets). *Fix any round  $t$  and any tree node  $u$  that is active at round  $t$ . With probability at least  $1 - T^{-2}$ , we have that:*

$$\sum_{\tau=1}^t \widehat{g}_\tau(\text{act}_\tau(u)) \geq \sum_{\tau=1}^t g_\tau(\text{OPT}_{[t]}(u)) - \frac{4 \ln T}{\beta_t} - 4\sqrt{td} \ln T$$

Moreover, if round  $t$  is the zoom-in round for node  $u$  and arm  $y \in u$  then the following also holds:

$$\sum_{\tau=1}^t \widehat{g}_\tau(\text{act}_\tau(u)) \leq \sum_{\tau=1}^t g_\tau(y) + 9tL(u) \ln T + \frac{\ln T}{\beta_t} + 4\sqrt{td} \ln T$$

*Proof.* We start from the lower bound which holds for any  $t \in [\tau_0(u), \tau_1(u)]$ . From Equation (6.5.37), we can first relate function  $\widehat{g}(\cdot)$  with function  $\widetilde{g}^+(\cdot)$  as follows:

$$\begin{aligned} \sum_{\tau=1}^t \widehat{g}_\tau(\text{act}_\tau(u)) &= \sum_{\tau=1}^t \widetilde{g}_\tau^+(\text{act}_\tau(u)) + \sum_{\tau=1}^t \frac{4 \log(T) \beta_\tau}{\pi_\tau(\text{act}_\tau(u))} \\ &\geq \sum_{\tau=1}^t g_\tau(\text{act}_\tau(u)) + \sum_{\tau=1}^t \frac{4 \log(T) \beta_\tau}{\pi_\tau(\text{act}_\tau(u))} - \frac{2 \ln T}{\beta_t} \quad (\text{Eq. (6.5.43) of Cor. 6.3}) \\ &\geq \sum_{\tau=1}^t g_\tau(\text{OPT}_{[t]}(u)) - 4\sqrt{td} \ln T - \underbrace{\sum_{\tau=1}^t L(\text{act}_\tau(u)) + \sum_{\tau=1}^t \frac{4 \log(T) \beta_\tau}{\pi_\tau(\text{act}_\tau(u))}}_{\Gamma} - \frac{2 \ln T}{\beta_t} \end{aligned} \tag{6.5.48}$$

where the last inequality comes from the one-sided-Lipschitzness lemma (Lemma 6.10). From the zooming invariant (Lemma 6.3) we have that:

$$\sum_{\tau=1}^t \frac{\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} \geq tL(u) - \frac{1}{\beta_t} \tag{6.5.49}$$

From Lemma 6.6 we can upper bound the inherited bias of node  $u$  with respect to its diameter:  $\sum_{\tau=1}^t L(\text{act}_\tau(u)) \leq 4t \log(t)L(u)$ . Combining this with Equation (6.5.49) we obtain that the term  $\Gamma$  of Equation (6.5.48) is:

$$\Gamma \geq 4tL(u)(\log T - \log t) - \frac{1}{\beta_t} \geq -\frac{1}{\beta_t}$$

Thus, Equation (6.5.48) becomes:

$$\sum_{\tau=1}^t \hat{g}_\tau(\text{act}_\tau(u)) \geq \sum_{\tau=1}^t g_\tau(\text{OPT}_{[t]}(u)) - \frac{4 \ln T}{\beta_t} - 4\sqrt{td} \ln T$$

This concludes our proof for the lower bound.

We now turn our attention to the upper bound. From Equation (6.5.38), we can first relate function  $\hat{g}(\cdot)$  with function  $\tilde{g}^-(\cdot)$  as follows:

$$\begin{aligned} \sum_{\tau=1}^t \hat{g}_\tau(\text{act}_\tau(u)) &= \sum_{\tau=1}^t \tilde{g}_\tau^-(\text{act}_\tau(u)) + \sum_{\tau=1}^t \frac{(3 + 4 \log T)\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} \\ &\leq \sum_{\tau=1}^t g_\tau(\text{act}_\tau(u)) + \sum_{\tau=1}^t \frac{(3 + 4 \log T)\beta_\tau}{\pi_\tau(\text{act}_\tau(u))} + \frac{2 \ln T}{\beta_t} \quad (\text{Eq. (6.5.44) of Cor. 6.3}) \\ &\leq \sum_{\tau=1}^t g_\tau(\text{OPT}_\tau(\text{act}_\tau(u))) + \underbrace{\sum_{\tau=1}^t \frac{(3 + 4 \log T)\beta_\tau}{\pi_\tau(\text{act}_\tau(u))}}_{\Gamma_1} + \frac{2 \ln T}{\beta_t} \end{aligned} \tag{6.5.50}$$

where the last inequality comes from the fact that  $g_\tau(y) \leq g_\tau(\text{OPT}_\tau(v))$  for any node  $v$  and point  $y \in v$ . From the one-sided-Lipschitzness lemma (Lemma 6.10) we have that:

$$\begin{aligned} \sum_{\tau=1}^t g_\tau(\text{OPT}_\tau(\text{act}_\tau(u))) &\leq \sum_{\tau=1}^t g_\tau(y) + \sum_{\tau=1}^t L(\text{act}_\tau(u)) + 4\sqrt{td} \ln T \\ &\leq \sum_{\tau=1}^t g_\tau(y) + 4tL(u) \log T + 4\sqrt{td} \ln T \end{aligned} \tag{6.5.51}$$

where the last inequality comes from Lemma 6.6.

For term  $\Gamma_1$  since  $t = \tau_1(u)$  we can apply the aggregate zoom-in rule and obtain:

$$\Gamma_1 \leq (3 + 4 \log T)tL(u) - \frac{3 + 4 \log T}{\beta_t} \leq 5tL(u) \ln T - \frac{\ln T}{\beta_t}$$

where the last inequality comes from the fact that  $3 + 4 \log T \geq \ln T$  and  $3 + 4 \log T \leq 5 \ln T$ . Substituting the upper bound for  $\Gamma_1$  and Equation (6.5.51) in Equation (6.5.50) we get the result.  $\blacksquare$

#### 6.5.4 REGRET BOUNDS

Now, we bring the pieces together to derive the regret bounds. As in Theorem 6.1, we first derive a “raw” regret bound Equation (6.3.3) in terms of the parameters, then we tune the parameters and derive a cleaner regret bound Equation (6.3.4) in terms of  $|A_T|$  (the number of active nodes). Then, we upper-bound  $|A_T|$  via the adversarial zooming dimension to derive the final regret bound Equation (6.3.5).

For this section, we condition on the clean event for the algorithm, denoted  $\mathcal{E}_{\text{clean}}^{\text{alg}}$ . This event states that, essentially, all relevant high-probability properties from Chapter 6.5.3 actually hold. More formally, for each round  $t \in [T]$ , using failure probability  $\delta = T^{-2}$ , the following events hold:

- the event in Lemma 6.12 holds with  $\hat{\beta}_\tau \equiv \beta_\tau$ .
- the event in Lemma 6.13 holds for each nodes  $u$  that are active in round  $t$  and  $y = \text{OPT}_{[t]}(y)$ .

Taking appropriate union bounds, we find that  $\Pr[\mathcal{E}_{\text{clean}}^{\text{alg}}] \geq 1 - \delta$ . In the subsequent lemmas, we condition on both clean events,  $\mathcal{E}_{\text{clean}}^{\text{rew}}$  and  $\mathcal{E}_{\text{clean}}^{\text{alg}}$ , without further notice. Recall that  $C_{\text{prod}}(u) := \prod_v |\mathcal{C}(v)|$ , as in Lemma 6.1.

**Lemma 6.14.** *ADVERSARIALZOOMING incurs regret:*

$$R(T) \leq 4\sqrt{Td} \ln T + \frac{6 \ln T}{\beta_T} + \frac{\ln(T) \ln(C_{\text{dbl}} \cdot |A_T|)}{\eta_T} + \sum_{t=1}^T 4(1 + 2 \log T)\beta_t + 42 \log^2(T)\gamma_t.$$

*Proof.* We begin with transitioning from Lemma 6.9 to a statement that only includes realized rewards (rather than estimated ones). To do so, by the clean event  $\mathcal{E}_{\text{clean}}^{\text{alg}}$  (namely, Lemma 6.13) and

the fact that  $\text{OPT}_{[T]}(u^*) = x_{[T]}^*$  it follows that

$$\sum_{t=1}^T \hat{g}_t(\text{act}_t(u^*)) \geq \sum_{t=1}^T g_t(x_{[T]}^*) - \frac{2 \ln(1/\delta)}{\beta_T} - 4\sqrt{Td} \ln T$$

As a result, the lower bound of quantity  $Q$  becomes:

$$\ln \left( \frac{\Phi_T(\eta_T)}{\Phi_0(\eta_0)} \right) \geq \sum_{t=1}^T g_t(x_{[T]}^*) - \frac{4 \ln T}{\beta_T} - \frac{\ln(|A_T| \cdot \mathcal{C}_{\text{prod}}(u^*))}{\eta_T} - 4\sqrt{Td} \ln T$$

By the clean event  $\mathcal{E}_{\text{clean}}^{\text{alg}}$  (namely, Lemma 6.12) for  $t = T$  and  $\hat{\beta}_\tau = \beta_\tau$  we have that

$$\begin{aligned} \sum_{t=1}^T \beta_t \sum_{u \in A_t} \hat{g}_t(u) &\leq \sum_{t=1}^T \beta_t \sum_{u \in A_t} g_t(u) + 5 \log(T) \sum_{t=1}^T \beta_t |A_t| + 2 \ln T \\ &\leq \sum_{t=1}^T \beta_t + 5 \log(T) \sum_{t=1}^T \gamma_t + 2 \ln T \end{aligned}$$

where the last inequality is due to the fact that by assumption  $\beta_t |A_t| \leq \gamma_t$  and  $g_t(\cdot) \in [0, 1]$ . As a result, Equation (6.5.33) becomes:

$$\sum_{t=1}^T \hat{Q}_t \leq \sum_{t=1}^T g_t(x_t) + 4(1 + 2 \log T) \sum_{t=1}^T \beta_t + 42 \log^2(T) \sum_{t=1}^T \gamma_t \quad (6.5.52)$$

As a result, from Lemma 6.9 we get that:

$$\begin{aligned} \sum_{\tau=1}^T g_t(x_{[\tau]}^*) - \sum_{t=1}^T g_t(x_t) \\ \leq \frac{6 \ln T}{\beta_T} + \frac{\ln(|A_T| \cdot \mathcal{C}_{\text{prod}}(u^*))}{\eta_T} + 4(1 + 2 \log T) \sum_{t=1}^T \beta_t + 42 \log^2(T) \sum_{t=1}^T \gamma_t \end{aligned} \quad (6.5.53)$$

Using the fact that  $\mathcal{C}_{\text{prod}}(u^*) \leq C_{\text{dbl}}^{\log T}$  (Lemma 6.19), Equation (6.5.53) becomes:

$$\begin{aligned} \sum_{\tau=1}^T g_t(x_{[\tau]}^*) - \sum_{t=1}^T g_t(x_t) \leq & 4\sqrt{Td} \ln T + \frac{6 \ln T}{\beta_T} + \frac{\ln(T) \ln(C_{\text{dbl}} \cdot |A_T|)}{\eta_T} + \\ & + 4(1 + 2 \log T) \sum_{t=1}^T \beta_t + 42 \log^2(T) \sum_{t=1}^T \gamma_t \end{aligned} \quad (6.5.54)$$

■

**Lemma 6.15.** Tune  $\beta_t, \gamma_t, \eta_t$  as in Equation (6.3.6). *ADVERSARIALZOOMING* incurs regret

$$\sum_{t=1}^T g_t(x^*) - \sum_{t=1}^T g_t(x_t) \leq 100 \ln^2(T) \sqrt{d \cdot |A_T| \cdot T \cdot \ln(|A_T| \cdot T^3) \ln(C_{\text{dbl}} \cdot |A_T|)}.$$

*Proof.* First, we verify that the tuning in Equation (6.3.6) satisfies the various assumptions made throughout the proof. It is easy to see that the assumptions in Equation (6.3.2) hold; we omit the easy details. As for the assumption (6.2.10) on  $\tilde{\beta}_t$  parameters:

$$\begin{aligned} \sum_{\tau=t}^{t'} \tilde{\beta}_\tau &= \sum_{\tau=t}^{t'} \sqrt{\frac{2 \ln(|A_\tau| \cdot T^3) \ln(C_{\text{dbl}} \cdot |A_\tau|)}{d \cdot \ln^2 T} |A_\tau| \cdot \tau} \\ &\leq \sum_{\tau=t}^{t'} \frac{1}{\sqrt{\tau}} \quad (2 \ln(|A_\tau| \cdot T^3) \ln(C_{\text{dbl}} \cdot |A_\tau|) \leq d \cdot |A_\tau| \text{ asymptotically}) \\ &\leq \frac{\ln(T) \cdot \sqrt{2d \cdot |A_t| \cdot \ln(|A_t| \cdot T^3) \ln(C_{\text{dbl}} \cdot |A_t|)}}{2} \sum_{\tau=t}^{t'} \frac{1}{\sqrt{\tau}} \quad (d \cdot |A_t| \geq 4) \\ &\leq \frac{\ln(T) \cdot \sqrt{2d \cdot |A_t| \cdot \ln(|A_t| \cdot T^3) \ln(C_{\text{dbl}} \cdot |A_t|)}}{2} \int_t^{t'} \frac{1}{\sqrt{\tau}} d\tau \\ &\leq \ln(T) \cdot \sqrt{2d \cdot |A_t| \cdot \ln(|A_t| \cdot T^3) \ln(C_{\text{dbl}} \cdot |A_t|)} (\sqrt{t'} - \sqrt{t}) \\ &\leq \frac{1}{\beta_{t'}} - \frac{1}{\beta_t} \quad (|A_t| \leq |A_{t'}|, \forall t \leq t') \end{aligned}$$

Now, let us plug in the parameter values into Lemma 6.14. For the term  $\sum_{t=1}^T \gamma_t$ , we have:

$$\begin{aligned} \sum_{t=1}^T \gamma_t &= (2 + 4 \log T) \sum_{t=1}^T \sqrt{\frac{2|A_t| \cdot \ln(|A_t| \cdot T^3) \ln(C_{\text{dbl}} \cdot |A_t|)}{t \cdot d \cdot \ln^2(T)}} \\ &\leq 5 \frac{\log T}{\ln T} \sqrt{|A_T| \cdot \ln(|A_T| \cdot T^3) \ln(C_{\text{dbl}} \cdot |A_T|)} \cdot \sum_{t=1}^T \sqrt{\frac{1}{d \cdot t}} \quad (|A_t| \leq |A_T|, \forall t \in [T]) \\ &\leq \frac{3}{d} \sqrt{|A_T| \cdot \ln(|A_T| \cdot T^3) \ln(C_{\text{dbl}} \cdot |A_T|) \cdot T}, \end{aligned}$$

where the last inequality comes from the fact that since  $1/\sqrt{t}$  is a non-increasing function:  $\sum_{t=1}^T \frac{1}{\sqrt{t}} \leq \int_0^T \frac{1}{\sqrt{t}} dt = 2\sqrt{T}$ . Substituting gives the result. ■

Next, we upper-bound the “estimated gap” using properties of multiplicative weights (this is

Equation (6.4.14) in Section 6.4).

**Lemma 6.16** (Estimated Gap of Zoomed-In Node). *If node  $u$  gets zoomed-in at round  $t$ , then:*

$$\sum_{\tau \in [t]} \widehat{g}_\tau(\text{act}_\tau(u_t^*)) - \sum_{\tau \in [t]} \widehat{g}_\tau(\text{act}_\tau(u)) \leq \frac{\ln \left( \frac{9C_{\text{prod}}(u_t^*)}{C_{\text{prod}}(u) \cdot \beta_t^2} \right)}{\eta_t}.$$

*Proof.* Substituting  $\pi_t(u) = (1 - \gamma)p_t(u) + \frac{\gamma}{|A_t|}$  in Lemma 6.8, we have that:

$$\begin{aligned} p_t(u) &\geq \left( \frac{\beta_t}{e^{L(u)} - 1 - \tilde{\beta}_t} - \frac{\gamma_t}{|A_t|} \right) \cdot \frac{1}{1 - \gamma_t} \\ &\geq \frac{\beta_t}{e^{L(u)} - 1 - \tilde{\beta}_t} - \frac{\gamma_t}{|A_t|} \quad (0 < \gamma_t < 1/2) \\ &= \frac{\beta_t}{e^{L(u)} - 1 - \tilde{\beta}_t} - \beta_t \quad (\beta_t \cdot |A_t| \leq \gamma_t) \\ &\geq \frac{\beta_t^2}{e^{L(u)}} \end{aligned} \tag{6.5.55}$$

where the last inequality is due to the fact that  $L(u) \leq L(u_0) = 1$ .

We denote by  $W_{t,\eta}$  the quantity

$$\sum_{u' \in A_t} \frac{1}{C_{\text{prod}}(u')} \exp \left( \eta \sum_{\tau=1}^t \widehat{g}_\tau(\text{act}_\tau(u')) \right)$$

By the definition of the probability being the normalized weight:  $p_t(u) = w_{t,\eta_t}(u)/W_{t,\eta_t}$ , so Equation (6.5.55) becomes:

$$w_{t,\eta_t}(u) \geq \frac{\beta_t^2}{e^{L(u)}} \cdot W_{t,\eta_t} \geq \frac{\beta_t^2}{e^{L(u)}} \cdot w_{t,\eta_t}(u_t^*) \tag{6.5.56}$$

where the second inequality comes from the fact that the weights are non-negative. Using the definition of the weights update rule from Equation (6.2.3), Equation (6.5.56) becomes:

$$\frac{1}{C_{\text{prod}}(u)} \exp \left( \eta_t \sum_{\tau=1}^t \widehat{g}_\tau(\text{act}_\tau(u)) \right) \geq \frac{\beta_t^2}{e^{L(u)}} \cdot \frac{1}{C_{\text{prod}}(u_t^*)} \exp \left( \eta_t \sum_{\tau=1}^t \widehat{g}_\tau(\text{act}_\tau(u_t^*)) \right)$$

Taking logarithms on both sides of the latter, reordering and dividing by  $\eta_t$  we get that:

$$\sum_{\tau=1}^t \widehat{g}_\tau(\text{act}_\tau(u_t^*)) - \sum_{\tau=1}^t \widehat{g}_\tau(\text{act}_\tau(u)) \leq \frac{2 \ln(1/\beta_t) + L(u) + \ln(C_{\text{prod}}(u^*)/C_{\text{prod}}(u))}{\eta_t} \tag{6.5.57}$$

Using the fact that  $L(u) \leq \ln 3$  we get the result. ■

Using this, we can derive a statement about the actual gap in the realized rewards for any round. We first state an auxiliary lemma, which we will use in order to derive a simplified statement of the actual gap in the realized rewards for any round.

**Lemma 6.17** (Upper Bound on Number of Nodes Created). *In the worst-case, at any round  $t$ , the total number of nodes that AdvZoomDim has activated is:  $|A_t| \leq (9t)^{d/(d+2)}$ , where  $d = \text{CovDim}$ .*

*Proof.* In order to find the worst-case number of nodes that can be activated, we are going to be thinking from the perspective of an *adversary*, whose goal is to make the algorithm activate as many nodes as possible. However, from Lemma 6.7 we have established that in order to zoom-in on a node  $u$  a probability mass of at least  $\mathcal{M}(u) \geq \frac{1}{9L^2(u)}$  is required. So, the best that the adversary can do is activate nodes “greedily”, i.e., when  $\mathcal{M}(u) = \frac{1}{9L^2(u)}$  to make node  $u$  become active.

Let  $\zeta$  be diameter of the smallest node  $\mathbf{u}$  that the adversary has been able to construct after  $t$  rounds, i.e.,  $\zeta = L(\mathbf{u})$  where  $\mathbf{u} = \arg \min_{u \in A_t} L(u)$ . We fix the constant multiplier of the covering dimension to be  $\gamma = 1$ . Then, from the definition of the covering dimension, in the worst case, the adversary has been able to activate  $\zeta^{-\text{CovDim}}$  such nodes. However, since for each of these nodes the probability mass spent on the node is at least  $1/9\zeta^2$ , and the total probability mass available for  $t$  rounds is  $t$  (i.e., a total of 1 for each round) we have that in the worst case:

$$\zeta^{-\text{CovDim}} \cdot \frac{1}{9\zeta^2} = t$$

Solving the latter we get that in this case  $\zeta = (9t)^{-1/(\text{CovDim}+2)}$ . By the definition of the covering dimension this means that the maximum number of active nodes at this round is:

$$|A_t| \leq (9t)^{\text{CovDim}/(\text{CovDim}+2)}$$

■

**Lemma 6.18.** *Suppose node  $u$  is zoomed-in in some round  $t = \tau_1(u)$ . For each arm  $x \in u$ , its adversarial*

gap at time  $t$  is

$$t \cdot \text{AdvGap}_t(x) \leq 9tL(u) \ln T + 4\sqrt{td} \cdot \ln T + \frac{\ln T + \ln \left( \frac{9C_{\text{prod}}(u_t^*)}{\beta_t^2 C_{\text{prod}}(u)} \right)}{\eta_t}$$

*Simplified:*

$$\text{AdvGap}_t(x) \leq \mathcal{O} \left( \ln(T) \cdot \sqrt{\ln(C_{\text{dbl}} \cdot t)} \cdot L(u) \right).$$

Before proving Lemma 6.18, we upper bound  $C_{\text{prod}}(u)$  in terms of the doubling constant  $C_{\text{dbl}}$ .

**Lemma 6.19.**  $C_{\text{prod}}(u) \leq C_{\text{dbl}}^{\log T}$ .

*Proof.* The maximum number of children that a node  $u$  can activate is  $|\mathcal{C}(u)| \leq C_{\text{dbl}}$ . As a result, and since the total number of ancestors that node  $u$  has is  $h(u)$  we have that:

$$C_{\text{prod}}(u) \leq C_{\text{dbl}}^{h(u)} \leq C_{\text{dbl}}^{\log T}$$

where the last inequality is due to the fact that  $\forall v : h(v) \leq \log T$  (Lemma 6.4). ■

*Proof of Lemma 6.18.* To prove this lemma, we apply the clean event  $\mathcal{E}_{\text{clean}}^{\text{alg}}$  (namely, Lemma 6.13) in Lemma 6.16 and for notational convenience we use:  $t = \tau_1(u)$ .

$$\sum_{\tau=1}^t g_\tau(x_{[t]}^*) - \sum_{\tau=1}^t g_\tau(x) \leq 9tL(u) \ln T + 4\sqrt{td} \ln T + \frac{\ln T}{\beta_t} + \frac{\ln \left( \frac{9C_{\text{prod}}(u_t^*)}{\beta_t^2 C_{\text{prod}}(u)} \right)}{\eta_t} \quad (6.5.58)$$

where  $x \in u$ . This proves the first part of Lemma 6.18. We move next to proving its simplified version. Since  $C_{\text{prod}}(u) > 1$  and  $C_{\text{prod}}(u) \leq C_{\text{dbl}}^{\log T}$  for all nodes  $u$ , the latter can be relaxed to:

$$\text{AdvGap}_t(x) \leq 9L(u) \ln T + \frac{\ln T}{t\beta_t} + 4\sqrt{\frac{d}{t}} \cdot \ln T + \frac{2\ln T \cdot \ln (C_{\text{dbl}} \cdot t \cdot |A_t|)}{t\eta_t}$$

where we have used again the notation  $t = \tau_1(u)$ . Substituting the values for  $\eta_t$  and  $\beta_t$  from Theo-

rem 6.1 the latter becomes:

$$\begin{aligned}
\text{AdvGap}_t(x) &\leq 9L(u) \ln T + \frac{\ln(T) \cdot \sqrt{|A_t| d \ln(T)}}{\sqrt{2t \ln(|A_t| T^3) \cdot \ln(C_{\text{dbl}} \cdot |A_t|)}} + 4\sqrt{\frac{d}{t} \cdot \ln(T)} \\
&\quad + \frac{2 \ln(T) \sqrt{\ln(C_{\text{dbl}} \cdot t \cdot |A_t|) \cdot |A_t| \cdot d \cdot \ln(T)}}{\sqrt{2t \cdot \ln(|A_t| \cdot T^3)}} \\
&\leq 9L(u) \ln T + \frac{\ln(T) \sqrt{|A_t| d}}{\sqrt{2t}} + \frac{\ln(T) \sqrt{d}}{\sqrt{t}} + \frac{2 \ln(T) \sqrt{|A_t| \cdot d \cdot \ln(C_{\text{dbl}} \cdot t \cdot |A_t|)}}{\sqrt{t}} \\
&\leq 9L(u) \ln T + \frac{4 \ln(T) \sqrt{d \cdot |A_t| \cdot \ln(C_{\text{dbl}} \cdot t \cdot |A_t|)}}{\sqrt{t}}
\end{aligned} \tag{6.5.59}$$

where the second inequality comes from the fact that  $|A_t| > 1$  and  $C_{\text{dbl}} > 1$ .

We know from Lemma 6.17 that at any round  $t$ , the number of active nodes is upper bounded as  $|A_t| \leq (9t)^{\text{CovDim}/(\text{CovDim}+2)}$ . As a result, Equation (6.5.59) becomes:

$$\text{AdvGap}_t(x) \leq 9L(u) \ln T + \frac{4 \cdot 3^{\frac{\text{CovDim}}{\text{CovDim}+2}} \cdot \ln(T) \cdot \sqrt{d \cdot \frac{2\text{CovDim}+2}{\text{CovDim}+2} \cdot \ln(C_{\text{dbl}} \cdot 9t)}}}{t^{\frac{1}{\text{CovDim}+2}}}$$

But,  $t^{-\frac{1}{\text{CovDim}+2}}$  is the smallest possible diameter (i.e.,  $L(v)$ ) that an adversary could have been able to force our algorithm to construct, as we stated earlier for the chosen  $\zeta$  of Lemma 6.17. In other words and using the fact that  $(\text{CovDim}+1)/(\text{CovDim}+2) \leq 1$  we get:

$$\text{AdvGap}_t(x) \leq 7 \cdot 3^{\frac{\text{CovDim}}{\text{CovDim}+2}} \cdot \ln(T) \cdot \sqrt{d \cdot \ln(9C_{\text{dbl}} \cdot t)} \cdot L(u)$$

Or simply:

$$\text{AdvGap}_t(x) \leq \mathcal{O}\left(\ln(T) \cdot \sqrt{d \cdot \ln(C_{\text{dbl}} \cdot t)} \cdot L(u)\right)$$

where the  $\mathcal{O}(\cdot)$  notation hides constant terms. ■

Using Lemma 6.18 we can derive the regret statement of **ADVERSARIALZOOMING** in terms of **AdvZoomDim** (i.e., Equation (6.3.5)). We clarify that in order to achieve Equation (6.3.5) we will eventually have to prove a stricter bound on the number of active nodes.

We are now ready to state the regret guarantee of **ADVERSARIALZOOMING** using the notion of **AdvZoomDim**. This corresponds to the derivation in Equation (6.3.5).

**Lemma 6.20.** *With probability at least  $1 - T^{-2}$ , ADVERSARIALZOOMING incurs regret:*

$$R(T) \leq 100 \cdot \frac{3z}{z+2} \cdot d^{\frac{1}{2}} \cdot T^{\frac{z+1}{z+2}} \cdot (\gamma \cdot C_{\text{dbl}} \cdot 2^z \cdot \log T)^{\frac{1}{z+2}} \sqrt{2 \ln(T^3 \cdot \gamma \cdot C_{\text{dbl}} \cdot \log T) \cdot \ln(C_{\text{dbl}} \cdot T)}$$

where  $z = \text{AdvZoomDim}$  and  $d = \text{CovDim}$ .

*Proof.* Starting from  $L(\text{root}) = 1$  every time that a zoom-in happens on ADVERSARIALZOOMING, the diameter of the interval gets halved. We call this process an increase in *scale*. Let  $\mathcal{S}$  be the total number of scales from node `root` to the smallest created node after  $T$  rounds.  $\mathcal{S}$  depends on the problem instance. Let  $Z(\varepsilon_i)$  (where  $i \in [\mathcal{S}]$ ) denote the number of nodes  $u$  with diameter  $L(u) \geq \varepsilon_i$  with gap  $\text{AdvGap}_\rho(x) \leq \mathcal{O}(\varepsilon_i \cdot \ln(T) \cdot \sqrt{d \cdot \ln(C_{\text{dbl}} \cdot \rho)})$  for  $x \in \mathcal{A}_{\text{repr}}$  such that  $x \in u$  at some round  $\rho = c/\varepsilon_i^2$ , where  $c > 0$  is a constant.

In order for a node  $u$  to get zoomed-in, a probability mass of  $\mathcal{M}(u) \geq 1/9L^2(u)$  is required. Since  $\mathcal{M}(u) = \sum_{\tau=\tau_0(u)}^{\tau_1(u)} \pi_\tau(u) \leq \tau_1(u) - \tau_0(u) + 1 \leq \tau_1(u)$ , then this means that for the de-activation time of node  $u$  it holds that  $\tau_1(u) \geq 1/9L^2(u)$ . We choose  $\varepsilon_i$  and  $c$  in a way that  $\rho = \tau_1(u)$  for the maximum number of nodes. Then, all of these nodes belong in the set  $Z(\varepsilon_i)$  and all of them get zoomed-in. When a node  $u$  gets zoomed-in at most  $|\mathcal{C}(u)| \leq C_{\text{dbl}}$  children-nodes get activated. Inductively, after  $T$  rounds and given that there is a total probability mass of  $T$ , we have the following:

$$C_{\text{dbl}} \cdot Z(\varepsilon_0) \cdot \frac{1}{9\varepsilon_0^2} + C_{\text{dbl}} \cdot Z(\varepsilon_1) \cdot \frac{1}{9\varepsilon_1^2} + \dots + C_{\text{dbl}} \cdot Z(\varepsilon_{\mathcal{S}}) \cdot \frac{1}{9\varepsilon_{\mathcal{S}}^2} = T \Leftrightarrow \frac{C_{\text{dbl}}}{9} \sum_{i \in [\mathcal{S}]} \frac{Z(\varepsilon_i)}{\varepsilon_i^2} = T \quad (6.5.60)$$

On the other hand, the number of active nodes after  $T$  rounds is at most the number of zoomed in nodes  $u$  at each scale, multiplied by  $|\mathcal{C}(u)| \leq C_{\text{dbl}}$ .

$$|A_T| \leq C_{\text{dbl}} \sum_{i \in [\mathcal{S}]} Z(\varepsilon_i) \quad (6.5.61)$$

We next cover each  $Z(\varepsilon_i)$  with sets of diameter  $\varepsilon_i/2$  (to guarantee that each center of the nodes

belongs in only one set). Using the definition of the zooming dimension, Eq. (6.5.61) becomes:

$$\begin{aligned}
T &= \frac{C_{\text{dbl}}}{9} \sum_{i \in [\mathcal{S}]} \gamma \cdot \frac{1^{-z} \varepsilon_i^{-z}}{\varepsilon_i^2} = \frac{\gamma \cdot C_{\text{dbl}} \cdot 2^z}{9} \sum_{i \in [\mathcal{S}]} \varepsilon_i^{-z-2} \\
&= \frac{\gamma \cdot C_{\text{dbl}} \cdot 2^z}{9} \sum_{i \in [\mathcal{S}]} 2^{-(\mathcal{S}-i)(z+2)} \varepsilon_{\mathcal{S}}^{-z-2} \quad (\varepsilon_i = 2\varepsilon_{t+1}) \\
&\leq \varepsilon_{\mathcal{S}}^{-z-2} \cdot \frac{\gamma \cdot C_{\text{dbl}} \cdot 2^z}{9} \cdot \mathcal{S}
\end{aligned}$$

where  $z = \text{AdvZoomDim}$  and  $\gamma$  is the chosen constant multiplier from the definition of the zooming dimension. As a result, re-arranging, the latter inequality becomes:

$$\varepsilon_{\mathcal{S}} = \left( \frac{9T}{\gamma \cdot C_{\text{dbl}} \cdot \mathcal{S} \cdot 2^z} \right)^{-\frac{1}{z+2}} \quad (6.5.62)$$

From Equation (6.5.61) and plugging in the zooming dimension definition we have that:

$$\begin{aligned}
|A_T| &= \gamma \cdot C_{\text{dbl}} \cdot 2^z \sum_{i \in [\mathcal{S}]} \varepsilon_i^{-z} \leq \gamma \cdot C_{\text{dbl}} \cdot 2^z \cdot \mathcal{S} \cdot \varepsilon_{\mathcal{S}}^{-z} \\
&\leq (\gamma \cdot C_{\text{dbl}} \cdot 2^z \cdot \mathcal{S})^{\frac{2}{z+2}} \cdot (9T)^{\frac{z}{z+2}}
\end{aligned} \quad (6.5.63)$$

Since  $\mathcal{S} \leq \log T$  and using this bound together with Lemma 6.15 we have that with probability at least  $1 - T^{-2}$  we get the result.  $\blacksquare$

## 6.6 AdvZoomDim UNDER STOCHASTIC REWARDS

We study AdvZoomDim under stochastic rewards, and upper-bound it by ZoomDim, thus proving Lemma 6.2. We prove this lemma in a slightly more general formulation, with an explicit dependence on the representative set  $\mathcal{A}_{\text{repr}} \subset \mathcal{A}$ .

**Lemma 6.21.** *Consider an instance of Lipschitz bandits with stochastic rewards. Fix a representative set  $\mathcal{A}_{\text{repr}} \subset \mathcal{A}$ . For any  $\gamma > 0$ , with probability at least  $1 - 1/T$  it holds that:*

$$\text{ZoomDim}_{\gamma,f} \leq \text{AdvZoomDim}_{\gamma,f} \leq \text{ZoomDim}_{\gamma}, \quad (6.6.1)$$

where  $f = \left(O\left(\sqrt{d} \cdot \ln^2(T) \cdot \ln(|\mathcal{A}_{\text{repr}}|)\right)\right)^{\log(C_{\text{dbl}}) - \text{ZoomDim}_\gamma}$ .

As before, we use  $d$  to denote the covering dimension, for some constant multiplier  $\gamma_0 > 0$ , and we suppress the logarithmic dependence on  $\gamma_0$ .

To prove Lemma 6.21, we relate the stochastic and adversarial gap of each arm.

**Proposition 6.1.** *Consider an instance of Lipschitz bandits with stochastic rewards. Fix time  $t$ . For any arm  $x \in \mathcal{A}_{\text{repr}}$ , with probability at least  $1 - 1/T$  it holds that:*

$$|\text{AdvGap}_t(x) - \text{Gap}(x)| \leq 3 \sqrt{\frac{2 \ln(T \cdot |\mathcal{A}_{\text{repr}}|)}{t}}.$$

*Proof.* Let us first fix an arm  $x \in \mathcal{A}_{\text{repr}}$ . We apply the Azuma-Hoeffding inequality (Lemma C.2) to the following martingale:

$$Y_t = \sum_{\tau=1}^t (g_\tau(x^*) - g_\tau(x)) - t \cdot \text{Gap}(x)$$

where  $x^* = \arg \max_{x \in \mathcal{A}} \mu(x)$ , and  $\mu(x)$  is the mean reward for arm  $x$  in the stochastic instance.

Noting that  $|Y_{t+1} - Y_t| \leq 1$ , and fixing  $\delta > 0$ , we have  $\Pr[Y_t \geq \sqrt{2t \cdot \ln(1/\delta)}] \leq \delta$ .

Unwrapping the definition of  $Y_t$ , with probability at least  $1 - \delta$  we have:

$$\sum_{\tau=1}^t (g_\tau(x^*) - g_\tau(x)) \leq \text{Gap}(x) + \sqrt{\frac{2 \ln(1/\delta)}{t}} \quad (6.6.2)$$

We next lower bound the left-hand side of Equation (6.6.2).

$$\begin{aligned} \frac{1}{t} \sum_{\tau=1}^t (g_\tau(x^*) - g_\tau(x)) &= \frac{1}{t} \sum_{\tau=1}^t \left[ g_\tau(x^*) - g_\tau(x) + g_\tau(x_{[t]}^*) - g_\tau(x_{[t]}^*) \right] \\ &= \text{AdvGap}_t(x) + \underbrace{\frac{1}{t} \sum_{\tau=1}^t g_\tau(x^*) - \frac{1}{t} \sum_{\tau=1}^t g_\tau(x_{[t]}^*)}_{\Delta} \end{aligned}$$

We move on to lower bounding quantity  $\Delta$ .

$$\Delta \geq \sum_{\tau=1}^t \mathbb{E}[g_\tau(x^*)] - \sqrt{\frac{2 \ln(1/\delta)}{t}} - \sum_{\tau=1}^t \mathbb{E}\left[g_\tau(x_{[t]}^*)\right] - \sqrt{\frac{2 \ln(1/\delta)}{t}} \geq -2\sqrt{\frac{2 \ln(1/\delta)}{t}} \quad (6.6.3)$$

where the last inequality comes from the fact that  $x^*$  is the mean-optimal arm. Substituting Equation (6.6.3) in Equation (6.6.2) we obtain:

$$\text{AdvGap}_t(x) \leq \text{Gap}(x) + 3\sqrt{\frac{2 \ln(1/\delta)}{t}} \quad (6.6.4)$$

Similarly, using the symmetric side of the Azuma-Hoeffding inequality (Lemma C.2) for martingale  $Y_t$ , we obtain that with probability at least  $1 - \delta$ :

$$\begin{aligned} \text{Gap}(x) &\leq \frac{1}{t} \sum_{\tau=1}^t (g_\tau(x^*) - g_\tau(x)) + \sqrt{\frac{2 \ln(1/\delta)}{t}} \\ &\leq \frac{1}{t} \sum_{\tau=1}^t \left( g_\tau(x_{[t]}^*) - g_\tau(x) \right) + \sqrt{\frac{2 \ln(1/\delta)}{t}} \quad (x_{[t]}^* = \arg \max_{x \in \mathcal{A}} \sum_{\tau \in [t]} g_\tau(x)) \\ &= \text{AdvGap}_t(x) + \sqrt{\frac{2 \ln(1/\delta)}{t}} \end{aligned} \quad (6.6.5)$$

where the equality comes from the definition of  $\text{AdvGap}_t(x)$ . Putting Equation (6.6.4) and Equation (6.6.5) together we get that *for the fixed arm  $x \in \mathcal{A}_{\text{repr}}$* :

$$\Pr [| \text{AdvGap}_t(x) - \text{Gap}(x) |] \leq 3\sqrt{\frac{2 \ln(1/\delta)}{t}} \quad (6.6.6)$$

In order to guarantee that Equation (6.6.6) applies *for all arms  $y \in \mathcal{A}_{\text{repr}}$* , we apply the union bound; let  $\delta'$  be the failure probability. Tuning the failure probability of the original event to be  $\delta = \frac{1}{T \cdot |\mathcal{A}_{\text{repr}}|}$  (*i.e.*, the failure probability of the final event is  $\delta' = 1/T$ ) we get the stated result. ■

We are now ready for the proof of Lemma 6.21.

*Proof of Lemma 6.21.* We first prove the rightmost inequality in Equation (6.6.1). For that, we fix an instance of the stochastic Lipschitz MAB, and we focus on an arm  $x$  for which  $\text{AdvGap}_t(x) \leq 30 \ln(T) \cdot \sqrt{d \ln(C_{\text{dbl}} \cdot T)} \cdot \varepsilon$ , and  $\varepsilon = (3\sqrt{t})^{-1}$ . Then, from Proposition 6.1 we obtain that with probability at least  $1 - 1/T$ :

$$\begin{aligned} \text{Gap}(x) &\leq 30 \ln(T) \cdot \sqrt{d \ln(C_{\text{dbl}} \cdot T)} \cdot \varepsilon + \varepsilon \sqrt{18 \ln(T \cdot |\mathcal{A}_{\text{repr}}|)} \\ &\leq 31 \cdot \sqrt{d} \cdot \ln(T) \cdot \sqrt{\ln(C_{\text{dbl}} \cdot T) \cdot \ln(T \cdot |\mathcal{A}_{\text{repr}}|)} \cdot \varepsilon \end{aligned}$$

From the definition of  $\text{ZoomDim}$ , the set of the aforementioned arms can be covered by

$$\underbrace{\gamma \cdot 31 \cdot \sqrt{d} \cdot \ln(T) \cdot \sqrt{\ln(C_{\text{dbl}} \cdot T) \cdot \ln(T \cdot |\mathcal{A}_{\text{repr}}|)}}_{\varepsilon'} \cdot \varepsilon^{-\text{ZoomDim}_\gamma}$$

sets of diameter  $\varepsilon'$  for some constant  $\gamma > 0$ . Equivalently, the set of these arms can be covered with

$$\begin{aligned} & \gamma \cdot \left(\frac{\varepsilon'}{\varepsilon}\right)^{\log(C_{\text{dbl}})} \left(31 \cdot \sqrt{d} \cdot \ln(T) \cdot \sqrt{\ln(C_{\text{dbl}} \cdot T) \cdot \ln(T \cdot |\mathcal{A}_{\text{repr}}|)}\right)^{-\text{ZoomDim}_\gamma} \cdot \varepsilon^{-\text{ZoomDim}_\gamma} \\ &= \gamma \cdot \underbrace{\left(31 \cdot \sqrt{d} \cdot \ln(T) \cdot \sqrt{\ln(C_{\text{dbl}} \cdot T) \cdot \ln(T \cdot |\mathcal{A}_{\text{repr}}|)}\right)^{\log(C_{\text{dbl}}) - \text{ZoomDim}_\gamma}}_f \cdot \varepsilon^{-\text{ZoomDim}_\gamma} \end{aligned}$$

sets of diameter  $\varepsilon$ . Since we defined  $\text{AdvZoomDim}$  to be the infimum dimension for which the above holds,  $\text{AdvZoomDim}_{\gamma,f} \leq \text{ZoomDim}_\gamma$ . In order to conclude the proof and derive the stated expression for  $f$ , note that  $C_{\text{dbl}} \leq T$  and  $\ln(T \cdot |\mathcal{A}_{\text{repr}}|) \leq \ln(T) \cdot \ln(|\mathcal{A}_{\text{repr}}|)$ . This concludes the proof of this side of the inequality.

We proceed to prove the leftmost inequality in (6.6.1). We fix an instance of the stochastic Lipschitz MAB, and we focus on an arm  $x$  for which  $\text{Gap}(x) \leq \varepsilon$ , where  $\varepsilon = (3\sqrt{t})^{-1}$ . Then, from Proposition 6.1 we get that with probability at least  $1 - 1/T$  it holds that:

$$\text{AdvGap}_t(x) \leq \varepsilon + 3\varepsilon\sqrt{18\ln T} \leq 30\ln(T) \cdot \sqrt{d \cdot \ln(C_{\text{dbl}} \cdot T)} \cdot \varepsilon \quad (6.6.7)$$

From the definition of  $\text{AdvZoomDim}_{\gamma,f}$ , the set of these arms can be covered by  $\gamma \cdot f \cdot \varepsilon^{-\text{AdvZoomDim}_{\gamma,f}}$  sets of diameter  $\varepsilon$  for some constant  $\gamma \cdot f > 0$ . Since we defined  $\text{ZoomDim}$  to be the infimum dimension for which the above holds:  $\text{ZoomDim}_{\gamma,f} \leq \text{AdvZoomDim}_{\gamma,f}$ . ■

In order to get the result stated in Lemma 6.2, we note that  $|\mathcal{A}_{\text{repr}}| \leq T^{O(\text{poly}(d))}$ .

## 6.7 ADVZOOMDIM EXAMPLES

We provide a flexible “template” for examples with small  $\text{AdvZoomDim}$ . We instantiate this template for some concrete examples, which apply generically to adversarial Lipschitz bandits as well as to a more specific problem of adversarial dynamic pricing.

### Theorem 6.2: Example Template of AdvZoomDim

Fix action space  $(\mathcal{A}, \mathcal{D})$  and time horizon  $T$ . Let  $d$  be the covering dimension.<sup>a</sup>

Consider problem instances  $\mathcal{I}_1, \dots, \mathcal{I}_M$  with stochastic rewards, for some  $M$ . Suppose each  $\mathcal{I}_i$  has a constant zooming dimension  $z$ , with some fixed multiplier  $\gamma > 0$ . Construct the *combined instance*: an instance with adversarial rewards, where each round is assigned in advance (but otherwise arbitrarily) to one of these stochastic instances.

Then  $\text{AdvZoomDim} \leq z$  with probability at least  $1 - 1/T$ , with multiplier

$$\gamma' = \gamma \cdot \left( \mathcal{O} \left( M \ln(T) \sqrt{d \ln(C_{\text{dbl}} \cdot T) \cdot \ln(T \cdot |\mathcal{A}_{\text{repr}}|)} \right) \right)^{\log(C_{\text{dbl}}) - z}.$$

The representative set  $\mathcal{A}_{\text{repr}} \subset \mathcal{A}$  (needed to specify  $\text{AdvZoomDim}$ ) can be arbitrary.

This holds under the following assumptions on problem instances  $\mathcal{I}_i$ :

- There are disjoint subsets  $S_1, \dots, S_M \subset \mathcal{A}$  such that each stochastic instance  $\mathcal{I}_i, i \in [M]$  assigns the same “baseline” mean reward  $b_i$  to all arms in  $\cup_{j \neq i} S_j$ , mean rewards *at least*  $b_i$  to all arms inside  $S_i$ , and mean rewards *at most*  $b_i$  to all arms in  $\mathcal{A} \setminus \sup_{j \in [M]} S_j$ .
- For each stochastic instance  $\mathcal{I}_i, i \in [M]$ , the difference between the largest mean reward and  $b_i$  (called the *spread*) is at least  $1/3$ .

---

<sup>a</sup>As before, the covering dimension is with some constant multiplier  $\gamma_0 > 0$ , and we suppress the logarithmic dependence on  $\gamma_0$ .

We prove the theorem through a series of claims. Fix  $\varepsilon > 0$ . Let us argue about arms that are inclusively  $\varepsilon$ -optimal in the combined instance. Recall that such arms satisfy  $\text{AdvGap}_t(\cdot) \leq \mathcal{O}(\varepsilon \cdot \ln(T) \cdot \sqrt{d \cdot \ln(C_{\text{dbl}} \cdot T) \cdot \ln(T \cdot |\mathcal{A}_{\text{repr}}|)})$  for some round  $t \geq \varepsilon^{-2}/9$ , and we need to cover them with sets of diameter at most  $\varepsilon$ . So, fix some round  $t \geq \varepsilon^{-2}/9$ .

We now define the *induced adversarial instance* at round  $t$  which is comprised by the *realized* rewards example described above. Let  $I_\tau$  denote the stochastic instance that we face at round  $\tau \leq t$ . In the induced adversarial instance the total reward of an arm  $x \in \mathcal{A}$  after  $t$  rounds is:  $G(x) = \sum_{\tau \in [t]} g_{I_\tau}(x)$ , where for all  $i \in [M]$ ,  $g_i(x)$  is drawn from distribution with mean  $\mu_i(x)$ .

Let  $f_i$  denote the empirical frequency of instance  $\mathcal{I}_i$  up to time  $t$ . Then, the mean reward for an

arm  $x \in \mathcal{A}$  in the induced adversarial instance takes the following form  $\mu(x) = \sum_{i \in [M]} f_i \cdot \mu_i(x)$ , where  $\mu_i(\cdot)$  is the mean reward function for instance  $\mathcal{I}_i$ . For the family of examples we consider, this translates to arms  $x \in S_i$  for some  $i \in [M]$  having mean reward

$$\mu(x) = \sum_{\substack{j \in [M]; \\ j \neq i}} f_j \cdot b_j + f_i \cdot \mu_i(x),$$

and arms  $y \in \mathcal{A} \setminus \cup_{i \in [M]} S_i$  having mean reward  $\mu(y) = \sum_{j \in [M]} f_j \cdot b_j$ .

The next lemma relates the stochastic gap of the induced adversarial instance (denoted by  $\text{Gap}(x)$ ) with the stochastic gap of one of the instances  $\{\mathcal{I}\}_{i \in [M]}$  (denoted by  $\text{Gap}_i(x)$ ). To be more specific, let  $x_j^*$  denote the optimal arm for instance  $\mathcal{I}_j$ , and  $\bar{x}^*$  the mean-optimal arm in the induced stochastic instance (i.e.,  $\bar{x}^* = \arg \max_{x \in \mathcal{A}} \mu(x)$ ). Then,  $\text{Gap}_i(x) = \mu_i(x_i^*) - \mu_i(x)$ , but  $\text{AdvGap}(x) = \mu(\bar{x}^*) - \mu(x)$ .

**Lemma 6.22.** *If arm  $x$  has  $\text{Gap}(x) \leq \varepsilon$  in the induced adversarial instance, then  $\text{Gap}_i(x) \leq \mathcal{O}(\varepsilon)$  in some instance  $\mathcal{I}_i$ .*

In order to prove Lemma 6.22, we need some auxiliary claims, which we prove below. We prove first that the mean-optimal arm is among the peaks of the stochastic instances.

**Claim 6.1.** *The mean-optimal arm of the induced stochastic instance  $\bar{x}^*$  belongs in  $\{x_i^*\}_{i \in [M]}$ .*

*Proof.* We prove this claim by contradiction; assume that there exists arm  $y \notin \{x_i^*\}_{i \in [M]}$  and  $\mu(y) > \mu(x_i^*)$ ,  $\forall i \in [M]$ . If  $y$  does not belong in any of the  $S_i$ 's, then its total mean-payoff at round  $t$  is at most  $\mu(y) = \sum_{i \in [M]} f_i \cdot b_i$ . Pick any peak point  $y' \in \{x_i^*\}_{i \in [M]}$  at random such that  $y' \in S_j$  where instance  $\mathcal{I}_j$  is such that  $f_j > 0$ . Then, by our setting's assumptions:

$$\mu(y) = \sum_{i \in [M]} f_i \cdot b_i < \sum_{i \in [M]; i \neq j} f_i \cdot b_i + f_j \cdot \mu_j^* = \mu(y')$$

which is a contradiction to the fact that  $y$  is the mean-optimal arm. ■

Note here that we can only make this claim because the sets  $\{S_i\}_{i \in [M]}$  are disjoint; otherwise, we would not be able to claim that fixing point  $y'$  instead of  $y$  only changes the reward received for the rounds where instance  $\mathcal{I}_j$  appears.

We next state a useful property regarding the empirical frequency for instance  $\mathcal{I}_{i^*}$  (*i.e.*, the instance for which  $\bar{x}^* \in S_{i^*}$ ).

**Claim 6.2.** *The empirical frequency of instance  $\mathcal{I}_{i^*}$  is  $f_{i^*} \geq 1/3M$ .*

*Proof.* Let  $\phi^*$  the index of the instance with the maximum empirical frequency, *i.e.*,  $f_{\phi^*} = \max_{i \in [M]} f_i$ . Then, it holds that  $f_{\phi^*} \geq 1/M$ . Indeed, notice that if this is not the case:  $\sum_{i \in [M]} f_i \leq M \cdot f_{\phi^*} < 1$ , which is a contradiction with the fact that this sum is equal to 1.

Assume next that the instance with the maximum frequency,  $\phi^*$ , is different than the instance  $i^*$  for which it holds that  $\bar{x}^* \in S_{i^*}$ . Then it holds that:

$$\mu(\bar{x}^*) \geq \mu(x_{\phi^*}^*) \Leftrightarrow \sum_{\substack{j \in [M]; \\ j \neq i^*}} f_j \cdot b_j + f_{i^*} \cdot \mu_{i^*}^* \geq \sum_{\substack{j \in [M]; \\ j \neq \phi^*}} f_j \cdot b_j + f_{\phi^*} \cdot \mu_{\phi^*}^*$$

where the first inequality is by the definition of  $\bar{x}^*$  being the mean-optimal arm. Rearranging the above, we get:

$$f_{i^*} \cdot (\mu_{i^*}^* - b_{i^*}) \geq f_{\phi^*} \cdot (\mu_{\phi^*}^* - b_{\phi^*}) \Leftrightarrow f_{i^*} \geq \frac{\mu_{\phi^*}^* - b_{\phi^*}}{\mu_{i^*}^* - b_{i^*}} \quad (6.7.1)$$

where for the division we use the fact that  $\mu_i^* - b_i \geq 1/3$  for all  $i \in [M]$ . Using this, together with the fact that  $\mu_i - b_i \leq 1$ , Eq. (6.7.1) becomes:  $f_{i^*} \geq f_{\phi^*}/3$ . Substituting  $f_{\phi^*} \geq 1/M$  we get the result. ■

We proceed by characterizing the arms that *cannot* be  $\varepsilon$ -optimal in terms of their mean rewards in the induced instance; these arms  $x$  will have  $\text{Gap}(x) > \Omega(1)$ .

**Claim 6.3.** *Arms  $x \in \mathcal{A} \setminus \cup_{i \in [M]} S_i$  have  $\text{Gap}(x) \geq c$ , where  $c = \Omega(1/9M)$  is a constant.*

*Proof.* We start by the definition of  $\text{Gap}(x)$ :

$$\begin{aligned} \text{Gap}(x) &= \mu(\bar{x}^*) - \mu(x) \\ &\geq \sum_{\substack{j \in [M]; \\ j \neq i^*}} f_j \cdot b_j + f_{i^*} \cdot \mu_{i^*}^* - \sum_{j \in [M]} f_j \cdot b_j \quad (x \in \mathcal{A} \setminus \cup_{i \in [M]} S_i) \\ &= f_{i^*} \cdot (\mu_{i^*}^* - b_{i^*}) \end{aligned}$$

By assumption  $\mu_i^* - b_i \geq 1/3$  and hence, the above is lower bounded by  $\geq f_{i^*} \cdot \frac{1}{3} \geq \frac{1}{3M} \cdot \frac{1}{3}$ , where the last inequality is due to Claim 6.2. This concludes our proof. ■

Arms  $x \in \mathcal{A} \setminus \cup_{i \in [M]} S_i$  are not the only ones for which  $\text{Gap}(x) > \Omega(1)$ , as shown next.

**Claim 6.4.** *Arms  $x \in S_i$  with  $i \in [M] : f_i \leq 1/18M$  have  $\text{Gap}(x) \geq c'$ , where  $c' = \Omega(1/18M)$  is a constant.*

*Proof.* We start again by the definition of  $\text{Gap}(x)$ :

$$\begin{aligned}
\text{Gap}(x) &= \mu(\bar{x}^*) - \mu(x) \geq \mu(\bar{x}^*) - \mu(x_i^*) && (\mu(x) \leq \mu(x_i^*), \forall x \in S_i) \\
&= \sum_{\substack{j \in [M]; \\ j \neq i^*}} f_j \cdot b_j + f_{i^*} \cdot \mu_{i^*}^* - \sum_{\substack{j \in [M]; \\ j \neq i}} f_j \cdot b_j - f_i \cdot \mu_i^* \\
&= f_{i^*} \cdot (\mu_{i^*}^* - b_{i^*}) - f_i \cdot (\mu_i^* - b_i) \\
&\geq \frac{1}{3M} \cdot \frac{1}{3} - f_i \cdot (\mu_i^* - b_i) && (\text{Claim 6.2 and } \mu_i^* - b_i \geq 1/3) \\
&\geq \frac{1}{9M} - \frac{1}{18M} \cdot 1 && (\mu_i^* - b_i \leq 1 \text{ and } f_i \leq 1/18M) \\
&= \frac{1}{18M}
\end{aligned}$$

This concludes our proof. ■

We are now ready to prove Lemma 6.22.

*Proof of Lemma 6.22.* Due to Claims 6.3 and 6.4 for  $\text{Gap}(x) \leq \varepsilon$  where  $\varepsilon < o(1)$ , we only need to focus on arms  $x$  for which it holds that  $x \in S_j$  for some  $j \in [M]$  such that  $f_j > 1/18M$ .

Fix an arm  $x$  such that  $\text{Gap}(x) \leq \varepsilon$  and let  $j$  be such that  $x \in S_j$  and  $f_j > 1/18M$ . Then:

$$\begin{aligned}
\varepsilon &\geq \text{Gap}(x) = \mu(\bar{x}^*) - \mu(x) = \mu(\bar{x}^*) - \mu(x_j^*) + \mu(x_j^*) - \mu(x) \\
&= \mu(\bar{x}^*) - \mu(x_j^*) + \underbrace{(\mu_j^* - \mu_j(x)) \cdot f_j}_{\text{Gap}_j(x)} \\
&\geq \text{Gap}_j(x) \cdot f_j && (\mu(\bar{x}^*) \geq \mu(x_j^*)) \\
&\geq \frac{1}{18M} \cdot \text{Gap}_j(x) && (\text{Claim 6.4})
\end{aligned}$$

Hence, for arms  $x$  such that  $\text{Gap}(x) \leq \varepsilon$  it holds that  $\text{Gap}_j(x) \leq 18M \cdot \varepsilon = \mathcal{O}(\varepsilon)$  for some stochastic instance  $j \in [M]$ . This concludes our proof. ■

In the next step of the proof, we will connect an arm's stochastic gap of the induced adversarial instance with its adversarial gap. The result is similar in flavor to Proposition 6.1. Recall that  $\text{Gap}(x)$  corresponds to the stochastic gap for arm  $x$  in the induced adversarial instance.

**Proposition 6.2.** *Fix time  $t$ . For any arm  $x \in \mathcal{A}_{\text{repr}}$ , with probability at least  $1 - 1/T$  for the induced adversarial instance of the example it holds that:*

$$|\text{AdvGap}_t(x) - \text{Gap}(x)| \leq 3 \sqrt{\frac{2 \ln(T \cdot |\mathcal{A}_{\text{repr}}|)}{t}}$$

where  $\text{Gap}(x)$  is the stochastic gap of the mean rewards in the induced adversarial instance.

*Proof.* The proof is similar to the one of Proposition 6.1, but there are a few subtle differences.

Observe that:

$$\mathbb{E} \left[ \frac{1}{t} G(x) \right] = \mathbb{E} \left[ \frac{1}{t} \sum_{\tau \in [t]} g_{I_\tau}(x) \right] = \sum_{i \in [M]} f_i \cdot \mu_i(x) = \mu(x)$$

Fixing the sequence  $\{I_\tau\}_{\tau \in [t]}$ , we can define the following martingale:

$$Y_{t'} = \sum_{\tau \in [t']} (g_{I_\tau}(\bar{x}^*) - g_{I_\tau}(x)) - t' \cdot (\mu^{t'}(\bar{x}^*) - \mu^{t'}(x))$$

where  $\mu^{t'}(x)$  is the mean reward of arm  $x \in \mathcal{A}$  in the induced adversarial instance of sequence  $\{I_\tau\}_{\tau \leq t'}$  with  $t' \leq t$ . In other words, denoting by  $f_i^{t'}$  the empirical frequency of instance  $i$  in sequence  $\{I_\tau\}_{\tau \in t'}$  with  $t' \leq t$ :  $\mu^{t'}(x) = \sum_{i \in [M]} f_i^{t'} \cdot \mu(x)$ . To see that  $Y_{t'}$  is indeed a martingale, let  $f_i^{t'}$  be the empirical frequency of instance  $i$  in sequence  $\{I_\tau\}_{\tau \in t'}$  with  $t' \leq t$  and note that:

$$\begin{aligned} \mathbb{E}[Y_{n+1} | Y_1, \dots, Y_n] &= \mathbb{E} \left[ \sum_{\tau \in [n+1]} (g_{I_\tau}(\bar{x}^*) - g_{I_\tau}(x)) - (n+1) \cdot (\mu^{n+1}(\bar{x}^*) - \mu^{n+1}(x)) | Y_1, \dots, Y_n \right] \\ &= \sum_{i \in [M]} f_i^{n+1} \cdot \mu(\bar{x}^*) - \sum_{i \in [M]} f_i^{n+1} \cdot \mu(\bar{x}^*) + Y_n = Y_n \end{aligned}$$

where the second equation is due to the definition of  $\mu^{t'}(x)$  and the fact that:

$$\mathbb{E} [g_{I_\tau}(x) \cdot \mathbf{1}\{\tau \leq t' \leq T\}] = \sum_{i \in [M]} f_i^{t'} \cdot \mu_i(x).$$

Applying the Azuma-Hoeffding inequality (Lemma C.2), and since  $|Y_{t'+1} - Y_{t'}| \leq 1$  for rewards bounded in  $[0, 1]$  we have:  $\Pr[Y_{t'} \geq \sqrt{2t \ln(1/\delta)}] \leq \delta, \forall \delta > 0$ . The rest of the steps in the proof are similar to the proof of Proposition 6.1 and hence we omit them. ■

We are now ready to complete the proof of Theorem 6.2.

*Theorem 6.2.* It remains to show that the arms for which  $\text{AdvGap}_t(x)$  is adequately small in the induced adversarial instance, have small stochastic gap  $\text{Gap}(x)$  with high probability. Then, by Lemma 6.22 we can relate  $\text{AdvGap}_t(x)$  with the stochastic gap  $\text{Gap}_i(x)$  for some instance  $i \in [M]$ .

Let arm  $x \in \mathcal{A}_{\text{repr}}$  be such that  $\text{AdvGap}_t(x) \leq 30\varepsilon \cdot \ln(T) \cdot \sqrt{d \ln(C_{\text{dbl}} \cdot T)}$  for  $\varepsilon = (3\sqrt{t})^{-1}$ . Then, from Proposition 6.2 it holds that with probability at least  $1 - 1/T$ :

$$\begin{aligned} \text{Gap}(x) &\leq 30\varepsilon \cdot \ln(T) \cdot \sqrt{d \ln(C_{\text{dbl}} \cdot T)} + \varepsilon \sqrt{18 \ln(T \cdot |\mathcal{A}_{\text{repr}}|)} \\ &\leq \underbrace{31\varepsilon \cdot \ln(T) \cdot \sqrt{d \cdot \ln(C_{\text{dbl}} \cdot T) \cdot \ln(T \cdot |\mathcal{A}_{\text{repr}}|)}}_{\varepsilon'} \end{aligned}$$

From Lemma 6.22, since  $\text{Gap}(x) \leq \varepsilon'$ , there exists an instance  $j \in [M]$  such that  $\text{Gap}_j(x) \leq 18M\varepsilon'$ . From the definition of zooming dimension for this instance  $j$  these arms can be covered by at most  $\gamma \cdot (18M\varepsilon')^{-z}$  sets of diameter  $18M\varepsilon'$ , where  $z = \text{ZoomDim}_\gamma$ . So, these arms can be covered by:

$$\begin{aligned} &\gamma \cdot \left(\frac{18M\varepsilon'}{\varepsilon}\right)^{\log(C_{\text{dbl}})} \cdot (18M\varepsilon')^{-z} \\ &= \gamma \cdot \left(558 \cdot M \cdot \ln(T) \cdot \sqrt{d \cdot \ln(C_{\text{dbl}} \cdot T) \cdot \ln(T \cdot |\mathcal{A}_{\text{repr}}|)}\right)^{\log(C_{\text{dbl}})-z} \cdot \varepsilon^{-z} \end{aligned}$$

sets of diameter  $\varepsilon$ . Since the  $\text{AdvZoomDim}$  is the smallest dimension needed to cover these arms (because originally  $\text{AdvGap}_t(x) \leq O(\varepsilon \ln(T) \sqrt{d \ln(C_{\text{dbl}} \cdot T)})$ , then  $\text{AdvZoomDim}_{\gamma'} \leq z$  for  $\gamma' = \gamma \cdot \mathcal{O}\left(M \ln(T) \sqrt{d \ln(C_{\text{dbl}} \cdot T) \cdot \ln(T \cdot |\mathcal{A}_{\text{repr}}|)}\right)$ ). ■

We emphasize the generality of this theorem. First, the assignment of rounds in the combined

instance to the stochastic instances  $\mathcal{I}_1, \dots, \mathcal{I}_M$  can be completely arbitrary: e.g., the stochastic instances can appear consecutively in “phases” of arbitrary duration, or they can be interleaved in an arbitrary way. Second, the subsets  $S_1, \dots, S_M \subset \mathcal{A}$  can be arbitrary. Third, each stochastic instance  $\mathcal{I}_i, i \in [M]$  can behave arbitrarily on  $S_i$ , as long as  $\mu_i^* - b'_i \geq 1/3$ , where  $\mu_i^*$  and  $b'_i$  are, resp., the largest and the smallest rewards on  $S_i$ . The baseline reward can be any  $b_i \leq b'_i$ , and outside  $\cup_{j \in [M]} S_j$  one can have any mean rewards that are smaller than  $b_i$ .

Now we can take examples from stochastic Lipschitz bandits and convert them to (rather general) examples for adversarial Lipschitz bandits. Rather than attempt a comprehensive survey of examples for the stochastic case, we focus on two concrete examples that we adapt from (Kleinberg et al., 2019): concave rewards and “distance to the target”. For both examples, we posit action space  $\mathcal{A} = [0, 1]$  and distances  $\mathcal{D}(x, y) = |x - y|$ . Note that the covering dimension is  $d = 1$ . The expected reward of each arm  $x$  in a given stochastic instance  $i$  is denoted  $\mu_i(x)$ . In both examples,  $\mu_i(\cdot)$  will have a single peak, denoted  $x_i^* \in \mathcal{A}$ , and the baseline reward satisfies  $\mu_i(x_i^*) - b_i \geq 1/3$ .

- *Concave rewards*: For each instance  $i$ ,  $\mu_i(x)$  is a strongly concave function on  $S_i$ , in the sense that  $\mu''_i(x)$  exists and  $\mu''_i(x) < \varepsilon$  for some  $\varepsilon > 0$ . Then the zooming dimension is  $z = 1/2 < d = 1$ , with appropriately chosen multiplier  $\gamma > 0$ . <sup>††</sup>
- *“Distance to target”*: For each instance  $i$ ,  $\mu_i(x) = \min(0, \mu_i(x_i^*) - \mathcal{D}(x, x_i^*))$  for all arms  $x \in S_i$ . Then the zooming dimension is in fact  $z = 0$ .

**Dynamic pricing.** Both examples can be “implemented” in stochastic dynamic pricing, via appropriate distributions over the private values. In fact, the concave-rewards case is known as *regular demands*, a very common assumption in theoretical economics. Below are some concrete instantiations for the two examples.

Note that based on definition, the mean reward for dynamic pricing takes the following form:

$$\mu_i(x) = x \cdot \Pr_{v_i}[x \leq v_i] = x \cdot (1 - \Pr_{v_i}[v_i < x])$$

---

<sup>††</sup>A somewhat subtle point is that an algorithm tailored to concave-rewards instances can achieve  $\tilde{\mathcal{O}}(\sqrt{T})$  regret, e.g., via uniform discretization (Kleinberg and Leighton, 2003). However, this algorithm would not be optimal in the worst case: it would only achieve regret  $\tilde{\mathcal{O}}(T^{3/4})$  whereas the worst-case optimal regret rate is  $\tilde{\mathcal{O}}(T^{2/3})$ .

As a result, in order to provide concrete instantiations, one only needs to control the distribution of  $v_i$ , *i.e.*, term  $\Pr_{v_i}[v_i < x]$ . At a high level, the distribution of  $v_i$ 's expresses the preferences of the buyers' population.

For the concave-rewards instantiation, if  $v_i \sim \text{Unif}[0, 1]$ , then  $\mu_i(x) = x(1-x) = x - x^2$  the mean-reward  $\mu_i(x)$  satisfies the strong concavity requirements, and hence, the zooming dimension of instance  $\mathcal{I}_i$  is  $z = 1/2$ . In this example,  $S_i$  can be any subset of  $[0, 1]$ . In order to obtain disjoint sets, for the different instances, one needs to use different uniform distributions  $\text{Unif}[a_j, b_j]$  for different instances  $\{j \in [M]\}$ .

As for the distance-to-target instantiation, if  $v_i$  is drawn from distribution with cdf

$$F(x) = 1 - \frac{1}{4x} + \frac{|x - 1/2|}{x}, x \in [0.3, 0.7] = S_i \quad (6.7.2)$$

Then,  $\mu_i(x) = \frac{1}{4} - |x - \frac{1}{2}| = \mu_i^* - \mathcal{D}(x, x_i^*)$ . In order to construct disjoint sets  $\{S_i\}_{i \in [M]}$  one needs to use the general form of Equation (6.7.2) as follows:

$$F(x) = 1 - \frac{a_i}{x} + \frac{|x - b_i|}{x}, x \in S_i$$

where  $a_i = \mu_i^*$  and  $b_i = x_i^*$ .

## **Part III**

# **Behavioral Model Misspecifications in Incentive-Aware ML**

# 7

## Behavioral Model Misspecifications as Adversarial Corruptions: Contextual Search and Extensions to Pricing

### 7.1 CHAPTER OVERVIEW

For Parts I and II we focused on achieving different forms of robustness for the institution that deploys the Machine Learning algorithms against strategic agents. One core assumption that we made was that the institution knew exactly the functional form of the agents' utilities (e.g., in strategic prediction the institution did not know the original prediction of an agent, but they did

know that the goal of the agent was to maximize the probability of being chosen at the next round). In that sense, all our guarantees held true *conditional* on every agent following exactly the posited utility function. In this chapter we go beyond this assumption and we wish to address settings where there exist a few behavioral model misspecifications. Our goal is to provide algorithms for the institution that scale optimally when no model misspecification arises, but their performance degrades gracefully with said misspecifications. We study this for the general family problems known as “*contextual search*” and we model these arbitrary model misspecifications as “adversarial corruptions”/adversarial noise.

In the most standard version of contextual search at every round  $t$ , a *context*  $\mathbf{x}_t \in \mathbb{R}^d$  arrives. Associated with this context is an unknown *true value*  $v_t \in \mathbb{R}$ , which we here assume is a linear function of the context so that  $v_t = \langle \theta^*, \mathbf{x}_t \rangle$  for some unknown vector  $\theta^* \in \mathbb{R}^d$ , called the *ground truth*. Based on the observed context  $\mathbf{x}_t$ , the decision-maker or *learner* selects a *query*  $\omega_t \in \mathbb{R}$  with the goal of minimizing some *loss* that depends on the query as well as the true value; examples include the *absolute/symmetric loss*,  $|v_t - \omega_t|$ , and the  $\varepsilon$ -*ball loss*,  $\mathbb{1}\{|v_t - \omega_t| > \varepsilon\}$ , both of which measure discrepancy between  $\omega_t$  and  $v_t$ . Finally, the learner observes whether or not  $v_t \geq \omega_t$ , but importantly, the true value  $v_t$  is never revealed, nor is the loss that was suffered. As we saw in Chapter 1, contextual search has been the mathematical construction used to model feature-based dynamic pricing settings, like for example the algorithm with which AirBnB suggests prices to superhosts, given a context that corresponds to the features of the listing.

In this chapter, we present the first contextual search algorithms that can handle adversarial noise. In particular, we allow some agents to behave in ways that are arbitrarily inconsistent with the ground truth. Inspired by the recent line of work on stochastic bandit learning with adversarial corruptions (Lykouris et al., 2018, Gupta et al., 2019a, Zimmert and Seldin, 2021), we impose no assumptions on the order of corrupted rounds and obtain guarantees that gracefully degrade with their number while attaining near-optimality when all agents behave according to the linear model.

We first provide a unifying framework encompassing different loss functions and agent response models (Chapter 7.2). In particular, we assume that the agent behaves according to a *perceived value*  $\tilde{v}$  and the precise response model determines the transformation from true to perceived value. The loss functions can depend on either the true value (to capture parameter estimation objectives) or

the perceived value (for pricing objectives). This formulation allows us to capture adversarially corrupted agent responses, a setting not studied in prior work, as well as stochastic noise settings.

The first algorithm presented in this chapter (Chapter 7.3) works for all of the aforementioned loss functions ( $\varepsilon$ -ball, absolute, pricing). We prove that, with probability  $1 - \delta$ , it suffers *regret* of  $\mathcal{O}(C \cdot d^3 \cdot \text{poly log}(T/\delta))$ , where  $C$  is the unknown number of adversarially corrupted rounds and  $T$  is the total number of rounds (*time horizon*). The guarantee is logarithmic in  $T$  when  $C \approx 0$  and degrades gracefully as  $C$  becomes larger. The algorithm builds on the `PROJECTEDVOLUME` algorithm (Lobel et al., 2018), which is optimal for the  $\varepsilon$ -ball loss when  $C = 0$ .

The main technical advance is a method for maintaining a set of candidates for  $\theta^*$  (*knowledge set*), successively removing candidates by hyperplane cuts while ensuring that  $\theta^*$  is never removed. When  $C = 0$ , this is done via `PROJECTEDVOLUME` which removes all parameters  $\theta$  that are inconsistent with the response in a way that each *costly* query guarantees enough progress measured via the volume of the set of remaining parameters. However, when some responses are corrupted, such an aggressive elimination method may remove the ground truth  $\theta^*$  from the parameter space.

To deal with this key challenge, we run the algorithm in epochs, each corresponding to one query of `PROJECTEDVOLUME`, and only proceed to the next epoch if we can find a hyperplane cut that makes volumetric progress and does not eliminate  $\theta^*$ . We start from an easier setting where we assume a known upper bound  $\bar{c}$  on the number of corrupted responses ( $C \leq \bar{c}$ ) and only move to the next epoch when we can find a cut with enough volumetric progress that includes all parameters that are “*misclassified*” by at most  $\bar{c}$  queries, i.e., parameters that were inconsistent with the agents’ responses at most  $\bar{c}$  times. Note that  $\theta^*$  is consistent with all non-corrupted responses, and hence, it is always included in the new knowledge set (as it can only suffer at most  $\bar{c}$  misclassifications).

Our first challenge is to identify such a hyperplane cut, i.e., one that makes volumetric progress without eliminating  $\theta^*$ . As discussed, cuts associated with `PROJECTEDVOLUME` queries do make enough volumetric progress, but risk removing  $\theta^*$  due to their aggressive elimination. Interestingly, we can use ideas from convex analysis (specifically, the Carathéodory theorem) to show that, after collecting  $\mathcal{O}(d^2\bar{c})$  queries, we can combine them appropriately to produce the hyperplane cut with the desired properties. To guarantee the existence of such a cut, we identify a point in the parameter space that is outside of a convex body including all the *protected parameters* (the

ones with misclassification at most  $\bar{c}$ ) and then apply the separating hyperplane theorem.

A second challenge is that the separating hyperplane theorem does not provide a way to compute the corresponding cut. To deal with this, we use geometric techniques (volume cap arguments) to provide a sampling process that, with significant probability, identifies a point  $q$  that is sufficiently far from the aforementioned separating hyperplane. We compute this hyperplane by running the classical learning-theoretic Perceptron algorithm repeatedly using points sampled from this process.

There are two remaining, intertwined challenges. On the one hand, the running time of Perceptron depends on the number of subregions created by removing all possible combinations of  $\bar{c}$  queries which is exponential in  $\bar{c}$ . On the other hand, our algorithm needs to be agnostic to  $\bar{c}$ . We deal with both of these via a multi-layering approach introduced in ([Lykouris et al., 2018](#)) that runs multiple parallel versions of the aforementioned algorithm with only  $\bar{c} \approx \log T$  (Chapter [7.3.2](#)). This results in a final algorithm that is quasipolynomial in the time horizon and does not assume knowledge of  $C$ .

The second algorithm is based on gradient descent (Chapter [7.6](#)) and has a guarantee of  $\mathcal{O}(\sqrt{T} + C)$  for the absolute loss. This algorithm is simple and has efficient running time but does not provide logarithmic guarantees when  $C \approx 0$  and does not extend to non-Lipschitz loss functions such as the pricing loss. The key idea in its analysis lies in identifying a simple proxy loss function based on which we can run gradient descent and directly apply its corresponding regret guarantee.

### 7.1.1 RELATED WORK

The work on contextual search is closely related to dynamic pricing when facing an agent with *unknown* demand curve. In the non-contextual version, there is a single item with infinite supply that is sold: the learner at each round posts a price for the item, and the agent decides whether to buy it or not based on their valuation function. This problem was formalized by [Kleinberg and Leighton \(2003\)](#) who studied the settings where the valuation is fixed, i.i.d., and adversarial and provided optimal pricing-loss regret guarantees of  $\Theta(\log \log T)$ ,  $\Theta(\sqrt{T})$ , and  $\Theta(T^{2/3})$  respectively. The present chapter studies a contextual extension that falls between the first and the third category, since all agents behave according to the same valuation except for  $C$  of them.

**Contextual Search and Dynamic Pricing.** At a high level, there are two approaches to handle the contextual setting. The first approach uses binary search techniques and extends the category where there exists a fixed parameter that when combined with the context determines the valuation of the agent. Such techniques are very efficient in that they result in logarithmic regret guarantees and can handle contexts that are arbitrary and even selected by an adaptive adversary. The general binary search approach was introduced by [Cohen et al. \(2020\)](#) who provided a binary search algorithm based on the ellipsoid method with a regret  $\mathcal{O}(d^2 \log(d/\varepsilon))$  for the  $\varepsilon$ -ball loss and  $\mathcal{O}(d^2 \log T)$  for the absolute and pricing loss. [Lobel et al. \(2018\)](#) improved these bounds by obtaining the optimal regret  $\mathcal{O}(d \log(d/\varepsilon))$  for the  $\varepsilon$ -ball loss and regret  $\mathcal{O}(d \log T)$  for the absolute and the pricing loss. [Paes Leme and Schneider \(2018\)](#) obtained regret guarantees of  $\mathcal{O}(d^4)$  and  $\mathcal{O}(d^4 \log \log(dT))$  for the absolute and pricing loss, which are optimal with respect to  $T$ . Finally, [Liu et al. \(2021\)](#) obtained the optimal bounds (with respect to both  $d$  and  $T$ ) of  $\mathcal{O}(d \log d)$  and  $\mathcal{O}(d \log \log T)$  for the absolute and the pricing loss respectively. These binary search techniques work by recursively refining a version space that contains the underlying parameter; this is what allows them to provide the logarithmic regret guarantees as they make exponential progress in refining the *volume* of the version space at each round. This strength comes at a cost though in that it makes them very brittle even in the presence of a few corruptions. In particular, a single mistake may render the version space incorrect and the binary nature of the feedback makes it challenging to recover. This chapter addresses this shortcoming by allowing some rounds to be arbitrarily (and even adversarially) corrupted and providing guarantees that gracefully degrade with the number of these corrupted rounds. Finally, most of the above works are not designed to handle even non-adversarial noise. The two exceptions are [Cohen et al. \(2020\)](#) who can handle a low-noise regime for all loss functions (in Section 7.5, we extend our results to this setting) and the concurrent and independent work of [Liu et al. \(2021\)](#) whose results extend to a stochastic noise model where the feedback is flipped with a fixed, constant probability; their results only hold for the absolute loss and cannot handle adversarial noise.

The second methodological approach for contextual pricing is based on statistical methods such as linear regression and the central limit theorem ([Goldenshluger and Zeevi, 2013](#), [Bastani and Bayati, 2020](#), [Javanmard and Nazerzadeh, 2019](#), [Qiang and Bayati, 2016](#), [Nambiar et al., 2019](#), [Ban](#)

and Keskin, 2021, Shah et al., 2019, Chen et al., 2021). These algorithms require the context to be i.i.d. and not adversarially selected as they separate exploration and exploitation in ways that are agnostic to the context. On the positive side, they are more robust to stochastic noise in the valuations and target the second category of valuations we discussed before, i.e., valuations that are i.i.d. Beyond the contextual setting, after the work of Kleinberg and Leighton (2003), many papers incorporated important facets of dynamic pricing such as inventory constraints, multiple products, as well as different feedback and valuation models (see e.g, (Besbes and Zeevi, 2009, Broder and Rusmevichientong, 2012, den Boer and Zwart, 2014, den Boer, 2014, Keskin and Zeevi, 2014, Roth et al., 2016, 2020, Cesa-Bianchi et al., 2019, Podimata and Slivkins, 2021)).

**Adversarial Corruptions in Learning with Bandit Feedback.** To capture arbitrary model misspecifications, we posit that agents behave according to a fixed ground truth in all but  $C$  rounds, during which they can deviate from it arbitrarily. The corruption budget  $C$  can be selected adaptively and is unknown to the algorithm designer. This model was introduced by Lykouris et al. (2018) in the context of multi-armed bandits and their results for this setting were strengthened by Gupta et al. (2019a) and Zimmert and Seldin (2021). It has been subsequently used for several other settings including linear optimization (Li et al., 2019), Gaussian bandit optimization (Bogunovic et al., 2020), assortment optimization (Chen et al., 2019), reinforcement learning (Lykouris et al., 2021), prediction with expert advice (Amir et al., 2020), learning product rankings (Golrezaei et al., 2021), and dueling bandits (Agarwal et al., 2021). This chapter differs from these in that it involves a continuous action space which requires new analytical tools, while all prior results involve discrete (potentially large) action spaces. We note that a subsequent work by Chen and Wang (2020) considers a similar setting. The approach taken at the present chapter has orthogonal strengths: we consider a more complicated contextual setting and attain logarithmic regret, whereas they focus on incorporating inventory constraints.

Apart from corruptions, there are other multi-armed bandit approaches to go beyond i.i.d. rewards (Slivkins and Upfal, 2008, Bubeck and Slivkins, 2012, Gur et al., 2014, Besbes et al., 2015, Keskin and Zeevi, 2017b, Cheung et al., 2021, Wei and Luo, 2021).

**Ulam’s Game and Noisy Binary Search.** The non-contextual version of our problem bears simi-

larities to *Ulam's game* (Ulam, 1976), where one wants to make the least number of queries to an adversary in order to identify a target number from set  $\{1, 2, \dots, n\}$ . The adversary can only give binary feedback and may lie at most  $C$  times over the course of the game, where  $C$  is known to the learner (Spencer (1992), see also Pelc (2002) for a comprehensive survey). Rivest et al. (1980) provide the optimal query complexity for this problem which is  $\Theta(\log n + C \log \log n + C \log C)$ ; further discussion on how this bound relates to our guarantees is provided at the end of Chapter 7.4.2. The algorithm proposed is intuitively a halving-type algorithm keeping track of all possible, feasible configurations for the timing of the lies. The present chapter extends this seminal paper in three directions. First, we cover the case of *unknown*  $C$ . Second, we look at the contextual version of the problem. Third, our main algorithm obtains no-regret guarantees not only for the absolute and the absolute loss, but for the *pricing* loss as well. To achieve these, our algorithms and proof techniques are completely different from Rivest et al. (1980). Ulam's game has been studied in multiple different variants (Pelc, 1987, Spencer and Winkler, 1992, Aslam and Dhagat, 1991, Dagan et al., 2018, Karp and Kleinberg, 2007, Nowak, 2008, 2009).

**Beyond our Assumptions.** Our model relies heavily on two assumptions. First, we assume that agents' valuations at each non-corrupted round are linearly dependent on the observed context  $x_t$  (possibly with the addition of a small i.i.d. idiosyncratic noise). To the best of our knowledge, this is the viewpoint taken by almost all prior work that considers binary feedback with the exception of the works of Mao et al. (2018) who consider *Lipschitz* dependence of the valuation in  $x_t$  and Shah et al. (2019) who posit an exponential relationship to the context. The second main assumption of this chapter is that the agents are *myopic*, i.e., they make decisions optimizing their utilities only for the current round, without caring about future rounds. While this is a common assumption in prior work, there have also been works on dynamic pricing mechanisms where the agents are assumed to be long-living/non-myopic and thus "*strategic*" (Amin et al., 2013, 2014, Mohri and Munoz Medina, 2014, Mohri and Medina, 2015, Feldman et al., 2016, Drutsa, 2017, Liu et al., 2018, Golrezaei et al., 2019b,a, Rhuggenaath et al., 2020, Zhiyanov and Drutsa, 2020, Romano et al., 2021, Kanoria and Nazerzadeh, 2021). Most of these works assume that agents optimize an infinite-horizon, discounted utility when making decisions. This can be viewed as a structured version of corruption as the agents only lie if they benefit from that and, at a high level, the goal of the

learner in these cases is to design algorithms that will induce (approximately) truthful behavior from the agents and thus remove their incentive to deviate from the behavioral model. In contrast, we allow for arbitrary misspecifications in particular rounds so our algorithms cannot completely eliminate deviations from the prescribed behavioral model but need to be able to handle such misspecifications.

## 7.2 MODEL

In this section, we provide a general framework (Chapter 7.2.1) to study contextual search under different agent response models (Chapter 7.2.2) and different losses (Chapter 7.2.3).

### 7.2.1 PROTOCOL

We consider the following repeated interaction between the learner and nature. Following classical works in contextual search (Cohen et al., 2020, Lobel et al., 2018) we assume that the learner has access to a parameter space  $\mathcal{K} = \{\mathbf{u} \in \mathbb{R}^d : \|\mathbf{u}\|_2 \leq 1\}$  and a context space  $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 = 1\}$ . We denote by  $\Omega = [0, 1]$  the decision space of the learner and by  $\mathcal{V} = [0, 1]$  a value space; in the pricing setting,  $\Omega$  can be thought of as the set of possible prices available to the learner and  $\mathcal{V}$  as a set of values associated with incoming agents. Domain  $\mathcal{V}$  helps express both the true value of the agents and the perceived value driving their decisions. Finally, we consider an agent response model determining the transformation from the agent's *true value* to a *perceived value* that drives the decision at each round. All of the above are known to the learner throughout the learning process.

The setting proceeds for  $T$  rounds. Before the first round, nature chooses a ground truth  $\theta^* \in \mathcal{K}$ ; this is fixed across rounds and is *not* known to the learner.  $\theta^*$  determines both the agent's *true* value function  $v : \mathcal{X} \rightarrow \mathcal{V}$  and the learner's loss function  $\ell : \Omega \times \mathcal{V} \times \mathcal{V} \rightarrow [0, 1]$ . We note that both value and loss functions are also functions of the ground truth  $\theta^*$ ; given that  $\theta^*$  is fixed throughout this process, we drop the dependence on  $\theta^*$  to ease notation. The functional form of both  $v(\cdot)$  and  $\ell(\cdot)$  as a function of the ground truth  $\theta^*$  is known to the learner but the learner does not know  $\theta^*$ . In what follows, we use  $\text{sgn}$  to denote the sign function, i.e.,  $\text{sgn}(x) = 1$  if  $x \geq 0$  and  $-1$  otherwise. For each round  $t = 1, \dots, T$ :

1. Nature chooses (potentially adaptively & adversarially) and reveals context  $\mathbf{x}_t \in \mathcal{X}$ .
2. Nature picks but *does not* reveal perceived value  $\tilde{v}_t \in \mathcal{V}$  based on the response model.
3. Learner selects query point  $\omega_t \in \Omega$  (randomized) and observes  $y_t = \text{sgn}(\tilde{v}_t - \omega_t)$ .
4. Learner incurs (but does *not* observe) loss:  $\ell(\omega_t, v(\mathbf{x}_t), \tilde{v}_t) \in [0, 1]$ .

Nature is an adaptive adversary (subject to the agent response model), i.e., it knows the learner's algorithm and the realization of all randomness up to and including round  $t - 1$  (i.e., it knows all  $\omega_\tau, \forall \tau \leq t - 1$ ), but does not know the learner's randomness at the current round  $t$ . Moreover, the learner only observes the context  $\mathbf{x}_t$  and the *binary* variable  $y_t$  as described in Steps 1 and 3 of the protocol, and has access to neither the perceived value  $\tilde{v}_t$  nor the loss  $\ell(\omega_t, v(\mathbf{x}_t), \tilde{v}_t)$ . Finally, in the pricing setting,  $y_t$  corresponds to whether the agent of round  $t$  made a purchase or not.

### 7.2.2 AGENT RESPONSE MODELS

We assume that the agents' true value function is:  $v(\mathbf{x}) = \langle \mathbf{x}, \boldsymbol{\theta}^* \rangle$  for any  $\mathbf{x} \in \mathcal{X}$  (i.e., independent of their response model). The agent response model affects the perceived value  $\tilde{v}$  at round  $t$ , which then affects both the loss incurred and the feedback observed by the learner. The agent response model that is mostly studied in contextual search works is *full rationality*. This assumes that the agent always behaves according to their true value, i.e.,  $\tilde{v}_t = v(\mathbf{x}_t) = \langle \mathbf{x}_t, \boldsymbol{\theta}^* \rangle$ . In learning-theoretic terms, this consistency with respect to a ground truth is typically referred to as *realizability*.

Our main focus in this chapter is the study of *adversarially corrupted agents*. There, nature selects the rounds where these agents arrive ( $c_t = 1$ , if adversarially corrupted agents arrive, else  $c_t = 0$ ), together with an upper bound  $C$  on this number of rounds (i.e.,  $\sum_{t \in [T]} c_t \leq C$ ). Neither the sequence  $\{c_{\mathbf{g}_t}\}_{t \in [T]}$  nor the number  $C$  are ever revealed to the learner. If  $c_t = 0$ , then nature is constrained to  $\tilde{v}_t = v(\mathbf{x}_t)$ , but can select adaptively and adversarially  $\tilde{v}_t$  if  $c_t = 1$ . This is inspired by the model of adversarial corruptions in stochastic bandit learning (Lykouris et al., 2018).

The results presented in this chapter extend to *bounded rationality* which posits that the perceived value is the true value plus some noise parameter. The noise parameter is drawn from a  $\sigma$ -subgaussian distribution  $\text{subG}(\sigma)$ , *fixed* across rounds and *known* to the learner, i.e., nature selects it before the first round and reveals it. At every round  $t$  a realized noise  $\xi_t \sim \text{subG}(\sigma)$  is drawn,

but  $\xi_t$  is never revealed to the learner. The agent's perceived value is then  $\tilde{v}_t = v(\mathbf{x}_t) + \xi_t$ . This stochastic noise model has been studied in contextual search as a way to incorporate idiosyncratic market shocks (Cohen et al., 2020).

We note that, to ease presentation, our model treats agents as different but homogeneous: each of them interacts with the learner once. The same model can also be used to model a single agent that shows up for all  $T$  rounds but is myopic in his/her choices.

### 7.2.3 LOSS FUNCTIONS AND OBJECTIVE

We study 3 variants for the learner's loss function: the  $\varepsilon$ -ball, the absolute, and the pricing loss. Abstracting away from  $t$  subscripts and dependencies on contexts  $\mathbf{x}$ , the loss  $\ell(\omega, v, \tilde{v})$  evaluates the loss of a query  $\omega$  when the true value is  $v$  and the perceived value is  $\tilde{v}$ .

The first class of loss functions includes parameter estimation objectives that estimate the value of  $\theta^*$ . One such function is the  *$\varepsilon$ -ball loss* which is defined with respect to an accuracy parameter  $\varepsilon > 0$ . The  $\varepsilon$ -ball loss is 1 if the difference between the query point  $\omega$  and the true value  $v$  is larger than  $\varepsilon$  and 0 otherwise. Formally,  $\ell(\omega, v, \tilde{v}) = \mathbb{1}\{|v - \omega| \geq \varepsilon\}$ . Another parameter estimation loss function is the *absolute* or *absolute* loss that captures the absolute difference between the query point and the true value, i.e.,  $\ell(\omega, v, \tilde{v}) = |v - \omega|$ . The aforementioned loss functions are unobservable to the learner as the true value  $v$  is latent; this demonstrates that binary feedback does not offer strictly more information than the bandit feedback as the latter reveals the loss of the selected query.

Another important objective in pricing is the revenue collected which is the price  $\omega$  in the event that the purchase occurred, i.e.,  $\tilde{v} \geq \omega$ . This can be expressed based on observable information by setting a reward equal to  $\omega$  when  $\tilde{v} \geq \omega$  and 0 otherwise. However, having this as a comparator leads to a benchmark with high objective, which tends to hinder logarithmic performance guarantees that are typical in binary search and are enabled by the fact that the loss of the comparator is 0. A loss function exploiting this structure is the *pricing loss* which is defined as the difference between the highest revenue that the learner could have achieved at this round (the agent's *perceived* value  $\tilde{v}$ ) and the revenue that the learner currently receives, i.e.,  $\omega$  if a purchase happens, and 0 otherwise. The outcome of whether a purchase happens or not is tied to whether  $\omega$  is higher or smaller than the perceived value  $\tilde{v}$ . Putting everything together:  $\ell(\omega, v, \tilde{v}) = \tilde{v} - \omega \cdot \mathbb{1}\{\omega \leq \tilde{v}\}$ .

We remark that the  $\varepsilon$ -ball and the absolute loss depend only on the true value  $v$  (and not the perceived value  $\tilde{v}$ ); indeed, when these losses are considered  $\tilde{v}$  affects only the feedback that the learner receives. That said, we define  $\ell(\cdot, \cdot, \cdot)$  with three arguments for unification purposes, since the pricing loss does depend on the feedback that the learner receives (and hence, on  $\tilde{v}$ ).

The learner's goal is to minimize regret. For adversarially corrupted agents, the loss of the best-fixed policy in hindsight is *at least* 0 and *at most*  $C$ . Hence, to simplify exposition, we slightly abuse notation and conflate loss and regret:  $R(T) = \sum_{t \in [T]} \ell(\omega_t, v(\mathbf{x}_t), \tilde{v}_t)$

It is no longer possible to provide sublinear guarantees for this quantity when facing boundedly rational agents and so we need to slightly relax the benchmark. To ease exposition, we defer further discussion on the extension to bounded rationality to Section 7.5.

### 7.3 CORRUPTED PROJECTED VOLUME: ALGORITHM AND MAIN GUARANTEE

In this section, we provide an algorithmic scheme that handles all the aforementioned agent response models and loss functions. The main result of this and the next section is an algorithm (Algorithm 7.4) for the adversarial corrupted agent response model when there is an *unknown* upper bound  $C$  on the number of corrupted agents. The regret of this algorithm is upper bounded by the following theorem. We first present the algorithm in this section and prove the stated theorem in Chapter 7.4.2.

#### Theorem 7.1: Regret of CorPV.AC

When run with an accuracy parameter  $\varepsilon > 0$  and an unknown corruption level  $C$ , CorPV.AC incurs regret  $\mathcal{O}(d^3 \cdot \log(T/\beta) \cdot \log(d/\varepsilon) \cdot \log(1/\beta) \cdot (\log T + C))$  with probability at least  $1 - \beta$  for the  $\varepsilon$ -ball loss. When run with  $\varepsilon = 1/T$ , its regret for the pricing and absolute loss is  $\mathcal{O}(d^3 \log(dT) \log(T) \cdot (\log T + C) \log(1/\beta))$  with probability at least  $1 - \beta$ . The expected runtime of the algorithm is quasi-polynomial; in particular, it is  $\mathcal{O}((d^2 \log T)^{\text{poly log } T} \cdot \text{poly}(d, \log T))$ .

#### 7.3.1 ALGORITHM FOR THE KNOWN-CORRUPTION SETTING

A useful intermediate setting is the case where we know an upper bound  $\bar{c}$  on the number of adversarial agents, i.e.,  $C \leq \bar{c}$ ; we refer to this as the  $\bar{c}$ -known-corruption setting. Our algorithm for

this setting, CorPV.KNOWN (Algorithm 7.1), builds on the PROJECTEDVOLUME algorithm of [Lobel et al. \(2018\)](#) which is optimal in terms of regret for the  $\varepsilon$ -ball loss when  $\bar{c} = 0$ . The main idea in PROJECTEDVOLUME is to maintain a *knowledge set*  $\mathcal{K}_t$  which includes all candidate parameters  $\theta$  that are *consistent* with what has been observed so far. The true parameter  $\theta^*$  is always consistent and therefore is never eliminated from the knowledge set. Further the volume of the knowledge set is intuitively a measure of progress for the algorithm. We now briefly describe the main components of PROJECTEDVOLUME and refer the reader to Appendix D.1.1 for an algorithmic sketch and more details.

Given a context  $x_t$ , there are two scenarios. Before we describe them, we define the *width* of a body  $\mathcal{K}_0$  in direction  $x$  as  $w(\mathcal{K}_0, x) = \sup_{\theta, \theta' \in \mathcal{K}_0} \langle \theta - \theta', x \rangle$ . A small width along a certain direction  $x$ , means that we have adequately learned said direction, i.e., we do not need to refine our estimate of  $\theta^*$  further in this direction. Hence, if the width of the knowledge set in the direction of  $x_t$  is  $w(\mathcal{K}_t, x_t) \leq \varepsilon$ , the algorithm makes an *exploit* query  $\omega_t = \langle x_t, \theta_t \rangle$  for *any* point  $\theta_t \in \mathcal{K}_t$  which guarantees an  $\varepsilon$ -ball loss equal to 0. Otherwise, if  $w(\mathcal{K}_t, x_t) > \varepsilon$ , then the algorithm further refines the estimate of  $\theta^*$  in the direction of  $x_t$  by making an *explore* query  $\omega_t = \langle x_t, \kappa_t \rangle$ , where  $\kappa_t$  is the (approximate) centroid of  $\mathcal{K}_t$ . For a convex body  $\mathcal{K}_0$ , the centroid is defined as  $\kappa^* = \frac{1}{\text{vol}(\mathcal{K}_0)} \int_{\mathcal{K}_0} u du$ , where  $\text{vol}(\cdot)$  denotes the volume of a set. Although computing the exact centroid of a convex set is #P-hard ([Rademacher, 2007](#)), one can efficiently approximate it ([Lobel et al., 2018](#)).

By querying  $\omega_t = \langle x_t, \kappa_t \rangle$ , the algorithm learns that  $\theta^*$  lies in one of the two halfspaces passing through  $\kappa_t$  with normal vector  $x_t$ , and updates the knowledge set by taking intersection with this halfspace. This ensures that the updated knowledge set still contains  $\theta^*$ . We use  $(\mathbf{h}, \omega)$ ,  $\mathbf{H}^+(\mathbf{h}, \omega)$ , and  $\mathbf{H}^-(\mathbf{h}, \omega)$  to denote the hyperplane with normal vector  $\mathbf{h} \in \mathbb{R}^d$  and intercept  $\omega$ , and the positive and negative halfspaces it creates with intercept  $\omega$ , i.e.,  $\{\mathbf{x} \in \mathbb{R}^d : \langle \mathbf{h}, \mathbf{x} \rangle \geq \omega\}$  and  $\{\mathbf{x} \in \mathbb{R}^d : \langle \mathbf{h}, \mathbf{x} \rangle \leq \omega\}$ , respectively. By properties of  $\kappa_t$ , the volume of the updated knowledge set is a constant factor of the initial volume, leading to geometric volume progress. For technical reasons, PROJECTEDVOLUME keeps a set  $S_t$  of dimensions with small width and works with the so-called *cylindrification*  $\text{Cyl}(\mathcal{K}_t, S_t)$  rather than the knowledge set  $\mathcal{K}_t$ . Although we do the same to build on their analysis in a black-box manner, the distinction between  $\mathcal{K}_t$  and  $\text{Cyl}(\mathcal{K}_t, S_t)$  is not important for understanding our algorithmic ideas and the corresponding definitions are deferred

to Chapter 7.3.3.

Having described PROJECTEDVOLUME that works when there are no corruptions, we turn to our algorithm. In the presence of even a few corruptions, PROJECTEDVOLUME may quickly eliminate  $\theta^*$  from  $\mathcal{K}_t$  (see Appendix D.1.2 for such an attack). To deal with this issue, we run the algorithm in epochs consisting of multiple queries. At each epoch, we combine all its queries to compute a hyperplane cut that both preserves  $\theta^*$  in the knowledge set and also makes enough volumetric progress on the latter's size. We face three important design decisions discussed separately below: what occurs inside an epoch, when to stop an epoch, and how to initialize the next one.

**What occurs within an epoch?** CORPV.KNOWN (Algorithm 7.1) formalizes what happens within an epoch  $\phi$ . The knowledge set is updated only at its end; this means that all rounds  $t$  in epoch  $\phi$  have the same knowledge set  $\mathcal{K}_\phi$  (and hence, the same centroid  $\kappa_\phi$ ). If the width of the knowledge set in the direction of  $x_t$  is smaller than  $\varepsilon$ , then, as in PROJECTEDVOLUME, we make an *exploit* query precisely described in Chapter 7.3.3. Otherwise, we make an *explore* query  $\omega_t = \langle x_t, \kappa_\phi \rangle$ , described below. The epoch keeps track of all explore queries that occur within its duration in a set  $\mathcal{A}_\phi$ . When it ends ( $\phi' = \phi + 1$ ), the knowledge set  $\mathcal{K}_{\phi+1}$  of the new epoch is initialized. In this subsection,  $\text{Cyl}(\mathcal{K}_\phi, S_\phi)$  can be thought as the knowledge set  $\mathcal{K}_\phi$  and the sets  $S_\phi$  and  $L_\phi$  can be ignored; these quantities are needed for technical reasons and are discussed in Chapter 7.3.3.

---

#### ALGORITHM 7.1: CORRUPTEDPROJECTEDVOLUME-KNOWN (CORPV.KNOWN)

---

- 1 **Global parameters:** Budget  $\bar{c}$ , accuracy  $\varepsilon$
  - 2 Initialize  $\phi = 1$ ,  $\mathcal{K}_\phi \leftarrow \mathcal{K}$ ,  $S_\phi \leftarrow \emptyset$ ,  $\kappa_\phi \leftarrow \text{apx-centr}(\text{Cyl}(\mathcal{K}_\phi, S_\phi))$ ,  $L_\phi \leftarrow \text{orth-basis}(\mathbb{R}^d)$ ,  $\mathcal{A}_\phi \leftarrow \emptyset$
  - 3 **for**  $t \in [T]$  **do**
  - 4     Observe context  $x_t$  and set  $\phi' \leftarrow \phi$
  - 5     **if**  $w(\text{Cyl}(\mathcal{K}_\phi, S_\phi), x_t) \leq \varepsilon$  **or**  $L_\phi = \emptyset$  **then** query point  $\omega_t = \text{CORPV.EXPLOIT}(x_t, \mathcal{K}_\phi)$
  - 6     **else**  $(\phi', \mathcal{A}_\phi) \leftarrow \text{CORPV.EXPLORE}(x_t, \phi, \kappa_\phi, L_\phi, \mathcal{A}_\phi)$
  - 7     **if**  $\phi' = \phi + 1$  **then** *// epoch changed in CORPV.EXPLORE*
  - 8         Compute separating cut:  $(\tilde{\mathbf{h}}, \tilde{\omega}) \leftarrow \text{CORPV.SEPARATINGCUT}(\kappa_\phi, S_\phi, L_\phi, \mathcal{A}_\phi)$
  - 9         Make updates  $(\mathcal{K}_{\phi+1}, S_{\phi+1}, L_{\phi+1}) \leftarrow \text{CORPV.EPOCHUPDATES}(\mathcal{K}_\phi, S_\phi, L_\phi, \tilde{\mathbf{h}}, \tilde{\omega})$
  - 10        Initialize next epoch:  $\phi \leftarrow \phi'$ ,  $\kappa_\phi \leftarrow \text{apx-centroid}(\text{Cyl}(\mathcal{K}_\phi, S_\phi))$ , and  $\mathcal{A}_\phi \leftarrow \emptyset$ .
- 

CORPV.EXPLORE (Algorithm 7.2) describes how we handle an explore query. When  $\bar{c} = 0$ , we can eliminate the halfspace that lies in the opposite direction of the feedback  $y_t$  after each explore query. However, when  $\bar{c} > 0$ , this may eliminate  $\theta^*$ . Instead, we keep all explore queries that

occurred in epoch  $\phi$  as well as the halfspace consistent with the observed feedback in  $\mathcal{A}_\phi$  and wait until we have enough data to identify a halfspace of the knowledge set that includes  $\theta^*$  and makes sufficient volumetric progress; we refer to this as a *separating cut*. We then move to epoch  $\phi' = \phi + 1$ .

---

**ALGORITHM 7.2: CorPV.EXPLORE**


---

- 1 **Parameters:**  $x_t, \phi, \kappa_\phi, L_\phi, \mathcal{A}_\phi$
  - 2 Select query point  $\omega_t = \langle x_t, \kappa_\phi \rangle$  and observe feedback  $y_t$ .
  - 3 Update set  $\mathcal{A}_\phi$ : **if**  $y_t = +1$ :  $\mathcal{A}_\phi \leftarrow \mathcal{A}_\phi \cup \mathbf{H}^+(\Pi_{L_\phi} x_t, \omega_t)$  **else**  $\mathcal{A}_\phi \leftarrow \mathcal{A}_\phi \cup \mathbf{H}^-(\Pi_{L_\phi} x_t, \omega_t)$ .
  - 4 **if**  $|\mathcal{A}_\phi| \geq \tau$  **then** Move to next epoch  $\phi' \leftarrow \phi + 1$ . //  $\tau := 2d \cdot \bar{c} \cdot (d + 1) + 1$
  - 5 **else** Stay in the same epoch  $\phi' \leftarrow \phi$ .
  - 6 **return**  $(\phi', \mathcal{A}_\phi)$
- 

**When does the epoch end?** We next explain how many queries are enough to guarantee that such a separating cut exists. To guarantee that  $\theta^*$  is preserved after the cut, we need to make sure that we only eliminate the halfspace where candidate parameters  $\theta$  are misclassified (i.e., are inconsistent with the agent's response) by at least  $\bar{c} + 1$  explore queries. Note that because there are at most  $\bar{c}$  corruptions,  $\theta^*$  can be misclassified by at most  $\bar{c}$  queries. In other words, if the set of all candidate parameters  $\theta$  that are misclassified by at most  $\bar{c}$  explore queries are on the non-eliminated halfspace of the hyperplane, then this hyperplane can serve as a separating cut. We refer to the set of these parameters as the *protected region* as we aim to ensure that they are not eliminated.

At first glance, one might think that, after  $2\bar{c} + 1$  explore queries, we can directly use one of them as a separating cut. Interestingly, although this is the case for  $d = 2$ , we show that for  $d = 3$ , if we are restricted to separating cuts on the direction of existing explore queries, even arbitrarily many such queries do not suffice (see Chapter 7.4.3). One key technical component in our analysis is to show that when we combine  $\tau = 2d(d + 1)\bar{c} + 1$  explore queries, there exists a hyperplane that separates the convex hull of the protected region from a point  $p^*$  that is close to  $\kappa_\phi$  but also outside of that convex hull (separating hyperplane theorem). Since that hyperplane crosses close to  $\kappa_\phi$ , we ensure enough volumetric progress. Since the non-eliminated halfspace includes all the parameters in the protected region, we ensure that we do not eliminate  $\theta^*$ . The proof of this argument relies on the Carathéodory theorem and is informally sketched in the left figure of Figure 7.1.

**How do we initialize the next epoch?** Since the existence of the separating cut is established, if we were able to compute this cut, we would be able to compute the knowledge set of the next epoch by

taking its intersection with the positive halfspace of the cut. However, the separating hyperplane theorem provides only an existential argument and no direct way to compute the separating cut. To deal with this, recall that the separating cut should have  $\mathbf{p}^*$  on its negative halfspace and the whole protected region in its positive halfspace. To compute it, we use Perceptron ([Rosenblatt, 1958](#)), which is typically used to provide a linear classifier for a set of (positive and negative) points in the *realizable* setting (i.e., when there exists a hyperplane that correctly classifies these points). Perceptron proceeds by iterating across the points and suggesting a classifier. Every time that a point is misclassified, Perceptron makes an update. If the entire protected region is classified as positive and  $\mathbf{p}^*$  as negative by Perceptron, then we return its hyperplane as the separating cut; otherwise we feed one point that violates the intended labeling to Perceptron. Perceptron makes a mistake and updates its classifier. The main guarantee of Perceptron is that, if there exists a classifier with margin of  $\gamma > 0$  (i.e., smallest distance to any data point is  $\gamma$ ), the number of mistakes that Perceptron makes is at most  $1/\gamma^2$  (precisely, the bound is in Lemma [D.16](#)).

The problem is that we do not know  $\mathbf{p}^*$  and, even if we deal with this,  $\mathbf{p}^*$  does not necessarily have a large enough margin from the protected region. To overcome this, we use a sampling process that with big enough probability identifies a different point  $\mathbf{q}$ , *in the vicinity* of  $\mathbf{p}^*$ , whose margin to the protected region is lower bounded by  $\gamma$ . If  $\mathbf{q}$  does have the desired margin, the mistake bound of Perceptron controls the running time needed to identify the separating hyperplane. Otherwise, we proceed with a new random point. This takes care of the small margin issue with  $\mathbf{p}^*$ .

In order to pin down  $\mathbf{p}^*$ , we construct a set of points  $\Lambda_\phi$ , which we call *landmarks*, such that at least one of them is outside of the convex hull of the protected region. We run multiple versions of Perceptron, each with a random  $\mathbf{p}^* \in \Lambda_\phi$  and a point  $\mathbf{q}$  randomly selected in a ball around  $\mathbf{p}^*$  of an appropriately defined radius  $\zeta$ , which we denote by  $\mathcal{B}_{L_\phi}(\mathbf{p}^*, \zeta)$ ; this can be computed efficiently by normalizing  $\mathcal{B}_{L_\phi}(\mathbf{p}^*, \zeta)$  to a unit ball and using the techniques presented in ([Blum et al., 2016](#), Section 2.5). If  $\mathbf{q}$  has a big-enough margin then the mistake bound of Perceptron ensures that **CorPV.SEPARATINGCUT** (Algorithm [7.3](#)) returns the separating cut. Volume cap arguments show that point  $\mathbf{q}$  has the required margin with big enough probability (informally sketched in the right figure of Figure [7.1](#)), which bounds the number of the outer *while* loops and thereby the running time.



Figure 7.1: Informal sketch of the Carathéodory's theorem in two dimensions (left) and the computation of a separating cut (right). The ellipsoid corresponds to the current knowledge set  $\mathcal{K}$ . The blue area corresponds to the  $\bar{c}$ -protected region. The patterned area corresponds to its convex hull. On the left, any point  $p$  of the convex hull can be written as a convex combination of points  $\hat{p}, \tilde{p}$  from the protected region. Thus, all points inside the convex hull of the protected region (horizontal-line pattern) have been misclassified *at most*  $(d + 1)\bar{c}$  times. On the right, point  $p^*$  has been misclassified *at least*  $2d(d + 1)\bar{c} + 1$ . Sampling points from the spherical cap of the magenta ball (vertical-line pattern) gives a big enough margin for Perceptron. The black line corresponds to the valid separating cut.

---

**ALGORITHM 7.3: CorPV.SEPARATINGCUT**


---

```

1 Parameters:  $\kappa_\phi, S_\phi, L_\phi, \mathcal{A}_\phi$  // size of small dimensions  $\delta := \frac{\varepsilon}{4(d+\sqrt{d})}$ 
2 Fix landmarks  $\Lambda_\phi = \{\kappa_\phi \pm \bar{v} \cdot e_i, \forall e_i \in E_\phi\}$  where  $E_\phi \leftarrow \text{orth-basis}(L_\phi)$  and  $\bar{v} = \frac{\varepsilon - 2\sqrt{d}\delta}{4\sqrt{d}}$ 
3 Let  $w := (\tilde{h}, \tilde{\omega}) \in \mathbb{R}^{d+1}$ . // Perceptron hyperplane
4 while true do
5   Initialize  $w = [1]^d$  and mistake counter to  $M \leftarrow 0$ . // Perceptron initialization
6   Sample a random point  $q$  from ball  $B_{L_\phi}(p^*, \zeta)$  with radius  $\zeta = \bar{v}$  around random  $p^* \in \Lambda_\phi$ .
7   while  $M < \frac{d-1}{\zeta^2 \cdot \ln^2(3/2)}$  do // Perceptron mistake bound
8     Set  $m \leftarrow 0$ .
9     if  $q \in H^+(\tilde{h}, \tilde{\omega})$  then  $w \leftarrow w - q$ ; set  $m \leftarrow m + 1$ . // Perceptron update
10    if  $\kappa_\phi \in H^-(\tilde{h}, \tilde{\omega})$  then  $w \leftarrow w + q$ ; set  $m \leftarrow m + 1$ . // Perceptron update
11    for subsets  $D_\phi \subseteq \mathcal{A}_\phi$  such that  $|D_\phi| = \bar{c}$  do
12      Let  $P$  be the polytope created by halfspaces of  $\mathcal{A}_\phi \setminus D_\phi$  and  $H^-(\tilde{h}, \tilde{\omega})$ .
13      if  $P \neq \emptyset$  then  $w \leftarrow w + z$  for  $z \in P$ ; set  $m \leftarrow m + 1$ . // Perceptron update
14    if  $m \neq 0$  then increase mistake counter  $M \leftarrow M + m$ .
15  else return  $(\tilde{h}, \tilde{\omega})$ 

```

---

We discuss next the computational complexity of our algorithm. As written in lines 11-13 of Algorithm 7.3, checking whether the protected region is contained in the positive halfspace of the Perceptron hyperplane requires going over all  $(\frac{|\mathcal{A}_\phi|}{\bar{c}})$  ways to remove  $\bar{c}$  hyperplanes and checking whether the resulting region intersects the negative halfspace (if this happens, then points with misclassification of at most  $\bar{c}$  may be misclassified). This suggests a running time that is exponential in  $\bar{c}$ . Fortunately, as detailed in Chapter 7.3.2, to handle the unknown corruption or the other intricacies in our actual behavioral model beyond the  $\bar{c}$ -known-corruption, we only run this algorithm with  $\bar{c} \approx \log(T)$ . So the final running time of our algorithms is quasi-polynomial in  $T$ .

### 7.3.2 ADAPTING TO AN UNKNOWN CORRUPTION LEVEL

We now provide the algorithm when the corruption level  $C$  is unknown (Algorithm 7.4). The places where CorPV.AC differs from CorPV.KNOWN are in lines 4, 7-8, 14-18. This section extends ideas from [Lykouris et al. \(2018\)](#) for multi-armed bandits to contextual search which poses an additional difficulty as the search space is continuous. This is not as straightforward as a doubling trick for the unknown  $C$ , as both the loss and the corruption  $c_t$  are unobservable; doubling tricks require identifying a proxy for the quantity under question and doubling once a threshold is reached.

The basic idea is to maintain multiple copies of CorPV.KNOWN, which we refer to as *layers*. At every round, we decide which copy to play probabilistically. Each copy  $j$  keeps its own environment with its corresponding epoch  $\phi(j)$  and knowledge set  $\mathcal{K}_{j,\phi(j)}$ . Smaller values  $j$  for the copies are *less robust* to corruption and we impose a monotonicity property among them by ensuring that the knowledge sets are nested, i.e.,  $\mathcal{K}_{j,\phi(j)} \subseteq \mathcal{K}_{j',\phi(j')}$  for  $j \leq j'$ . This allows more robust layers to correct mistakes of less robust layers that may inadvertently eliminate  $\theta^*$  from their knowledge set.

More formally, we run  $\log T$  parallel versions of the  $\bar{c}$ -known-corruption algorithm with a corruption level of  $\bar{c} \approx \log(T)$ . At the beginning of each round  $t$ , the algorithm randomly selects layer  $j$  with probability  $2^{-j}$  (line 4) and executes the layer's algorithm for this round. Since the adversary does not know the randomness in the algorithm, this makes layers  $j$  with  $C \leq 2^j$  robust to corruption level of  $C$ . The reason is that the expected number of corruptions occurring at layer  $j$  is at most 1 and, with high probability, less than  $\log T$  which is accounted by the  $\bar{c} = \log T$  upper bound on corruption based on which we run CorPV.KNOWN on this layer.

However, there is a problem: all layers with  $C > 2^j$  are not robust to corruption of  $C$  so they may eliminate  $\theta^*$  and, to make things worse, the algorithm follows the recommendation of these layers with large probability. As a result, we need a way to supervise their decisions by more robust layers. To achieve that, we use nested active sets; when the layer  $j_t$  selected at round  $t$  proceeds with a separating cut on its knowledge set, we also make the same cut on all less robust layers  $j' < j_t$  (lines 14-15). This allows non-robust layers that have eliminated  $\theta^*$  from their knowledge set to correct their mistakes by removing the incorrect parameters of their version space that they had converged to from their knowledge sets. There are two additional points that arise in the contextual

---

**ALGORITHM 7.4: CorPV.AC (Adversarial Corruption version)**


---

```

1 Global parameters: Failure probability  $\beta$ , budget  $\bar{c} := 2 \log(T/\beta)$ , accuracy  $\varepsilon$ 
2 Initialize layer-specific quantities for all layers  $j \in [\log T]$ :  $\phi(j) = 1$ ,  $\mathcal{K}_{j,\phi(j)} \leftarrow \mathcal{K}$ ,  $S_{j,\phi(j)} \leftarrow \emptyset$ ,
    $\kappa_{j,\phi(j)} \leftarrow \text{apx-centroid}(\text{Cyl}(\mathcal{K}_{j,\phi(j)}, S_{j,\phi(j)}))$ ,  $L_{j,\phi(j)} \leftarrow \text{orthonorm-basis}(\mathbb{R}^d)$ ,  $\mathcal{A}_{j,\phi(j)} \leftarrow \emptyset$ 
3 for  $t \in [T]$  do
4   Sample layer  $j_t \in [\log T]$ :  $j_t = j$  with probability  $2^{-j}$ ; with remaining probability,  $j_t = 1$ .
5   Observe context  $\mathbf{x}_t$  and set  $\phi' \leftarrow \phi(j_t)$ .
6   if  $w(\text{Cyl}(\mathcal{K}_{j_t,\phi(j_t)}, S_{j_t,\phi(j_t)})) \leq \varepsilon$  or  $L_{j_t,\phi(j_t)} \neq \emptyset$  then
7     Find smallest more robust exploit layer  $j \geq j_t$ :  $j = \min_{j' \geq j_t} w(\mathcal{K}_{j',\phi(j')}, \mathbf{x}_t) \leq \varepsilon$ .
8     Compute exploit query point for this layer:  $\omega_t = \text{CorPV.EXPLOIT}(\mathbf{x}_t, \mathcal{K}_{j,\phi(j)})$ .
9   else  $(\phi', \mathbf{p}^*, \mathcal{A}_{j_t,\phi(j_t)}) \leftarrow \text{CorPV.EXPLORE}(\mathbf{x}_t, \phi(j_t), \kappa_{j_t,\phi(j_t)}, L_{j_t,\phi(j_t)}, \mathcal{A}_{j_t,\phi(j_t)})$ 
10  if  $\phi' = \phi(j_t) + 1$  then                                // epoch changed in CorPV.EXPLORE
11     $(\tilde{\mathbf{h}}, \tilde{\omega}) \leftarrow \text{CorPV.SEPARATINGCUT}(\kappa_{j_t,\phi(j_t)}, S_{j_t,\phi(j_t)}, L_{j_t,\phi(j_t)}, \mathcal{A}_{j_t,\phi(j_t)})$ 
12     $(\mathcal{K}_{j_t,\phi'}, S_{j_t,\phi'}, L_{j_t,\phi'}) \leftarrow \text{CorPV.EPOCHUPDATES}(\mathcal{K}_{j_t,\phi(j_t)}, S_{j_t,\phi(j_t)}, L_{j_t,\phi(j_t)}, \tilde{\mathbf{h}}, \tilde{\omega})$ 
13     $\phi(j_t) \leftarrow \phi'$ ,  $\kappa_{j_t,\phi(j_t)} \leftarrow \text{apx-centroid}(\text{Cyl}(\mathcal{K}_{j_t,\phi(j_t)}, S_{j_t,\phi(j_t)}))$ , and  $\mathcal{A}_{j_t,\phi(j_t)} \leftarrow \emptyset$ .
14  for  $j' \leq j_t$  do                                // Make less robust layers consistent with  $j_t$ 
15     $(\mathcal{K}_{j',\phi(j')}, S', L') \leftarrow \text{CorPV.EPOCHUPDATES}(\mathcal{K}_{j',\phi(j')}, S_{j',\phi(j')}, L_{j',\phi(j')}, \tilde{\mathbf{h}}, \tilde{\omega})$ 
16    if  $\kappa_{j',\phi(j')} \notin \mathcal{K}_{j',\phi(j')}$  or  $S' \neq S_{j',\phi(j')}$  then
17       $\phi(j') \leftarrow \phi(j') + 1$ ,  $(S_{j',\phi(j')+1}, L_{j',\phi(j')+1}) \leftarrow (S', L')$ ,  $\mathcal{A}_{j',\phi(j')} \leftarrow \emptyset$ 
18       $\kappa_{j',\phi(j')} \leftarrow \text{apx-centroid}(\text{Cyl}(\mathcal{K}_{j',\phi(j')}, S_{j',\phi(j')}))$ .

```

---

search setting. First, the aforementioned cut may not make enough volumetric progress in the knowledge sets of layers  $j' < j_t$ . As a result, as described in lines 16-18, we only move to the next epoch for layer  $j'$  if its centroid is removed from the knowledge set or another change discussed in Chapter 7.3.3 is triggered. Second, with respect to exploit queries, we want to make sure that we do not keep confidence on non-robust layers. As a result, we follow the exploit recommendation of the largest layer  $j \geq j_t$  that has converged to exploit recommendation in this direction, i.e.,  $w(\mathcal{K}_{j,\phi(j)}) \leq \varepsilon$  (lines 7-8). This eventually allows us to bound the regret from all non-robust layers

by the smallest robust layer  $\lceil \log C \rceil$  (see Chapter 7.4.2).

### 7.3.3 REMAINING COMPONENTS OF THE ALGORITHM.

The presentation of the algorithm until this point has disregarded some technical parts. We now discuss each of them so that the algorithm is fully defined.

**Cylindrification, small, and large dimensions.** To facilitate relating the volume progress to a bound on the explore queries, similar to [Lobel et al. \(2018\)](#), we keep two sets of vectors/dimensions  $S_\phi$  and  $L_\phi$  whose union creates an orthonormal basis. The set  $S_\phi$  has *small dimensions*  $\mathbf{s} \in S_\phi$  with width  $w(\mathcal{K}_\phi, \mathbf{s}) \leq \delta$  for  $\delta := \frac{\varepsilon}{4(d+\sqrt{d})}$ . The set  $L_\phi$  is any basis for the subspace orthogonal to  $S_\phi$ , with the property that  $\forall \mathbf{l} \in L_\phi : w(\mathcal{K}_\phi, \mathbf{l}) > \delta$ . The set  $L_\phi$  completes an orthonormal basis maintaining that  $\mathbf{l} \in L_\phi : w(\mathcal{K}_\phi, \mathbf{l}) > \delta$ . When an epoch ends, sets  $S_\phi$  and  $L_\phi$  are updated together with the knowledge set  $\mathcal{K}_\phi$  as described in `CORPV.EPOCHUPDATES` (Algorithm 7.5): if the new direction  $\tilde{\mathbf{h}}$  of the separating cut projected to the large dimensions has width  $w(\Pi_{L_\phi} \mathcal{K}_{\phi+1}, \tilde{\mathbf{h}}) \leq \delta$ , we add it to  $S_{\phi+1}$  and we update  $L_{\phi+1}$  to keep the invariant that no large dimension has width larger than  $\delta$ .

---

#### ALGORITHM 7.5: CORPV.EPOCHUPDATES

---

- 1 **Parameters:**  $\mathcal{K}_\phi, S_\phi, L_\phi, \tilde{\mathbf{h}}, \tilde{\omega}$
  - 2 Update  $\mathcal{K}_{\phi+1} \leftarrow \mathcal{K}_\phi \cap \mathbf{H}^+(\tilde{\mathbf{h}}, \tilde{\omega})$  and save temporary sets  $S' \leftarrow S_\phi$  and  $L \leftarrow L_\phi$ .
  - 3 **if**  $w(\Pi_{L_\phi} \mathcal{K}_{\phi+1}, \tilde{\mathbf{h}}) \leq \delta$  **then** *// size of small dimensions*  $\delta := \frac{\varepsilon}{4(d+\sqrt{d})}$
  - 4     Add hyperplane to small dimensions  $S' \leftarrow S_\phi \cup \{\tilde{\mathbf{h}}\}$ .
  - 5     Compute orthonormal basis for new large dimensions  $L$  (without  $S'$ ).
  - 6     Update  $L_{\phi+1} \leftarrow L \setminus \{e_i \in L : w(\mathcal{K}_{\phi+1}, e_i) \leq \delta\}$  and  $S_{\phi+1} \leftarrow S' \cup (L \setminus L_{\phi+1})$ .
  - 7 **return**  $(\mathcal{K}_{\phi+1}, S_{\phi+1}, L_{\phi+1})$
- 

Overall, the potential function we use to make sure that we make progress depends on the projected volume of the knowledge set on the large dimensions  $L_\phi$ , as well as the number of small dimensions  $S_\phi$ . This is why in lines 7-8 of Algorithm 7.4, we update the epoch of less robust layers when one of these two measures of progress is triggered. Sets  $S_\phi$  and  $L_\phi$  serve in explaining which dimensions are identified well enough so that we can focus our attention on making progress in the remaining dimensions. For this to happen, an important notion is that of *Cylindrification* which creates a box covering the knowledge set and removes the significance of the small dimensions.

**Definition 7.1** (Cylindrification, Definition 4.1 of [Lobel et al. \(2018\)](#)). *Given a set of orthonormal vectors  $S = \{\mathbf{s}_1, \dots, \mathbf{s}_n\}$ , let  $L = \{\mathbf{u} | \langle \mathbf{u}, \mathbf{s} \rangle = 0; \forall \mathbf{s} \in S\}$  be a subspace orthogonal to  $\text{span}(S)$  and  $\Pi_L \mathcal{K}$  be the projection of convex set  $\mathcal{K} \subseteq \mathbb{R}^d$  onto  $L$ . We define:*

$$\text{Cyl}(\mathcal{K}, S) := \left\{ \mathbf{z} + \sum_{i=1}^n b_i \mathbf{s}_i \mid \mathbf{z} \in \Pi_L \mathcal{K} \text{ and } \min_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, \mathbf{s}_i \rangle \leq b_i \leq \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, \mathbf{s}_i \rangle \right\}.$$

By working with the Cylindrification  $\text{Cyl}(\mathcal{K}_\phi, S_\phi)$  (Definition 7.1) rather than the original set of small dimensions  $S_\phi$ , we can ensure that we make queries that make volumetric progress with respect to the large dimensions, that have been less well understood. This is the reason why the landmark  $\mathbf{p}^*$  we identify lives in the large dimensions while being close to the centroid  $\boldsymbol{\kappa}_\phi$  (line 2).

**Exploit queries for different loss functions.** When the width of the knowledge set on the direction of the incoming context is small, i.e.,  $w(\mathcal{K}_\phi, \mathbf{x}_t) \leq \varepsilon$ , we proceed with an exploit query. This module evaluates the loss of each query  $\omega$  with respect to any parameter that is consistent with the knowledge set, i.e.,  $\boldsymbol{\theta}^* \in \mathcal{K}_\phi$ . It then employs a min-max approach by selecting the query  $\omega_t$  that has the minimum loss for the worst-case selection of  $\boldsymbol{\theta} \in \mathcal{K}_\phi$ . For the  $\varepsilon$ -ball loss, any query point  $\omega_t = \langle \mathbf{x}_t, \boldsymbol{\theta}' \rangle$  with  $\boldsymbol{\theta}' \in \mathcal{K}_\phi$  results in loss equal to 0; this is what PROJECTEDVOLUME also does to achieve optimal regret for the  $\varepsilon$ -ball loss function.

---

#### **ALGORITHM 7.6: CorPV.EXPLOIT**

---

- 1 **Parameters:**  $\mathbf{x}_t, \mathcal{K}_\phi$
  - 2 Compute query point  $\omega_t = \min_{\omega \in \Omega} \max_{\boldsymbol{\theta} \in \mathcal{K}_\phi} \ell(\omega, \langle \boldsymbol{\theta}, \mathbf{x}_t \rangle, \langle \boldsymbol{\theta}, \mathbf{x}_t \rangle)$
  - 3 **return**  $\omega_t$
- 

Moving to the pricing loss and assuming that the query point is  $\omega_t = \langle \mathbf{x}_t, \boldsymbol{\theta} \rangle$  for some  $\boldsymbol{\theta} \in \mathcal{K}_\phi$ , although the distance of  $\boldsymbol{\theta}^*$  to hyperplane  $(\mathbf{x}_t, \omega_t)$  is less than  $\varepsilon$ , there is a big difference based on which side of the hyperplane  $\boldsymbol{\theta}^*$  lies in (i.e., whether  $\boldsymbol{\theta}^* \in \mathbf{H}^+(\mathbf{x}_t, \omega_t)$  or not). Specifically, if  $\omega_t > \langle \mathbf{x}_t, \boldsymbol{\theta}_t^* \rangle$  then a fully rational agent does not buy and we get zero revenue, thereby incurring a loss of  $\langle \mathbf{x}_t, \boldsymbol{\theta}^* \rangle$ . On the other hand, querying  $\omega^* = \langle \mathbf{x}_t, \boldsymbol{\theta}^* \rangle$  would lead to a purchase from a fully rational agent, and hence, to a pricing loss of 0. As we discuss in Chapter 7.6, this discontinuity in pricing loss poses further complications in extending other algorithms to contextual search.

To deal with this discontinuity, we can query point  $\omega_t$  with  $\omega_t = \langle \mathbf{x}_t, \boldsymbol{\theta}^* \rangle - \varepsilon$ , as the value of

the fully rational agent is certainly above this price. In fact, when dealing with boundedly rational agents (Section 7.5), such a lower price is essential even if we know  $\theta^*$  in order to account for the noise and there the definition of  $\omega_t$  accounts for the distributional information about the noise.

## 7.4 ANALYSIS

In this section, we provide the analysis of the algorithm introduced in Chapter 7.3. We first analyze the result for the intermediate  $\bar{c}$ -known corruption setting. This setting allows us to introduce our key additional ideas and serves as a building block to extend to both the setting where  $C$  is unknown (Theorem 7.1) as well as the bounded rationality behavioral model (Theorem 7.2).

### 7.4.1 EXISTENCE OF A SEPARATING HYPERPLANE AT THE END OF ANY EPOCH

We first show, in Lemma 7.1, that after  $\tau = 2d \cdot \bar{c}(d + 1) + 1$  rounds, there exist  $\mathbf{h}_\phi^* \in \mathbb{R}^d$  and  $\omega_\phi^* \in \mathbb{R}$  such that the hyperplane  $(\mathbf{h}_\phi^*, \omega_\phi^*)$  is a separating cut, i.e., it passes close to the approximate centroid  $\kappa_\phi$  (and therefore also to the centroid  $\kappa_\phi^*$ ), and has in the entirety of one of its halfspaces only parameters “misclassified” at least  $\bar{c} + 1$  explore times. The results of this subsection hold for *any* scalar  $\delta < \frac{\varepsilon}{2\sqrt{d+4d}}$ . For the analysis, we make three simplifications (all without loss of generality) in an effort to ease the notation. First, we assume that  $y_t = +1, \forall t \in [T]$ . This is indeed without loss of generality since the algorithm can always negate the received context  $x_t$  and the chosen query  $\omega_t$  to force  $y_t = +1$  (Step 3 of Algorithm 7.2). Second, for rounds where nature’s answer is arbitrary, we assume that the perceived value is  $\tilde{v}_t = \langle \mathbf{x}_t, \boldsymbol{\theta}_t \rangle$ , where  $\boldsymbol{\theta}_t \in \mathcal{K}_0$  and it can change from round to round. For all other rounds  $\boldsymbol{\theta}_t = \boldsymbol{\theta}^*$ . Third, we assume that all hyperplanes have unit  $\ell_2$  norm.

**Lemma 7.1.** *For any epoch  $\phi$ , scalars  $\delta \in \left(0, \frac{\varepsilon}{2\sqrt{d+4d}}\right)$  and  $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\delta}{4\sqrt{d}}$ , after  $\tau = 2d \cdot \bar{c}(d + 1) + 1$  rounds, there exists a hyperplane  $(\mathbf{h}_\phi^*, \omega_\phi^*)$  orthogonal to small dimensions  $S_\phi$ , such that halfspace  $\mathbf{H}^+(\mathbf{h}_\phi^*, \omega_\phi^*)$  always contains  $\boldsymbol{\theta}^*$  and  $\text{dist}(\kappa_\phi, (\mathbf{h}_\phi^*, \omega_\phi^*)) \leq \bar{\nu}$ , where by  $\text{dist}(\kappa, (\mathbf{h}, \omega))$  we denote the distance of point  $\kappa$  from hyperplane  $(\mathbf{h}, \omega)$ , i.e.,  $\text{dist}(\kappa, (\mathbf{h}, \omega)) = \frac{|\langle \kappa, \mathbf{h} \rangle - \omega|}{\|\mathbf{h}\|}$ .*

At a high level, the tuning of  $\bar{\nu}$  depends on two factors. First, in order to make sure that we make enough progress in terms of volume elimination, despite the fact that we do not make a cut through

$\kappa_\phi$ , we need  $\bar{\nu}$  to be close enough to  $\kappa_\phi$  (Lemma D.11). Second, we need to guarantee that there exists at least one point with a very high undesirability level (Lemma D.6). For the analysis, we define the  $\nu$ -margin projected undesirability levels, which we later use for some fixed  $\nu < \bar{\nu}$ :

**Definition 7.2** ( $\nu$ -Margin Projected Undesirability Level). *Consider an epoch  $\phi$ , a scalar  $\nu$ , and a point  $\mathbf{p}$  in  $\mathcal{K}_\phi$ . Given the set  $\mathcal{A}_\phi = \{(\Pi_{L_\phi} \mathbf{x}_t, \omega_t)\}_{t \in [\tau]}$ , we define  $\mathbf{p}$ 's  $\nu$ -margin projected undesirability level, denoted by  $u_\phi(\mathbf{p}, \nu)$ , as the number of rounds within epoch  $\phi$ , for which*

$$u_\phi(\mathbf{p}, \nu) = \sum_{t \in [\tau]} \mathbb{1} \{ (\langle \mathbf{p} - \kappa_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) < 0 \}.$$

Intuitively,  $u_\phi(\mathbf{p}, \nu)$  gives penalty to a point  $\mathbf{p}$  if it is far (more than  $\nu$ ) from the negative halfspace of the query (when projected to the large dimensions  $L_\phi$ ). We can then show (Lemma D.4) that the undesirability level of a point  $\mathbf{p}$  during an epoch  $\phi$  corresponds to the number of times during epoch  $\phi$  that  $\mathbf{p}$  and  $\theta_t$  were at opposite sides of hyperplane  $(\Pi_{L_\phi} \mathbf{x}_t, \nu + \omega_t)$  for any  $\nu > \bar{\nu}$ .

Armed with this, we define the  $\bar{c}$ -protected region in large dimensions,  $\mathcal{P}(\bar{c}, \nu)$ , which is the set of points in  $\mathcal{K}_\phi$  with  $\nu$ -margin projected undesirability level at most  $\bar{c}$ . Formally:

$$\mathcal{P}(\bar{c}, \nu) = \{ \mathbf{p} \in \mathcal{K}_\phi : u_\phi(\mathbf{p}, \nu) \leq \bar{c} \}$$

The next lemma establishes that if we keep set  $\mathcal{P}(\bar{c}, \nu)$  intact in the convex body formed for the next epoch  $\mathcal{K}_{\phi+1}$ , then we are guaranteed to not eliminate point  $\theta^*$  (proof in Appendix D.3.1).

**Lemma 7.2.** *If  $\nu > \underline{\nu}$  (where  $\underline{\nu} = \sqrt{d}\delta$ ), then the ground truth  $\theta^*$  is included in the set  $\mathcal{P}(\bar{c}, \nu)$ .*

We next show that there exists a hyperplane cut, that is orthogonal to all small dimensions in a way that guarantees that the set  $\mathcal{P}(\bar{c}, \nu)$  is *preserved* in  $\mathcal{K}_{\phi+1}$  (i.e.,  $\mathcal{P}(\bar{c}, \nu) \subseteq \mathcal{K}_{\phi+1}$ ). Note that due to Lemma 7.2, it is enough to guarantee that we have  $\theta^* \in \mathcal{K}_{\phi+1}$ . However,  $\mathcal{P}(\bar{c}, \nu)$  is generally non-convex and it is not easy to directly make claims about it. Instead, we focus on its convex hull, denoted by  $\text{conv}(\mathcal{P}(\bar{c}, \nu))$ ; for any point in  $\text{conv}(\mathcal{P}(\bar{c}, \nu))$  we can upper bound its undesirability by applying Carathéodory's Theorem, which says that any point in the convex hull of a (possibly non-convex) set can be written as a convex combination of at most  $d + 1$  points of that set. Using this result, we can bound the  $\nu$ -margin projected undesirability levels of all the points in  $\text{conv}(\mathcal{P}(\bar{c}, \nu))$ .

**Lemma 7.3.** For any scalar  $\nu$ , epoch  $\phi$  and any point  $\mathbf{p} \in \text{conv}(\mathcal{P}(\bar{c}, \nu))$ , its  $\nu$ -margin projected undesirability level is at most  $\bar{c} \cdot (d + 1)$ , i.e.,  $u_\phi(\mathbf{p}, \nu) \leq \bar{c} \cdot (d + 1)$ .

*Proof.* From Carathéodory's Theorem, since  $\mathbf{p} \in \mathbb{R}^d$  and is inside  $\text{conv}(\mathcal{P}(\bar{c}, \nu))$ , it can be written as the convex combination of *at most*  $d + 1$  points in  $\mathcal{P}(\bar{c}, \nu)$ . Denoting these points by  $\{\mathbf{z}_1, \dots, \mathbf{z}_{d+1}\}$  such that  $\mathbf{z}_i \in \mathcal{P}(\bar{c}, \nu), \forall i \in [d+1]$ ,  $\mathbf{p}$  can be written as  $\mathbf{p} = \sum_{i=1}^{d+1} a_i \mathbf{z}_i$  where  $a_i \geq 0, \forall i \in [d+1]$  and  $\sum_{i=1}^{d+1} a_i = 1$ . Hence, the  $\nu$ -margin projected undesirability level of  $\mathbf{p}$  in epoch  $\phi$  is:

$$\begin{aligned} u_\phi(\mathbf{p}, \nu) &= \sum_{t \in [\tau]} \mathbb{1} \{ (\langle \mathbf{p} - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) < 0 \} && \text{(Definition 7.2)} \\ &= \sum_{t \in [\tau]} \mathbb{1} \left\{ \sum_{i \in [d+1]} a_i \underbrace{(\langle \mathbf{z}_i - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu)}_{Q_i} < 0 \right\} && \text{(Carathéodory's Theorem)} \\ &\leq \sum_{t \in [\tau]} \sum_{i \in [d+1]} \mathbb{1} \{ (\langle \mathbf{z}_i - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) < 0 \} \\ &\leq \sum_{i \in [d+1]} u_\phi(\mathbf{z}_i, \nu) \leq \bar{c} \cdot (d + 1) && (\mathbf{z}_i \in \mathcal{P}(\bar{c}, \nu) \text{ and definition of } \mathcal{P}(\bar{c}, \nu)) \end{aligned}$$

where the first inequality comes from the fact that if  $Q_i \geq 0$  for all  $\mathbf{z}_i, i \in [d+1]$ , then the corresponding summand contributes 0 undesirability points to  $u_\phi(\mathbf{p}, \nu)$ , since  $a_i \geq 0$  as this is a convex combination. As a result, each undesirability point on the left hand side of the latter inequality can be attributed to at least one  $\mathbf{z}_i$  from the right hand side. ■

Next, we prove that there exists some point  $\mathbf{q} \in \mathcal{K}_\phi$  such that  $u_\phi(\mathbf{q}, \nu) \geq \bar{c} \cdot (d + 1) + 1$ . Note that by the previous lemma, we know that  $\mathbf{q} \notin \text{conv}(\mathcal{P}(\bar{c}, \nu))$ . As a result, *any* hyperplane separating  $\mathbf{q}$  from  $\text{conv}(\mathcal{P}(\bar{c}, \nu))$  preserves  $\mathcal{P}(\bar{c}, \nu)$  (and as a result,  $\boldsymbol{\theta}^*$ ) for  $\mathcal{K}_{\phi+1}$ . To make sure that we also make progress in terms of volume elimination, we show below that there exists a separating hyperplane in the space of large dimensions (i.e., orthogonal to all small dimensions). For our analysis, we introduce the notion of *landmarks*.

**Definition 7.3** (Landmarks). Let basis  $E_\phi = \{\mathbf{e}_1, \dots, \mathbf{e}_{d-|S_\phi|}\}$  be such that  $E_\phi$  is orthogonal to  $S_\phi$ , any scalar  $\delta \in (0, \frac{\varepsilon}{2\sqrt{d+4d}})$ , and a scalar  $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\delta}{4\sqrt{d}}$ . We define the  $2(d - |S_\phi|)$  landmarks to be the points such that  $\Lambda_\phi = \{\boldsymbol{\kappa}_\phi \pm \bar{\nu} \cdot \mathbf{e}_i, \forall \mathbf{e}_i \in E_\phi\}$ .

Landmarks possess the convenient property that at every round where the observed context  $x_t$  was such that  $w(\mathcal{K}_\phi, x_t) \geq \varepsilon$ , at least one of them gets a  $\nu$ -margin projected undesirability point, when  $\nu < \bar{\nu}$  (Lemma D.6). The tuning of  $\bar{\nu}$  explains the constraint imposed on  $\delta$ , i.e.,  $\delta < \frac{\varepsilon}{2\sqrt{d} + 4\delta}$ . This constraint is due to the fact that since  $\nu > \underline{\nu}$  and  $\nu < \bar{\nu}$ , then it must be the case that  $\underline{\nu} < \bar{\nu}$ , where  $\underline{\nu} = \sqrt{d}\delta$  and  $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}}{4\sqrt{d}}$ . Since, at every round at least one of the landmarks gets a  $\nu$ -margin projected undesirability point, then if we make  $\tau$  sufficiently large, then, by the *pigeonhole principle*, at least one of the landmarks has  $\nu$ -margin projected undesirability at least  $\bar{c} \cdot (d + 1) + 1$ , which allows us to distinguish it from points in  $\text{conv}(\mathcal{P}(\bar{c}))$ . Formally (with proof in D.3.1):

**Lemma 7.4.** *For scalar  $\nu \in (\underline{\nu}, \bar{\nu})$ , after  $\tau = 2d \cdot \bar{c} \cdot (d + 1) + 1$  rounds in epoch  $\phi$ , there exists a landmark  $p^* \in \Lambda_\phi$  such that  $p^* \notin \text{conv}(\mathcal{P}(\bar{c}, \nu))$ .*

We can now prove the main lemma of this subsection. Note that during the computation of  $\mathbf{h}_\phi^*$ , nature does not provide any new context  $x$ , and hence, we incur no additional regret.

*Proof of Lemma 7.1.* By Lemma 7.4, for  $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\delta}{4\sqrt{d}}$  and  $\delta \in \left(0, \frac{\varepsilon}{2\sqrt{d} + 4d}\right)$ , there exists a landmark  $p^* \in \Lambda_\phi$  that lies outside of  $\text{conv}(\mathcal{P}(\bar{c}, \nu))$ . As a result, there exists a hyperplane separating  $p^*$  from the convex hull. We denote this hyperplane by  $(\mathbf{h}_\phi^*, \omega_\phi^*)$ . Recall that since  $p^* \in \Lambda_\phi$  then by definition  $\|\kappa_\phi - p^*\| = \bar{\nu}$ . As the hyperplane separates  $\kappa_\phi$  from  $p^*$ , it holds that  $\text{dist}(\kappa_\phi, (\mathbf{h}_\phi^*, \omega_\phi^*)) \leq \bar{\nu}$ . The fact that  $\theta^*$  is always in the preserved halfspace  $H_\phi(\mathbf{h}_\phi^*, \omega_\phi^*)$  follows directly from Lemma 7.2. ■

#### 7.4.2 PROOF OF THEOREM 7.1

We now provide the guarantee for the  $\bar{c}$ -known-corruption setting. Before delving into the details, we make two remarks. First, the regret guarantee of Proposition 7.1 is *deterministic*; only the runtime is randomized. Second, although the expected runtime is exponential in  $\bar{c}$ , the algorithm is eventually run with  $\bar{c} \approx \log(T)$ , which renders it quasipolynomial.

**Proposition 7.1.** *For the  $\bar{c}$ -known-corruption setting, the regret of CORPV.KNOWN for the  $\varepsilon$ -ball loss is  $\mathcal{O}((d^2\bar{c} + 1)d \log(d/\varepsilon))$ . When run with parameter  $\varepsilon = 1/T$ , its guarantee for the absolute and pricing loss is  $\mathcal{O}((d^2\bar{c} + 1)d \log(dT))$ . The expected runtime is  $\mathcal{O}((d^2\bar{c})^{\bar{c}} \cdot \text{poly}(d \log(d/\varepsilon), \bar{c}))$*

**Runtime of CorPV.SEPARATINGCUT (Algorithm 7.3)**. For what follows, let  $\mathcal{B}_{L_\phi}(\mathbf{p}^*, \zeta)$  be the ball of radius  $\zeta$  around  $\mathbf{p}^*$  in the space of large dimensions, where  $\mathbf{p}^* \in \Lambda_\phi$  is the landmark such that  $u_\phi(\mathbf{p}^*, \nu) = \bar{c} \cdot (d + 1) + 1$ . Recall that we proved the existence of a landmark  $\mathbf{p}^* \in \Lambda_\phi$  with this property in Lemma 7.4.

**Lemma 7.5.** *For any epoch  $\phi$ , scalar  $\delta \in (0, \frac{\varepsilon}{2\sqrt{d+4d}})$ , and scalar  $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\delta}{4\sqrt{d}}$ , after  $\tau = 2d \cdot \bar{c}(d + 1) + 1$  rounds, algorithm CorPV.SEPARATINGCUT computes hyperplane  $(\tilde{\mathbf{h}}_\phi, \tilde{\omega}_\phi)$  orthogonal to all small dimensions  $S_\phi$  such that  $\text{dist}(\kappa_\phi^*, (\tilde{\mathbf{h}}_\phi, \tilde{\omega}_\phi)) \leq 3\bar{\nu}$ , and halfspace  $\mathbf{H}^+(\tilde{\mathbf{h}}_\phi, \tilde{\omega}_\phi)$  always contains  $\theta^*$ . With probability at least  $(40d \cdot \sqrt{d-1})^{-1}$  the complexity of this computation is:*

$$\mathcal{O}\left(\frac{(d-1)}{\bar{\nu}^2} \cdot (d^2 \cdot \bar{c})^{\bar{c}} \cdot O(CP(d, \bar{c} \cdot (2d(d+1)-1)+1))\right)$$

where  $CP(n, m)$  is the complexity of solving a Convex Program with  $n$  variables and  $m$  constraints.

*Proof.* From Lemma 7.1, there exists a hyperplane  $(\mathbf{h}_\phi^*, \omega_\phi^*)$  with distance at most  $\bar{\nu}$  from  $\kappa_\phi$  that has all of  $\mathcal{P}(\bar{c}, \nu)$  inside  $\mathbf{H}^+(\mathbf{h}_\phi^*, \omega_\phi^*)$ . By arguing about the volume contained in any specified “cap” of a multi-dimensional ball, we prove that with probability at least  $(20\sqrt{d-1})^{-1}$  we can identify a point  $\mathbf{q}$  lying on the halfspace

$$\mathbf{H}^+\left(\mathbf{h}_\phi^*, \langle \mathbf{h}_\phi^*, \mathbf{p}^* \rangle + \frac{\zeta \cdot \ln(3/2)}{\sqrt{d-1}}\right).$$

This is formally stated and proved in Lemma D.7. In each iteration of CorPV.SEPARATINGCUT using point  $\mathbf{q}$  (i.e., lines 7–14 in Algorithm 7.3), Perceptron can identify a hyperplane  $(\tilde{\mathbf{h}}_\phi, \tilde{\omega}_\phi)$  separating  $\mathbf{q}$  and  $\mathcal{P}(\bar{c}, \nu)$  after  $\frac{d-1}{\zeta^2 \cdot \ln^2(3/2)}$  samples. The number of samples depends on the Perceptron mistake

bound. Since  $\mathbf{q} \in \mathcal{B}_{L_\phi}(\mathbf{p}^*, \zeta)$ , then,

$$\begin{aligned}
\|\mathbf{q} - \kappa_\phi^*\| &= \|\mathbf{q} - \kappa_\phi + \kappa_\phi - \kappa_\phi^*\| \\
&\leq \|\mathbf{q} - \kappa_\phi\| + \|\kappa_\phi - \kappa_\phi^*\| && \text{(triangle inequality)} \\
&\leq \|\mathbf{q} - \kappa_\phi\| + \bar{\nu} && \text{(approximation of } \kappa_\phi^* \text{ in polynomial time)} \\
&= \|\mathbf{q} - \mathbf{p}^* + \mathbf{p}^* - \kappa_\phi\| + \bar{\nu} \\
&\leq \|\mathbf{q} - \mathbf{p}^*\| + \|\mathbf{p}^* - \kappa_\phi\| + \bar{\nu} && \text{(triangle inequality)} \\
&\leq \bar{\nu} + \bar{\nu} + \bar{\nu} && (\mathbf{q} \in \mathcal{B}_{L_\phi}(\mathbf{p}^*, \zeta) \text{ and Definition 7.3})
\end{aligned}$$

Hence,  $\text{dist}(\kappa_\phi^*, (\tilde{\mathbf{h}}_\phi, \tilde{\omega}_\phi)) \leq 3\bar{\nu}$ .

Assume for now that the random landmark  $\mathbf{p}^* \in \Lambda_\phi$  chosen at Step 6 of Algorithm 7.3 is the landmark such that  $u_\phi(\mathbf{p}^*, \nu) \geq \bar{c} \cdot (d+1) + 1$ . Because  $\mathbf{p}^*$  is chosen uniformly at random from set  $\Lambda_\phi$ , then the probability of it being the target landmark is  $(2d)^{-1}$ , and is independent with all other random variables of Algorithm 7.3.

At every iteration of the inner while-loop of Algorithm 7.3, the algorithm checks whether  $\mathbf{q}$  and  $\kappa_\phi$  are on the “correct” side of  $\mathbf{h}$  and possibly updates the Perceptron (this is done in time  $\mathcal{O}(1)$ ): the centroid  $\kappa_\phi$  needs to be part of the  $\mathcal{P}(\bar{c}, \nu)$ , while  $\mathbf{q}$  needs to be separated from  $\mathcal{P}(\bar{c}, \nu)$ . Thus, they must belong to  $\mathbf{H}^+(\tilde{\mathbf{h}}, \tilde{\omega})$  and  $\mathbf{H}^-(\tilde{\mathbf{h}}, \tilde{\omega})$  respectively. This is done in Steps 9–10 of Algorithm 7.3. The most computationally demanding part of Algorithm 7.3 is Steps 10–13, where we check whether the hyperplane cuts some part of  $\mathcal{P}(\bar{c}, \nu)$ . For that, we check all the possible

$$\binom{|\mathcal{A}_\phi|}{|D_\phi|} = \binom{|\mathcal{A}_\phi|}{\bar{c}} \leq (2d \cdot \bar{c}(d+1) + 1)^{\bar{c}} = \Theta((d^2 \bar{c})^{\bar{c}})$$

combinations of which  $\bar{c}$  hyperplanes to disregard as corrupted and solve a mathematical program which we refer to as CP \* for each such combination. We remark that this computation serves the purpose of identifying which  $\bar{c}$  hyperplanes in  $\mathcal{A}_\phi$  were corrupted, thus giving erroneous feedback regarding where  $\theta^*$  lies. These CPs have at most  $d$  variables (since  $\mathcal{K}_\phi \subseteq \mathbb{R}^d$ ) and  $|\mathcal{A}_\phi| - \bar{c} = \bar{c} \cdot (2d(d+1) - 1) + 1$  constraints. Denoting the complexity of solving a CP with  $n$  variables and  $m$

---

\*Technically this program is convex and not linear as we also take intersection with the  $\mathcal{K}_\phi$  which is a convex body.

constraints as  $O(\text{CP}(n, m))$  we therefore obtain that Steps 10–13 have computational complexity

$$\mathcal{O} \left( O(\text{CP}(d, \bar{c} \cdot (2d(d+1) - 1) + 1)) \cdot (d^2 \bar{c})^{\bar{c}} \right).$$

Given that the event that  $\mathbf{p}^*$  is the target landmark is independent from the event that  $\tilde{\mathbf{q}}$  is found at the desired halfspace we have that with probability at least  $\frac{1}{2d} \cdot \frac{1}{20\sqrt{d-1}}$ , the computational complexity of Algorithm 7.3 is:

$$\mathcal{O} \left( \frac{(d-1)}{\bar{\nu}^2} \cdot (d^2 \cdot \bar{c})^{\bar{c}} \cdot O(\text{CP}(d, \bar{c} \cdot (2d(d+1) - 1) + 1)) \right)$$

This concludes our proof. ■

**Bounding the Number of Epochs.** The second step is to establish that we make enough volumetric progress when using  $(\tilde{\mathbf{h}}_\phi, \tilde{\omega}_\phi)$  as our separating cut for epoch  $\phi$ . We remark that in the analysis of [Lobel et al. \(2018\)](#), when `PROJECTEDVOLUME` observes a context  $\mathbf{x}_t$  such that  $w(\text{Cyl}(\mathcal{K}_\phi, S_\phi), \mathbf{x}_t) \leq \varepsilon$ , then it can directly discard it, since  $\mathbf{x}_t$  does not contribute to the regret with respect to the  $\varepsilon$ -ball loss function. This is because  $\mathbf{x}_t$ 's are used in order to make the separating cuts. In our epoch-based setting, the separating cuts are different than the observed contexts, as we have argued. Importantly, if  $w(\text{Cyl}(\mathcal{K}_\phi, S_\phi), \tilde{\mathbf{h}}_\phi) \leq \varepsilon$ , we cannot relate this information to the regret of epoch  $\phi$ , because for all rounds comprising the epoch, the width of  $\mathcal{K}_\phi$  in the direction of the observed context was greater than  $\varepsilon$  (Step 6 of Algorithm 7.1). This is shown in the following lemma.

**Lemma 7.6.** *After at most  $\Phi = O(d \log(d/\varepsilon))$  epochs, `CORPV.KNOWN` (Algorithm 7.1) has reached a knowledge set  $\mathcal{K}_\Phi$  with width at most  $\varepsilon$  in every direction  $\mathbf{u}$ .*

*Proof of Lemma 7.6.* We construct a potential function argument, similar to the one of [Lobel et al. \(2018\)](#) and we highlight the places where our analysis differs from theirs.

We use  $\Gamma_\phi = \text{vol}(\Pi_{L_\phi} \mathcal{K}_\phi)$  as the potential function. Its lower bound is  $\Gamma_\phi \geq \Omega\left(\frac{\delta}{d}\right)^{2d}$ . To see this, note that Step 6 of `CORPV.EPOCHUPDATES` (Algorithm 7.5) ensures that for all  $\mathbf{u} \in L_\phi$  it holds that  $w(\Pi_{L_\phi} \mathcal{K}_\phi, \mathbf{u}) \geq \delta$ . It is known (([Lobel et al., 2018, Lemma 6.3](#))/[Lemma D.9](#)) that if  $\mathcal{K} \subseteq \mathbb{R}^d$  is a convex body such that  $w(\mathcal{K}, \mathbf{u}) \geq \delta$  for every unit vector  $\mathbf{u}$ , then  $\mathcal{K}$  contains a ball of diameter

$\delta/d$ . This means that  $\Pi_{L_\phi} \mathcal{K}_\phi$  contains a ball of radius  $\frac{\delta}{|L_\phi|}$ , so

$$\text{vol}(\Pi_{L_\phi} \mathcal{K}_\phi) \geq V(|L_\phi|) \left( \frac{\delta}{|L_\phi|} \right)^{|L_\phi|},$$

where by  $V(|L_\phi|)$  we denote the volume of the  $|L_\phi|$ -dimensional unit ball. Using that  $|L_\phi| \leq d$  and  $V(d) \geq \Omega\left(\frac{1}{d}\right)^d$ , the latter can be lower bounded by  $\Omega\left(\frac{\delta}{d}\right)^{2d}$ . Hence,  $\Gamma_\phi = \text{vol}(\Pi_{L_\phi} \mathcal{K}_\phi) \geq \Omega\left(\frac{\delta}{d}\right)^{2d}$ .

We split our analysis of the upper bound of  $\Gamma_\phi$  in two parts. First, we study the potential function between epochs where the set of large dimensions  $L_\phi$  does not change. Second, we study the potential function between where the set of large dimensions  $L_\phi$  becomes smaller. For both cases, we prove the following useful result (Lemma D.15) which relates the volume of  $\Pi_{L_\phi} \mathcal{K}_{\phi+1}$  with the volume of  $\Pi_{L_\phi} \mathcal{K}_\phi$  when  $\delta = \frac{\varepsilon}{4(d+\sqrt{d})}$ :

$$\text{vol}(\Pi_{L_\phi} \mathcal{K}_{\phi+1}) \leq \left(1 - \frac{1}{2e^2}\right) \text{vol}(\Pi_{L_\phi} \mathcal{K}_\phi) \quad (7.4.1)$$

When the set  $L_\phi$  does not change (i.e.,  $L_\phi = L_{\phi+1}$ ) Equation (7.4.1) becomes:

$$\text{vol}(\Pi_{L_{\phi+1}} \mathcal{K}_{\phi+1}) \leq \left(1 - \frac{1}{2e^2}\right) \text{vol}(\Pi_{L_\phi} \mathcal{K}_\phi) \quad (7.4.2)$$

When  $L_\phi$  does change, then the set of small dimensions increases from  $S_\phi$  to  $S_{\phi+1}$ . In order to correlate  $\text{vol}(\Pi_{L_{\phi+1}} \mathcal{K}_{\phi+1})$  with  $\text{vol}(\Pi_{L_\phi} \mathcal{K}_\phi)$  we make use of the following known inequality ((Lobel et al., 2018, Lemma 6.1) or Lemma D.14) that for a convex body  $\mathcal{K} \subseteq \mathbb{R}^d$  if  $w(\mathcal{K}, \mathbf{u}) \geq \delta'$  (for some scalar  $\delta' > 0$ , for every unit vector  $\mathbf{u}$ ), then, for every  $(d-1)$ -dimensional subspace  $L$  it holds that:

$$\text{vol}(\Pi_L \mathcal{K}) \leq \frac{d(d+1)}{\delta'} \text{vol}(\mathcal{K}) \quad (7.4.3)$$

We are going to apply Equation (7.4.3) for  $\mathcal{K} := \Pi_{L_\phi} \mathcal{K}_{\phi+1}$  and  $L := L_{\phi+1}$ . For that, we need to find  $\delta'$  for which  $w(\Pi_{L_\phi} \mathcal{K}) \geq \delta'$ .

We make use of the following lemma ((Lobel et al., 2018, Theorem 5.3)) which relates the width of  $\mathcal{K}_+$ :  $\mathcal{K}_+ = \mathcal{K} \cap \{\mathbf{x} | \langle \mathbf{u}, \mathbf{x} - \boldsymbol{\kappa} \rangle = 0\}$  (where  $\boldsymbol{\kappa}$  is the centroid of  $\mathcal{K}$  and  $\mathbf{u}$  is any unit vector) with

the width of  $\mathcal{K}$  in the direction of any unit vector  $\mathbf{v}$  as follows:

$$\frac{1}{d+1}w(\mathcal{K}, \mathbf{v}) \leq w(\mathcal{K}_+, \mathbf{v}) \leq w(\mathcal{K}, \mathbf{v}) \quad (7.4.4)$$

By the definition of large dimensions,  $w(\mathcal{K}_\phi, \mathbf{u}) \geq \delta, \forall \mathbf{u} \in L_\phi$ . So, if we were to cut  $\mathcal{K}_\phi$  with a hyperplane that passes precisely from the centroid  $\kappa_\phi^*$  then, from Equation (7.4.4):  $w(\mathcal{K}_+, \mathbf{u}) \geq \frac{\delta}{d+1}$ . Since, however, we make sure that we cut  $\mathcal{K}_\phi$  through the approximate centroid  $\kappa_\phi$ , then  $w(\mathcal{K}_+, \mathbf{u}) \geq \frac{\delta}{d+1}$ . This is because the halfspace  $\mathbf{H}^+(\tilde{\mathbf{h}}_\phi, \tilde{\omega}_\phi)$  returned from **CorPV.SEPARATINGCUT** (Algorithm 7.3) always contains  $\kappa_\phi$ . Since  $\|\kappa_\phi - \kappa_\phi^*\| \leq \bar{\nu}$  then  $\kappa_\phi^* \in \mathbf{H}^+(\tilde{\mathbf{h}}_\phi, \tilde{\omega}_\phi)$ . As a result, from Equation (7.4.3) we get:

$$\text{vol}(\Pi_{L_{\phi+1}} \mathcal{K}_{\phi+1}) \leq \frac{d(d+1)^2}{\delta} \text{vol}(\Pi_{L_\phi} \mathcal{K}_{\phi+1}) \quad (7.4.5)$$

We have almost obtained the target Equation (7.4.1). To complete the argument we need to correlate  $\text{vol}(\Pi_{L_\phi} \mathcal{K}_{\phi+1})$  with  $\text{vol}(\Pi_{L_\phi} \mathcal{K}_\phi)$ . Lemma D.15 proves the following inequality between the two:

$$\text{vol}(\Pi_{L_\phi} \mathcal{K}_{\phi+1}) \leq \left(1 - \frac{1}{2e^2}\right) \text{vol}(\Pi_{L_\phi} \mathcal{K}_\phi) \quad (7.4.6)$$

Combining Equations (7.4.5) and (7.4.6) and using the fact that we add at most  $d$  new directions to  $S_\phi$ , that the volume of  $\mathcal{K}_0$  is upper bounded by  $O(1)$ , and the lower bound for  $\Gamma_\phi$  we computed, we have:

$$\Omega\left(\frac{\delta}{d}\right)^{2d} \leq \Gamma_\Phi = \text{vol}(\Pi_{L_\Phi} \mathcal{K}_\Phi) \leq O(1) \cdot \left(\frac{d(d+1)^2}{\delta}\right)^d \cdot \left(1 - \frac{1}{2e^2}\right)^\Phi$$

where by  $\Phi$  we denote the total number of epochs. Solving the above in terms of  $\Phi$  and substituting  $\delta$  we obtain:  $\Phi \leq O(d \log \frac{d}{\varepsilon})$ . ■

We are finally ready to prove Proposition 7.1.

*Proof.* We first analyze the regret for the  $\varepsilon$ -ball loss function. Lemma 7.6 establishes that after  $\Phi = O(d \log(d/\varepsilon))$  epochs, the set of large dimensions  $L_\phi$  is empty. When  $L_\phi = \emptyset$ , then  $S_\phi$  must be an orthonormal basis for which  $w(\mathcal{K}_\phi, \mathbf{s}) \leq \delta, \forall \mathbf{s} \in S_\phi$ . For any  $\mathbf{x}_t$  after  $L_\phi = \emptyset$ , we have the following. First,  $\mathbf{x}_t = \sum_{i \in [|S_\phi|]} a_i \cdot s_i$ , and for any vector  $\mathbf{a}$  it holds that:  $\|\mathbf{a}\|_1 \leq \sqrt{d} \cdot \|\mathbf{a}\|_2$ . So, the

width of  $\mathcal{K}_\phi$  in the direction of  $\mathbf{x}_t$  when  $|S_\phi| = d$  is:

$$\begin{aligned}
 w(\mathcal{K}_\phi, \mathbf{x}_t) &= \max_{\mathbf{p}, \mathbf{q} \in \mathcal{K}_\phi} \langle \mathbf{x}_t, \mathbf{p} - \mathbf{q} \rangle && (\text{definition of width}) \\
 &\leq \sum_{i \in [|S_\phi|]} a_i \cdot \max_{\mathbf{p}, \mathbf{q} \in \mathcal{K}_\phi} \langle s_i, \mathbf{p} - \mathbf{q} \rangle && (\mathbf{x}_t = \sum_{i \in [|S_\phi|]} a_i \mathbf{s}_i \text{ and properties of } \max(\cdot)) \\
 &\leq \sum_{i \in [|S_\phi|]} a_i \cdot \delta && (\text{definition of small dimensions}) \\
 &\leq \|\mathbf{a}\|_1 \cdot \delta \leq \sqrt{d} \cdot \delta \cdot \|\mathbf{a}\|_2 && (\|\mathbf{a}\|_1 \leq \sqrt{d} \cdot \|\mathbf{a}\|_2)
 \end{aligned}$$

Substituting  $\delta = \frac{\varepsilon}{4(d+\sqrt{d})}$  the latter becomes:  $w(\mathcal{K}_\phi, \mathbf{x}_t) \leq \frac{\varepsilon}{4(\sqrt{d}+1)} \leq \varepsilon$ . Therefore, when  $L_\phi = \emptyset$ , then CORPV.KNOWN incurs no additional regret for any context it receives in the future, if we are interested in the  $\varepsilon$ -ball loss. Using the fact that each epoch contains  $\tau = 2d \cdot \bar{c}(d+1) + 1$  rounds during which we can incur a loss of at most 1 we have that the regret for the  $\varepsilon$ -ball loss is equal to:

$$R_{\varepsilon\text{-ball}}(T) = O\left(d \log \frac{d}{\varepsilon}\right) \cdot (2d \cdot \bar{c} \cdot (d+1) + 1) = \mathcal{O}\left((d^2 \bar{c} + 1)d \log\left(\frac{d}{\varepsilon}\right)\right)$$

For the absolute and the pricing loss, for every round after the set  $L_\phi$  has become empty, the queried point incurs a loss of at most  $\varepsilon$ . As a result the regret for both cases is *at most*

$$R_{\varepsilon\text{-ball}}(T) + \varepsilon \cdot T$$

Tuning  $\varepsilon = 1/T$  we get the result for these two loss functions. ■

**Extending to Unknown Corruption  $C$ .** To turn Proposition 7.1 to Theorem 7.1, similar to Lykouris et al. (2018), we separate the layers  $j$  of Algorithm 7.4 into corruption-tolerant ( $j \geq \log C$ ) and corruption-intolerant ( $j < \log C$ ). Since the corruption-tolerant layers, with high probability do not remove  $\theta^*$  from their parameter set, we view them as running independently for analysis purposes; each results to a regret equal to the one of Proposition 7.1 with  $\bar{c} = \log T$ . The corruption-intolerant layers may eliminate  $\theta^*$  but their knowledge set is eventually refined by the knowledge set of the first corruption-tolerant layer  $\lceil \log C \rceil$  thanks to global eliminations. Since the latter is selected with probability  $1/C$  at every round, the time it will take for it to make volumetric progress

is  $C$  times what it would happen if it was run independently. The full proof is provided in Appendix D.2.

**Relationship to Ulam’s Game.** Ulam’s game is a non-contextual (1-dimensional) version of our problem with a known corruption level  $C$  and the  $\varepsilon$ -ball loss. In that setting, [Rivest et al. \(1980\)](#) show that the optimal query complexity  $Q$  for localizing  $\theta^*$  to a region of volume  $\varepsilon$  satisfies  $\varepsilon \geq \sum_{i=0}^C \binom{Q}{i} \cdot 2^{-Q}$ . The authors state that this implies a query complexity lower bound of  $\Omega(\log(1/\varepsilon) + C \log \log(1/\varepsilon) + C \log C)$ . Beyond the fact that we consider the contextual setting, a subtle distinction between this analysis and ours is the difference between query complexity and regret. When measuring query complexity, we count every round until we can certify that we have localized  $\theta^*$ , but the  $\varepsilon$ -ball loss may be zero on rounds prior to this event. Thus, the  $\varepsilon$ -ball loss is always smaller than the query complexity in the single dimensional setting. In the contextual setting, query complexity is not a meaningful metric as the adversary can inject many queries in directions that we have already learned without changing the problem. However a notion of *explore-query complexity* that counts the number of times the algorithm makes a mistake or uses the response of the round is meaningful. Our analysis actually bounds the explore-query complexity. For this metric, the lower bound in ([Rivest et al., 1980](#)) suggests that a multiplicative relationship between  $C$  and some function of  $T$  (which appears in our bound) is unavoidable when  $\varepsilon = 1/T$ . It is an interesting open question to understand whether this is the case for the regret definition that only penalizes the number of mistakes and for other loss functions.

#### 7.4.3 DISCUSSION OF ALGORITHMIC CHOICES AFFECTING THE REGRET GUARANTEE

At this point, one would wonder whether `PROJECTEDVOLUME` has some particular special property that makes it amenable to our technique or whether we provided a generic reduction from any uncorrupted contextual search algorithm. It turns out that our approach relies on two properties of the uncorrupted algorithm: a) it needs to be binary-search, i.e., work with a knowledge set and refine it over time and b) separate the space in small and large dimensions. The latter is important as we do not make a cut on one of the existing contexts but rather combine them appropriately. As a result, an algorithm that works with projection on the large dimensions is always guaranteed to return a cut on that projection (therefore with sufficiently large width enabling volumetric progress).

This is a property that is particular to PROJECTEDVOLUME and is not shared by other algorithms. We elaborate upon this discussion below.

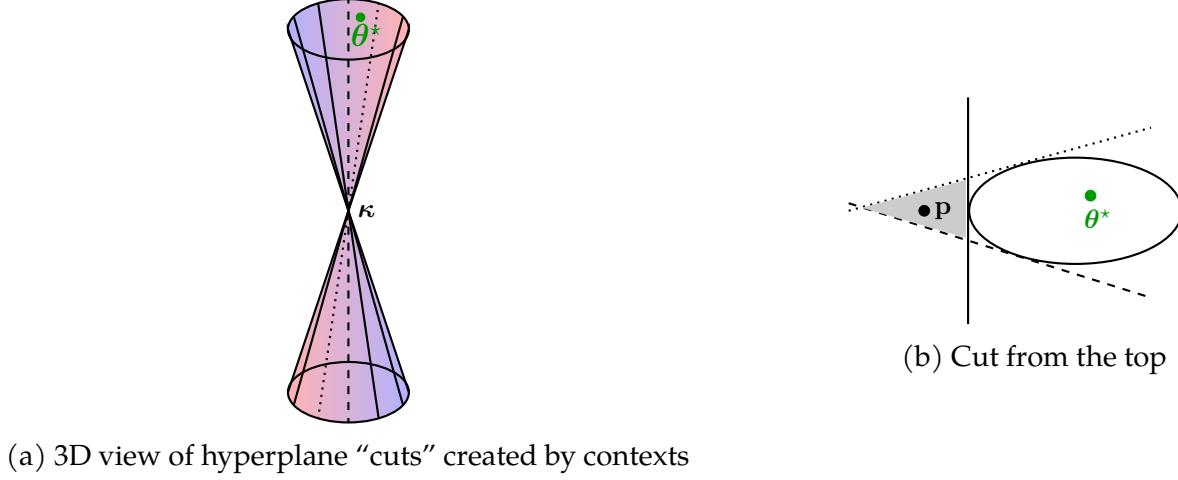


Figure 7.2: Sketch on why proper cuts do not suffice.

We finally discuss algorithmic choices affecting the regret guarantee.

**Not Making Updates by Single Context.** All the contextual search approaches that are based on binary search techniques rely on refining a similarly constructed knowledge set (or, in other words, version space) that contains the ground truth  $\theta^*$ . In all of the previous non-corrupted works, one could just use every explore query in order to refine this knowledge set. However, in our corrupted setting, this technique can result in the removal of  $\theta^*$  from the knowledge set (as can be seen by a simple one-dimensional example formalized in Appendix D.1.2). So, to employ such binary-search techniques, we need to be more careful about when and how we refine the knowledge set. Our approach is to only remove from the knowledge set parameters that are certifiably not  $\theta^*$ . To simplify the subsequent discussion, assume that contexts  $\{\mathbf{x}_t\}_{t \in [\tau]}$  lead only to *explore* queries and that we are still interested in the simpler  $\bar{c}$ -known corruption setting with  $\bar{c} = 1$ .

**Creating a separating cut by combining explore queries.** Ideally, if we could identify one context  $\mathbf{x} \in \{\mathbf{x}_t\}_{t \in [\tau]}$  such that the  $\bar{c}$ -protected region  $\mathcal{P}(\bar{c}, \nu)$  is inside the halfspace  $\mathbf{H}^+(\mathbf{x}, \kappa_\phi)$ , i.e.,  $\mathcal{P}(\bar{c}, \nu) \subseteq \mathbf{H}^+(\mathbf{x}, \kappa_\phi)$ , then we could update  $\mathcal{K}_{\phi+1}$  as  $\mathcal{K}_\phi \cap \mathbf{H}^+(\mathbf{x}, \kappa_\phi)$ . As we have explained, these properties ensure sufficient volumetric progress. In  $d = 2$ , indeed one of the contexts among  $\{\mathbf{x}_t\}_{t \in [\tau]}$  has the aforementioned property due to a monotonicity argument.

However, this is no longer true in  $d = 3$ , even if one sees arbitrarily many contexts in an epoch. To see this, consider Figure 7.2 and assume that all rounds are *uncorrupted*. In Figure 7.2a, each straight line corresponds to a context  $x_t$  and the shaded region corresponds to the halfspace with feedback  $y_t = +1$ , forming this “cone.” In Figure 7.2b we visualize a cross section of the knowledge set shown in Figure 7.2a and zoom in on only 3 of the halfspaces around  $\theta^*$ ; the dotted, the dashed and the solid.

We are going to reason about the undesirability level of points like  $p$ , lying in the shaded area of Figure 7.2b. Points  $p$  and  $\theta^*$  lie on the same side of both the dashed and dotted hyperplanes, and so these two do not contribute any undesirability to  $p$ . The solid line does contribute once to the undesirability level of  $p$  (and all the points in the shaded region). Recall that since  $\bar{c} = 1$ , we need a hyperplane with undesirability at least 2 in the *entirety* of one of its halfspaces. However, for any number of contexts, we can form the cone structure in Figure 7.2a, in which, for every hyperplane, there exists a shaded region like the one in Figure 7.2b, whose points have undesirability 1.

On the other hand, there exists another hyperplane (not associated with a single explore query) with undesirability at least  $\bar{c} + 1$ . This is the cut that separates the upper part of the cone in Figure 7.2a containing  $\theta^*$  from the lower part.

**Separating into Small and Large Dimensions.** In the beginning of this discussion, we assumed that we will focus only on *explore* queries. This is important, as cuts that are made in directions of small width do not necessarily adequately refine our estimate for  $\theta^*$ . That said, since we have established that the cut we make may not correspond to any of the observed contexts, we cannot automatically guarantee that the width of the direction for that cut will indeed be large enough.

To deal with this, we separate the dimensions into small and large and project all objects into the subspace spanned by the “large dimensions”. This guarantees that any cut that we create will also live in the large dimension subspace and will therefore have sufficiently large width to enable adequate volumetric progress. This is the place where our approach is tailored to the PROJECTEDVOLUME algorithm rather than being a generic reduction to any binary-search method for the uncorrupted case as, to the best of our knowledge, the PROJECTEDVOLUME is the only one that explicitly separates small and large dimensions, which makes it amenable for our purposes.

**Employing Landmarks.** So far we have clarified the need to handle small and large dimensions of the knowledge set separately. As a next step, Carathéodory’s Theorem provides an upper bound in the undesirability of all the points within  $\text{conv}(\mathcal{P}(\bar{c}))$ . The last step is to identify at least one point *in the large dimensions* that has undesirability strictly larger than the bound provided by Carathéodory’s theorem. This is where our “landmarks” construction comes into play and serves the following dual purpose. On the one hand, at least one of them has large enough undesirability that it cannot be in  $\text{conv}(\mathcal{P}(\bar{c}))$  and hence, this landmark can be separated from  $\text{conv}(\mathcal{P}(\bar{c}))$  using the Perceptron algorithm. On the other hand, because of their construction, the hyperplane returned by Perceptron is guaranteed to be valid, meaning that the knowledge set has large width in its direction.

**Remark 7.1.** *The additional  $d^2$  degradation in the regret compared to the uncorrupted case arises from the use of Carathéodory’s Theorem and the use of landmarks respectively. We view the former as inherent to our approach and hence, achieving a linear dependence on  $d$  would require fundamentally new ideas. Regarding the latter, landmarks may be an artifact of the particular analysis and there may be other ways to identify such a highly undesirable point while still retaining the main principles of our methodological approach.*

**Further details on the need to combine multiple explore queries.** In order to prove the results of this section, we use a simplified version of undesirability levels; we define a point’s  $\mathbf{p} \in \mathcal{K}_\phi$  undesirability level as the number of rounds within epoch  $\phi$ , for which

$$u_\phi(\mathbf{p}) = \sum_{t \in [\tau]} \mathbb{1} \{ \langle \mathbf{p} - \boldsymbol{\kappa}_\phi, \mathbf{x}_t \rangle \cdot y_t < 0 \}.$$

We next present two propositions regarding the number of contexts needed to guarantee that we have found an undesirable hyperplane for  $d = 2$  and  $d = 3$  respectively.

**Proposition 7.2.** *For  $d = 2$  and any corruption level  $\bar{c}$ , after  $3\bar{c} + 1$  rounds within an epoch, there exists a hyperplane  $(\mathbf{x}', \omega')$  among  $\{(\mathbf{x}_t, \omega_t)\}_{t \in [\tau]}$  with undesirability level at least  $\bar{c} + 1$  in the entirety of one of its halfspaces.*

*Proof.* Since there exist at most  $\bar{c}$  corrupted rounds among the  $3\bar{c} + 1$  rounds of epoch  $\phi$ , then at least  $2\bar{c} + 1$  are *uncorrupted*. We say that these rounds are part of the set  $U_\phi$ . For all  $t \in U_\phi$ , the learner’s

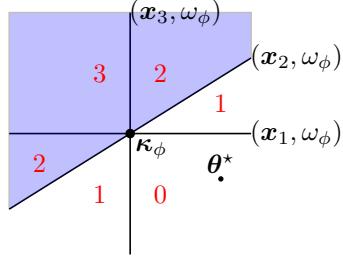


Figure 7.3: Sketch of the undesirability levels for epoch  $\phi$ , after  $2\bar{c} + 1$  uncorrupted rounds, assuming that each context appears once. Red numbers denote the undesirability levels. The opaque region denotes the knowledge set for epoch  $\phi + 1$ .

hyperplanes  $\{(x_t, \omega_t)\}_{t \in U_\phi}$  pass from the same centroid  $\kappa_\phi$  and they all *protect* the region where  $\theta^*$  lies. In other words, none among  $\{(x_t, \omega_t)\}_{t \in U_\phi}$  adds an undesirability point to  $\theta^*$  (see e.g., Figure 7.3 for  $\bar{c} = 1$  and each context appears only once). Since all hyperplanes point towards the same direction (i.e., the region containing  $\theta^*$  never gets an undesirability point), starting from the region where  $\theta^*$  lies and moving counter clockwise the undesirability levels of the formed regions first increase (moving from 0 to  $2\bar{c} + 1$ ) and then decrease (moving from  $2\bar{c} + 1$  to 0). Due to this being a concave function, it is clear to see that there always exists a hyperplane with undesirability level at least  $\bar{c} + 1$  in the entirety of one of its halfspaces. ■

**Proposition 7.3.** *For  $d = 3$ , any corruption  $\bar{c}$ , any centroid  $\kappa$ , and any number of rounds  $N$  within an epoch, there exists a  $\theta^*$  and a sequence  $\{x_t\}_{t \in [N]}$ , such that there does not exist a hyperplane  $(x', \omega')$ , where  $x' \in \{x_t\}_{t \in [N]}$ , with one of its halfspaces having undesirability at least  $\bar{c} + 1$ .*

*Proof.* For any convex body  $\mathcal{K}$  with centroid  $\kappa$ , we show how to construct a problematic instance of a  $\theta^*$  and  $N$  contexts. Fix the corruption level to be  $\bar{c} = 1$ , and  $c_t = 0, \forall t \in [N]$ . However, the learner does not know that none of the rounds is corrupted. Construct a sequence of contexts  $\{x_t\}_{t \in [N]}$  such that no two are equal and for  $\omega_t = \langle x_t, \kappa \rangle, \forall t \in [N]$ :

$$\{(x_{t_1}, \omega_{t_1})\} \cap \{(x_{t_2}, \omega_{t_2})\} = \kappa$$

and the smallest region  $r^*$  that contains  $\theta^*$  is defined by all  $\{x_t\}_{t \in [N]}$ . Intuitively, these hyperplanes form a conic hull.

Take any hyperplane  $h \in \mathbb{R}^3$  neither parallel nor orthogonal with any hyperplane among  $\{(x_t, \omega_t)\}_{t \in [N]}$

such that  $h \cap r^* = q \neq \emptyset$ . Take  $q$ 's projection in  $\mathbb{R}^2$ . Observe that we have constructed an instance where no matter how big  $N$  is, there does not exist any hyperplane with undesirability at least  $\bar{c} + 1$  (i.e., 2 when  $\bar{c} = 1$ ) in either one of its halfspaces. This instance easily generalizes for any  $\bar{c} > 1$ . ■

## 7.5 EXTENSION TO BOUNDED RATIONALITY

We now extend the algorithm and analysis to the bounded rationality behavioral model. We first recap the behavioral model. There is a noise parameter  $\xi_t$  sampled from a  $\sigma$ -subgaussian distribution  $\text{subG}(\sigma)$ , *fixed* across rounds and *known* to the learner, i.e., nature selects it before the first round and reveals it. At every round  $t$  a realized noise  $\xi_t \sim \text{subG}(\sigma)$  is drawn, but  $\xi_t$  is never revealed to the learner. The agent's perceived value is then  $\tilde{v}_t = v(\mathbf{x}_t) + \xi_t$ .

We focus on a *pseudo-regret* definition that compares to a benchmark that has access to  $\theta^*$  and  $\text{subG}(\sigma)$  but does not have access to the realization  $\xi_t$ . The resulting benchmark is:

$$L_{\theta^*}^*(\mathbf{x}) = \min_{\omega^*} \mathbb{E}_{\xi' \in \text{subG}(\sigma)} [\ell(\omega^*, \langle \mathbf{x}, \theta^* \rangle, \langle \mathbf{x}, \theta^* \rangle + \xi')]. \quad (7.5.1)$$

and the corresponding regret is  $R(T) = \sum_{t \in [T]} [\ell(\omega_t, v(\mathbf{x}_t), \tilde{v}_t) - L_{\theta^*}^*(\mathbf{x}_t)]$ .

We remark that  $\omega^*$  should be thought of as the optimal query that the learner could have issued had we known  $\theta^*$  but not the realization of  $\xi'$ . For more intuition, assume for example that  $\xi'$  comes from a normal distribution. Then, the optimal  $\omega^*$  in expectation for the  $\varepsilon$ -ball and the absolute loss is equal to  $\langle \mathbf{x}, \theta^* \rangle$ . However,  $\omega^*$  should be strictly *lower* than  $\langle \mathbf{x}, \theta^* \rangle$  when interested in the pricing loss, due to its discontinuity.

Our algorithm only differs from the one described in Section 7.3 in the EXPLOIT module (Algorithm 7.6) as  $\omega_t$  is defined in a similar way with the benchmark. More formally, we again consider the worst-case selection of  $\theta$  consistent with the knowledge set and select the query that minimizes our loss with respect to that, i.e.,  $\omega_t = \arg \min_{\omega} \max_{\theta \in \mathcal{K}_\phi} \mathbb{E}_{\xi' \in \text{subG}(\sigma)} [\ell(\omega, \langle \mathbf{x}_t, \theta^* \rangle, \langle \mathbf{x}_t, \theta^*, \mathbf{x}_t \rangle + \xi')]$ . The algorithm also doubles the corruption budget it should be robust to ( $\bar{c}$  in Algorithm 7.4) and handles the extra noise by treating its tail as corruption and upper bounding it by  $\bar{c}$ .

### Theorem 7.2: Regret for Boundedly Rational Agents

With probability at least  $1 - 2\beta$ , the guarantee of Theorem 7.1 extends to when rounds with fully rational agents are replaced by boundedly rational with  $\sigma \leq \frac{\varepsilon}{8\sqrt{2d}(\sqrt{d}+1)\ln T}$ .

We note that Corollary 1 in (Cohen et al., 2020) has a regret of  $\mathcal{O}(d^2 \log T)$  for pricing loss with  $\sigma \approx \frac{d}{T \log T}$ . For pricing loss,  $\varepsilon = \frac{1}{T}$  and our bound is weaker by a factor of  $d$  on the regret and a factor  $d^2$  on the subgaussian variance  $\sigma$ , but it allows for the simultaneous presence of adversarially corrupted agents.

*Proof of Theorem 7.2.* We first show that under the low-noise regime, the noise is bounded by  $\Xi = \sqrt{2}\sigma \ln T$  with high probability for all  $t$ . Indeed, by Hoeffding's inequality we have that  $\Pr[|\xi_t| > \Xi] \leq e^{-\ln^2 T}$ . Using the union bound, we have:  $\Pr[|\xi_t| > \Xi, \text{ for any } t \in [T]] \leq \beta' = \beta/T$ . Hence,

$$\Pr[|\xi_t| \leq \Xi, \forall t \in [T]] \geq 1 - \beta,$$

which contributes the additional  $\beta$  in the high-probability argument.

We next show that when  $\sigma \leq \frac{\varepsilon}{8\sqrt{2d}(\sqrt{d}+1)\ln T}$ , then our algorithm maintains  $\theta^*$  in  $\mathcal{K}_\phi$ . This is enough to ensure that the regret guarantee remains order unchanged. Since the perceived value of BR agents is  $\tilde{v}_t = v(\mathbf{x}_t) + \xi_t$ , then, in order to "protect"  $\theta^*$  (i.e., make sure that  $\theta^* \in \mathcal{K}_{\phi+1}$ ) we need the hyperplanes that we feed to `CORPVSEPARATINGCUT` to have a margin of  $\Xi$  (since  $\xi_t \leq \Xi$ ). To do so, it suffices to slightly change the lower bound of  $\nu$  for the  $\nu$ -margin projected undesirability levels that we use throughout the proof such that the new lower bound is  $\underline{\nu}' = \underline{\nu} + \Xi = \sqrt{d} \cdot \delta + \Xi$ . Since  $\nu$  is such that  $\underline{\nu}' \leq \nu \leq \bar{\nu}$ , then it must the case that  $\underline{\nu}' = \sqrt{d}\delta + \Xi \leq \bar{\nu} = \frac{\varepsilon(2\sqrt{d}+1)}{8\sqrt{d}(\sqrt{d}+1)}$ . Solving for  $\Xi$  we obtain the result. This concludes our proof. ■

## 7.6 GRADIENT DESCENT ALGORITHM

In this section, we propose our second algorithm, which is a variant of gradient descent and works for contextual search with absolute and  $\varepsilon$ -ball loss. This algorithm is significantly simpler than algorithms based on binary search methods and has a better running time. On the other hand, it does not provide logarithmic guarantees when  $C \approx 0$  and it does not extend to the pricing loss.

---

**ALGORITHM 7.7: CONTEXTUALSEARCH.GD**


---

- 1 Initialize  $\mathbf{z}_1 \in \mathcal{K}$  and  $\gamma_1 = 1/2$ .
  - 2 **for** rounds  $t = \{1, \dots, T\}$  **do**
  - 3     For context  $\mathbf{x}_t$ , query  $\omega_t = \langle \mathbf{x}_t, \mathbf{z}_t \rangle$  and receive feedback:  $y_t = \text{sgn}(\omega_t - \langle \boldsymbol{\theta}^*, \mathbf{x}_t \rangle)$ .
  - 4     Choose  $\mathbf{z}_{t+1} = \Pi_{\mathcal{K}}(\mathbf{z}_t - \gamma_t \nabla f_t(\mathbf{z}_t))$ , where  $\gamma_t = \min\{1/2, \sqrt{2/T}\}$  and  $f_t(\mathbf{z}) = -y_t \cdot \langle \mathbf{z}, \mathbf{x}_t \rangle$ .
- 

For the intuition behind Algorithm 7.7, we restrict our attention to the absolute loss and recall that our goal is to minimize it using only binary feedback. The algorithm optimizes a *proxy* function  $f_t(\mathbf{z}) : \mathcal{K} \rightarrow \mathbb{R}^d$ , which is Lipschitz. Specifically, denoting the binary feedback by  $y_t = \text{sgn}(\omega_t - \langle \mathbf{x}_t, \boldsymbol{\theta}^* \rangle)$ , the proxy function is  $f_t(\mathbf{z}) = -y_t \cdot \langle \mathbf{x}_t, \mathbf{z} \rangle$ . The query point at the next round  $t + 1$  is  $\omega_{t+1} = \langle \mathbf{x}_{t+1}, \mathbf{z}_{t+1} \rangle$ . Note here that  $y_t$  is the subgradient of the target loss function  $|\langle \boldsymbol{\theta}^* - \mathbf{z}, \mathbf{x}_t \rangle|$ . The proxy function  $f_t(\mathbf{z})$  is convenient because on the one hand, it is Lipschitz and on the other, its regret is an *upper bound* on the regret incurred by any algorithm optimizing the absolute loss for the same problem. In the presence of corruptions, the *same* algorithm suffers regret  $\mathcal{O}(\sqrt{T} + C)$ ; this is due to the fact that corruptions only add an extra set of  $C$  erroneous rounds, from which the algorithm can certainly “recover” as there is no notion of a shrinking knowledge set.

**Theorem 7.3: Regret of CONTEXTUALSEARCH.GD**

For an unknown corruption level  $C$ , CONTEXTUALSEARCH.GD incurs regret  $\mathcal{O}(\sqrt{T} + C)$  in expectation for the absolute loss and  $\mathcal{O}(\sqrt{T}/\varepsilon + C/\varepsilon)$  for the  $\varepsilon$ -ball loss.

*Proof.* Function  $f_t(\mathbf{z}) = -y_t \cdot \langle \mathbf{z}, \mathbf{x}_t \rangle$  is Lipschitz in  $\mathbf{z}$ . So, using the known guarantees for Online Gradient Descent and denoting by  $\mathbf{z}^* = \arg \min_{\mathbf{z}} \sum_{t \in [T]} f_t(\mathbf{z})$  we know that:

$$\sum_{t \in [T]} f_t(\mathbf{z}_t) - \sum_{t \in [T]} f_t(\mathbf{z}^*) = \mathcal{O}(\sqrt{T}) \quad (7.6.1)$$

Due to the definition of  $\mathbf{z}^*$  we can relax the left-hand side of Equation (7.6.1) and get:

$$\sum_{t \in [T]} f_t(\mathbf{z}_t) - \sum_{t \in [T]} f_t(\boldsymbol{\theta}^*) \leq \mathcal{O}(\sqrt{T}) \quad (7.6.2)$$

We now analyze the quantity on the left-hand side of Equation (7.6.2) as follows:

$$\sum_{t \in [T]} f_t(\mathbf{z}_t) - \sum_{t \in [T]} f_t(\boldsymbol{\theta}^*) = \sum_{t \in [T]} y_t \cdot (\langle \boldsymbol{\theta}^*, \mathbf{x}_t \rangle - \langle \mathbf{z}_t, \mathbf{x}_t \rangle) = \sum_{t \in [T]} |\langle \boldsymbol{\theta}^*, \mathbf{x}_t \rangle - \omega_t| \quad (7.6.3)$$

which is the quantity that we wish to minimize when we are trying to minimize the absolute loss given binary feedback when the round is not corrupted. Given that  $y_t$  is arbitrary for at most  $C$  rounds, the regret incurred by `CONTEXTUALSEARCH.GD` is at most  $\mathcal{O}(\sqrt{T} + C)$ .

For the  $\varepsilon$ -ball loss, we show how the latter compares with the absolute loss. Indeed:

$$\sum_{t \in [T]} |\langle \boldsymbol{\theta}^*, \mathbf{x}_t \rangle - \omega_t| \geq \varepsilon \sum_{t \in [T]} \mathbb{1}\{|\langle \boldsymbol{\theta}^*, \mathbf{x}_t \rangle - \omega_t| \geq \varepsilon\}$$

Combining the above with Equation (7.6.3), we establish that `CONTEXTUALSEARCH.GD` for the  $\varepsilon$ -ball loss incurs regret  $\mathcal{O}(\sqrt{T}/\varepsilon + C/\varepsilon)$  ■

## 7.7 DISCUSSION AND OPEN QUESTIONS

The results in this chapter open up many fruitful avenues for future research. First, the regret in both of our algorithms is sublinear when  $C = o(T)$ , but becomes linear when  $C = \Theta(T)$ . Designing algorithms that can provide sublinear regret against the ex-post best linear model, in the latter regime, is an exciting direction of future research and our model offers a concrete formulation of this problem. Second, our algorithm that attains the logarithmic guarantee has a regret of the order of  $\mathcal{O}(Cd^3\text{poly}(\log T))$ . It would be interesting to either refine our approach or provide new algorithms that improve the dependence on  $d$  and also remove the dependence on  $T$  for the absolute and  $\varepsilon$ -ball loss where such guarantees exist in the uncorrupted case. The dependence in both fronts is now optimized when all agents are fully rational (Cohen et al., 2020, Lobel et al., 2018, Paes Leme and Schneider, 2018, Liu et al., 2021). Finally, we note that our first algorithm has quasi-polynomial running time; it is an intriguing open question to provide a polynomial-time algorithm that enjoys logarithmic guarantee when  $C \approx 0$  for the loss functions we study.

# 8

## Nearly Tight Bounds for Corruption-Robust Contextual Search

### 8.1 CHAPTER OVERVIEW

In Chapter 7, we established that contextual search is a fundamental primitive in online learning with binary feedback and studying a corruption-robust version of it (i.e., a version that is robust to arbitrary model misspecifications) is of utmost importance. As a reminder, in its standard form at each round, the learner chooses an action based on contextual information and observes only a single bit of feedback (e.g., “yes” or “no”). We slightly change the notation from the previous chapter and importantly relax our use of boldface letters, to simplify exposition. Formally, in the

realizable and noise-free version, there exists a hidden vector  $\theta^* \in \mathbb{R}^d$  with  $\|\theta^*\| \leq 1$  that the learner wishes to learn over time. Each round  $t \in [T]$  begins with the learner receiving a context  $u_t \in \mathbb{R}^d$  with  $\|u_t\| = 1$ . The learner then chooses an action  $y_t \in \mathbb{R}$ , learns the sign  $\sigma_t = \text{sgn}(\langle u_t, \theta^* \rangle - y_t) \in \{+1, -1\}$  and incurs loss  $\ell(y_t, \langle u_t, \theta^* \rangle)$ . Importantly, the learner does not observe the loss, only the binary feedback. Table 8.1 summarizes the optimal regret\* guarantees for the three loss functions in contextual search.

| <b>Loss</b>         | $\ell(y_t, y_t^*)$                             | <b>Lower Bound</b>              | <b>Upper Bound</b>   |
|---------------------|--|---------------------------------|--|
| $\varepsilon$ -ball | $\mathbb{1}\{ y_t^* - y_t  \geq \varepsilon\}$ | $\Omega(d \log(1/\varepsilon))$ | $O(d \log(1/\varepsilon))$ ( <a href="#">Lobel et al., 2018</a> )  |
| absolute            | $ y_t^* - y_t $                                | $\Omega(d)$                     | $O(d \log d)$ ( <a href="#">Liu et al., 2021</a> )                 |
| pricing             | $y_t^* - y_t \mathbb{1}\{y_t \leq y_t^*\}$     | $\Omega(d \log \log T)$         | $O(d \log \log T + d \log d)$ ( <a href="#">Liu et al., 2021</a> ) |

Table 8.1: Optimal regret guarantees for realizable contextual search.

The matching (up to  $\log d$ ) upper and lower bounds in the previous table indicate that the noiseless version of the problem is very well understood. However, a lot of questions remain when the feedback is perturbed by some type of noise (as is often the case in practical settings). In the noisy model, the target value  $y_t^* = \langle u_t, \theta^* \rangle$  is perturbed to  $\tilde{y}_t^* = \langle u_t, \theta^* \rangle + z_t$ . Chapter 8 introduced contextual search in an adversarial noise model, where the perturbations  $z_t$  are chosen by an adaptive adversary without any further assumptions. Summarizing the results presented there, we showed how to bound the total loss of the algorithm as a function of the total corruption injected in the system, as measured by  $C_0 = \sum_t \mathbb{1}\{z_t \neq 0\}$  (i.e., the number of corrupted rounds). We also provided the first algorithm with polylogarithmic regret in this setting, achieving a regret of at most

$$O(d^3 \log(1/\varepsilon) \log^2(T) + C_0 \log(T) \log(2/\varepsilon))$$

for the  $\varepsilon$ -ball loss. Unfortunately, the running time of the algorithm was quasi-polynomial in the time horizon  $T$ :  $O(\text{poly}(d, \log T)^{\text{poly} \log T})$ .

As we mentioned in Chapter 8, the natural lower bound for this problem for the  $\varepsilon$ -ball loss is  $\Omega(C_0 + d \log(1/\varepsilon))$ . The  $d \log(1/\varepsilon)$  component comes from the lower bound for the uncorrupted

---

\*Similarly to Chapter 7, we use the terms “regret” and “total loss” interchangeably.

$(C_0 = 0)$  case established in [Cohen et al. \(2020\)](#). The  $C_0$  component is necessary since if the adversary corrupts the first  $C_0$  rounds, the learner cannot avoid incurring a loss of 1 in those rounds.

In this chapter, we address several of the open questions discussed at the end of the previous chapter (see Chapter [7.7](#)). First, we provide an algorithm with the optimal regret bound of  $O(C_0 + d \log(1/\varepsilon))$  for Corrupted Contextual Search with the  $\varepsilon$ -ball loss. Second, our algorithm has running time  $O(T^d \text{poly}(d, T))$ , so it runs in polynomial time for any fixed dimension, in contrast to the algorithm in [Krishnamurthy et al. \(2021\)](#) which is quasi-polynomial even for a fixed dimension.

The third main result is an improved algorithm for the Corrupted Contextual Search problem with the absolute loss. Note that any algorithm with regret  $g(\varepsilon)$  for the  $\varepsilon$ -ball loss implies an algorithm with regret  $g(1/T)$  for the absolute loss (for  $\varepsilon = 1/T$ ), and hence, our first result directly implies an algorithm with regret  $O(C_0 + d \log T)$  for the absolute loss. We provide a second algorithm improving over this in two aspects: it runs in polynomial time  $O(\text{poly}(d, T))$  (i.e., it no longer depends exponentially on the dimension); and has regret  $O(C_1 + d \log T)$  where  $C_1 = \sum_t |z_t|$  is a more stringent definition of corruption. For example, if the adversary injects  $1/T$  corruption in every round ( $|z_t| = 1/T$ ) then  $C_0 = T$  but  $C_1 = 1$ . This is another improvement over [\(Krishnamurthy et al., 2021\)](#) where all the algorithms scale with  $C_0$ , rather than the more refined  $C_1$ .

**Technical Innovation.** The algorithms in this chapter are *fundamentally different* from the approach used in [Cohen et al. \(2020\)](#), [Lobel et al. \(2018\)](#), [Leme and Schneider \(2018\)](#), [Liu et al. \(2021\)](#), [Krishnamurthy et al. \(2021\)](#). The algorithms in those papers keep track of a “*knowledge set*”, which is the set of all possible values of  $\theta$  that are consistent with the feedback obtained. This is particularly difficult to do with corruptions and for this reason in [\(Krishnamurthy et al., 2021\)](#) we had to develop a sophisticated machinery based on convex geometry to certify that a certain region can be removed from the knowledge set. Instead, in this chapter we keep track of probability density functions over the set of possible values of  $\theta$ . The density measures to what extent a given value is consistent with the feedback obtained so far. This leads to a more *forgiving* update, that never removes a value from consideration, just decreases its weight. Surprisingly, this forgiving update, if chosen properly, can yield the very fast, logarithmic regret guarantees when  $C_0 \approx 0$ .

To translate a density into a query  $y_t$ , we introduce the notion of the  $\varepsilon$ -window-median of a dis-

tribution supported in  $\mathbb{R}$ . The 0-window-median corresponds to the usual median, i.e., a point  $m$  such that the total mass above  $m$  is equal to the total mass below  $m$ . The  $\varepsilon$ -window-median corresponds to the point  $m$  such that the total mass above  $m + \varepsilon$  is equal to the total mass below  $m - \varepsilon$ . At each round, our algorithm takes the  $\varepsilon$ -window-median with respect to a projection of the density onto the given context and uses it both to compute the query point  $y_t$  and the density update.

The second algorithm of this chapter, the one that works for the absolute loss, uses a different machinery also based on densities. We propose a new update rule inspired by the update rule used in Eldan's stochastic localization procedure ([Eldan, 2013](#)). The advantage of this update rule is that the density obtained is log-concave. This allows us to compute its centroid in polynomial time (see ([Lee and Vempala, 2021](#), Chapter 9)). It also leads to a finer control over the amount of the corruption introduced, which leads to the  $C_1$ -bound instead of  $C_0$ .

For a complete discussion around all the various different streams of literature that corruption-robust contextual search is related to, we refer the reader to Chapter [7.1.1](#).

## 8.2 MODEL AND PRELIMINARIES

Given a vector  $v \in \mathbb{R}^d$  and a real number  $r > 0$ , we define the ball of radius  $r$  around  $v$  as  $B(v, r) = \{x \in \mathbb{R}^d; \|x - v\| \leq r\}$ , where  $\|\cdot\|$  is the  $\ell_2$  norm. We use  $\text{Vol}(B(v, r))$  to denote the volume of  $B(v, r)$ , i.e.,  $\text{Vol}(B(v, r)) = \int_{B(v, r)} 1 dx$ . Let  $\ell : \mathbb{R}^2 \rightarrow \mathbb{R}$  denote any loss function. Here, we consider two losses: (i) the  $\varepsilon$ -ball loss  $\ell(y, y^*) = \mathbb{1}\{|y - y^*| \geq \varepsilon\}$ , which penalizes each query  $y$  that is far from the target  $y^*$  by at least  $\varepsilon$ ; (ii) the absolute loss  $\ell(y, y^*) = |y - y^*|$ , which penalizes each query proportionally to how far it is from the target.

**Setting.** The corrupted contextual search setting corresponds to a repeated interaction between a learner and an adversary over  $T$  rounds. The adversary initially chooses a vector  $\theta^* \in B(0, 1)$  that is hidden from the learner. In each round  $t \in [T]$ :

- The adversary chooses a context  $u_t \in \mathbb{R}^d$ ,  $\|u_t\| = 1$  and reveals it to the learner.
- The adversary selects a corruption level  $z_t \in [-1, 1]$  (hidden from the learner).
- The learner queries  $y_t \in [-1, 1]$ .

- The learner receives feedback  $\sigma_t = \text{sgn}(y_t^* - y_t) \in \{-1, +1\}$  where  $y_t^* = \langle u_t, \theta^* \rangle + z_t$ .
- The learner incurs (but does not observe) loss  $\ell(y_t, y_t^*)$ .

Note that the decision of the agent to corrupt the present round is identical in process to the one mentioned in Chapter 7. That said, we depart from the corruptions model of that chapter by considering two different measures of the total amount of corruption:

$$C_0 = \sum_{t \in [T]} \mathbb{1}\{z_t \neq 0\} \quad \& \quad C_1 = \sum_{t \in [T]} |z_t|$$

Our goal is to bound the total regret  $\text{Regret} = \sum_{t \in [T]} \ell(y_t, y_t^*)$ . We do not impose any restriction on any specific corruption levels  $z_t$ . Instead, our eventual regret bounds are functions of the total amount of corruption ( $C_0$  or  $C_1$ ) added over the game. Importantly, our algorithms are completely agnostic to the level of corruption introduced by the adversary. The quantities  $C_0$  and  $C_1$  are used in the analysis but are not used by the algorithm.

**Densities.** A function  $f : \mathbb{R}^d \rightarrow \mathbb{R}_+$  is a density function if it integrates to 1, i.e.,  $\int_{\mathbb{R}^d} f(x)dx = 1$ . We say that a random variable  $Z$  is drawn from a probability distribution with density  $f$ , if for every measurable set  $S \subseteq \mathbb{R}^d$ , it holds that  $\mathbb{P}[Z \in S] = \int_S f(x)dx$ . Given a measurable set  $S$ , we refer to the function  $f(x) = \frac{\mathbb{1}\{x \in S\}}{\int_S 1 dx}$  as the uniform density over  $S$ . To simplify notation, we write  $\int_B f_t(x)dx$  instead of  $\int_{B(0,1)} f_t(x)dx$ .

**Log-concave densities.** We give a brief introduction to log-concave densities, which are used in Section 8.4. For a more complete introduction, see the book by [Lee and Vempala \(2021\)](#) or the survey by [Lovász and Vempala \(2007\)](#).

**Definition 8.1** (Log-Concave Functions). *A function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  is called log-concave if it is of the form  $f(x) = \exp(-g(x))$  for some convex function  $g : \mathbb{R}^d \rightarrow \mathbb{R}$ . If  $\int f(x)dx = 1$  we say that  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  is a log-concave density function.*

Two important examples of log-concave densities are the Gaussian density (where  $g(x) = \|x\|^2$ ) and the uniform over a convex set  $K \subseteq \mathbb{R}^d$  (where  $g(x) = 0$  for  $x \in K$  and  $g(x) = \infty$  for  $x \notin K$ ).

We denote with  $\text{cg}(f)$  the centroid of  $f$ :

$$\text{cg}(f) = \frac{\int_{\mathbb{R}^d} x f(x) dx}{\int_{\mathbb{R}^d} f(x) dx}$$

Note that if we take the uniform density over a convex set, this corresponds to the usual notion of the centroid of a convex set.

**A Note on Normalization.** In this chapter, we normalize the contexts such that  $\|u_t\| = 1$ , while previous contextual search papers instead assume  $\|u_t\| \leq 1$ . This is done only for simplicity of presentation. If we had  $\|u_t\| \leq 1$  we can feed  $\hat{u}_t = u_t/\|u_t\|$  to our algorithm, obtain a query  $\hat{y}_t$  based on  $\hat{u}_t$  and convert it to  $y_t = \hat{y}_t \|u_t\|$ .

### 8.3 A $O(C_0 + d \log(1/\varepsilon))$ ALGORITHM FOR THE $\varepsilon$ -BALL LOSS

#### 8.3.1 FIRST ATTEMPT: USING THE STANDARD MEDIAN

Our algorithm by keeping track of a density function  $f_t : B(0, 1) \rightarrow \mathbb{R}$  that evolves from round to round. Initially, we set  $f_1$  to be the uniform density over  $B(0, 1)$ , i.e.,  $f_1(x) = 1/\text{Vol}(B(0, 1))$  for all  $x \in B(0, 1)$ . We start by describing a natural algorithm that is not ultimately the algorithm we analyze, but will be useful for providing intuition. This algorithm is as follows: once the context  $u_t$  arrives, we compute the median  $y_t$  of  $f_t$  “in the direction  $u_t$ ”.

**Definition 8.2** (Median of a Distribution). *There are two equivalent ways to define the median of a distribution in a certain direction.*

1. Define a random variable  $Z = \langle X, u_t \rangle$ , where  $X$  is drawn from a density  $f_t$ . Then,  $y_t$  is called the median of  $Z$  if:  $\mathbb{P}[Z \geq y_t] = \mathbb{P}[Z \leq y_t]$ .
2.  $y_t \in \mathbb{R}$  is the median of distribution  $f$  if:  $\int f_t(x) \mathbf{1}\{\langle u_t, x \rangle \geq y_t\} dx = \int f_t(x) \mathbf{1}\{\langle u_t, x \rangle \leq y_t\} dx$ .

Note that since all the distributions that we work with in this chapter are derived from continuous density functions, they do not have point masses and hence, the median (and later the  $\varepsilon$ -window-median) is always well defined.

After we query  $y_t$ , we receive the feedback of whether  $y_t^* = \langle u_t, \theta^* \rangle + z_t$  is larger or smaller than  $y_t$ . We do not know the amount of corruption added, but if we believe that it is more likely that this feedback is uncorrupted than corrupted, then we can try to increase the density whenever  $\sigma_t(\langle u_t, x \rangle - y_t) \geq 0$ . For example, we could define:

$$f_{t+1}(x) = \begin{cases} 3/2 \cdot f_t(x), & \text{if } \sigma_t(\langle u_t, x \rangle - y_t) \geq 0 \\ 1/2 \cdot f_t(x), & \text{if } \sigma_t(\langle u_t, x \rangle - y_t) < 0 \end{cases}$$

Note that since  $y_t$  is chosen to be median, then  $f_{t+1}$  is still a density.

**Lemma 8.1.** *Function  $f_t(\cdot)$  is a valid probability density function for all rounds  $t$ .*

*Proof.* We proceed with induction. For the base case and by the definition of  $f_1(\cdot)$  to be a uniform density, the lemma holds. Assume now that  $f_t(\cdot)$  is a valid probability density for some round  $t = n$ , i.e.,  $\int_{B(0,1)} f_n(x) dx = 1$ . Then, for round  $t + 1 = n + 1$ :

$$\begin{aligned} \int_B f_{t+1}(x) dx &= \frac{3}{2} \int_B f_t(x) \mathbb{1}\{\sigma_t(\langle u_t, x \rangle - y_t) \geq 0\} dx + \frac{1}{2} \int_B f_t(x) \mathbb{1}\{\sigma_t(\langle u_t, x \rangle - y_t) < 0\} dx \\ &= \int_B f_t(x) \mathbb{1}\{\sigma_t(\langle u_t, x \rangle - y_t) \geq 0\} dx + \frac{1}{2} \int_B f_t(x) dx && (\text{grouping terms}) \\ &= \int_B f_t(x) \mathbb{1}\{\sigma_t(\langle u_t, x \rangle - y_t) \geq 0\} dx + \frac{1}{2} && (\text{inductive hypothesis}) \\ &= \frac{1}{2} \cdot 2 \cdot \int_B f_t(x) \mathbb{1}\{\sigma_t(\langle u_t, x \rangle - y_t) \geq 0\} dx + \frac{1}{2} \\ &= \frac{1}{2} \int_B f_t(x) \mathbb{1}\{\sigma_t(\langle u_t, x \rangle - y_t) \geq 0\} dx + \frac{1}{2} \int_B f_t(x) \mathbb{1}\{\sigma_t(\langle u_t, x \rangle - y_t) < 0\} dx + \frac{1}{2} \\ &= \frac{1}{2} + \frac{1}{2} = 1 \end{aligned}$$

where the penultimate equality is due to the definition of  $y_t$  being the median of distribution  $f_t(\cdot)$  (Definition 8.2). ■

Ideally, we would like the mass of the density around  $\theta^*$  to increase in all uncorrupted rounds. With this update rule, however, this is impossible to argue. To see why, if the hyperplane  $\{x \in \mathbb{R}^d; \langle u_t, x \rangle = y_t\}$  is far from  $\theta^*$  then the total density in a ball  $B(\theta^*, \varepsilon)$  will increase. However, if the hyperplane intersects the ball  $B(\theta^*, \varepsilon)$ , then some part of its density will increase and some will

decrease. Since the density is non-uniform in the ball, we cannot argue that it will increase in good rounds, i.e., rounds where  $y_t \in B(\theta^*, \varepsilon)$ .

### 8.3.2 THE $\varepsilon$ -WINDOW MEDIAN ALGORITHM

To address the issue above, we define a new notion, which we call the  $\varepsilon$ -window median.

**Definition 8.3** ( $\varepsilon$ -Window Median). *Given a random variable  $Z$  taking values in  $\mathbb{R}$  we say that an  $\varepsilon$ -window median of  $Z$  is a value  $y$  such that:  $\mathbb{P}[Z \leq y - \varepsilon/2] = \mathbb{P}[Z \geq y + \varepsilon/2]$ .*

We can also define the  $\varepsilon$ -window median for a density  $f_t(\cdot)$  as follows.

**Definition 8.4** ( $\varepsilon$ -Window Median for Densities). *Given a density  $f_t$  and a direction  $u_t \in B(0, 1)$ , we say that the  $\varepsilon$ -window median of  $f_t$  in the direction  $u_t$  is the  $\varepsilon$ -window median of a variable  $Z = \langle u_t, X \rangle$ , where  $X$  is drawn from a distribution with density  $f_t$ . Equivalently, this is the value  $y_t \in \mathbb{R}$  such that:*

$$\int f_t(x) \mathbb{1}\{\langle u_t, x \rangle \geq y_t + \varepsilon/2\} dx = \int f_t(x) \mathbb{1}\{\langle u_t, x \rangle \leq y_t - \varepsilon/2\} dx$$

---

#### ALGORITHM 8.1: $\varepsilon$ -WINDOW MEDIAN ALGORITHM

---

- 1 Initialize  $f_1(x)$  to be the uniform density over  $B(0, 1)$ .
- 2 **for** rounds  $t \in [T]$  **do**
- 3     Observe context  $u_t$ .
- 4     Query  $\varepsilon$ -window median of  $f_t$ :  $y_t$ .
- 5     Receive feedback  $\sigma_t$  and update the density as:

$$f_{t+1}(x) = \begin{cases} 3/2 \cdot f_t(x), & \text{if } \sigma_t(\langle u_t, x \rangle - y_t) \geq \varepsilon/2 \\ 1 \cdot f_t(x), & \text{if } -\varepsilon/2 \leq \sigma_t(\langle u_t, x \rangle - y_t) \leq \varepsilon/2 \\ 1/2 \cdot f_t(x), & \text{if } \sigma_t(\langle u_t, x \rangle - y_t) \leq -\varepsilon/2 \end{cases}$$


---

We first prove that  $f_{t+1}(x)$  is a valid density.

**Lemma 8.2.** *Function  $f_t(\cdot)$  is a valid probability density function for all rounds  $t$ .*

*Proof.* We prove this lemma by induction. For the base case, note that by definition the lemma holds for  $t = 1$ , since  $f_1(x)$  is the uniform density over  $B(0, 1)$ . Assume now that  $f_t(x)$  is a density

for some  $t = n$ , i.e.,  $\int_{\mathbb{B}} f_t(x)dx = 1$ . Then, for round  $t + 1 = n + 1$  we define the following sets:

$$\begin{aligned} U_+ &= \{x \in \mathbb{B}(0, 1) : \sigma_t(\langle u_t, x \rangle - y_t) \geq \varepsilon/2\} \\ U_0 &= \{x \in \mathbb{B}(0, 1) : -\varepsilon/2 \leq \sigma_t(\langle u_t, x \rangle - y_t) \leq \varepsilon/2\} \\ U_- &= \{x \in \mathbb{B}(0, 1) : \sigma_t(\langle u_t, x \rangle - y_t) \leq -\varepsilon/2\} \end{aligned}$$

As for  $f_{t+1}(x)$  we have:

$$\begin{aligned} \int_{\mathbb{B}} f_{t+1}(x)dx &= \frac{3}{2} \int_{U_+} f_t(x)dx + \int_{U_0} f_t(x)dx + \frac{1}{2} \int_{U_-} f_t(x)dx \\ &= \int_{U_+} f_t(x)dx + \frac{1}{2} \int_{U_0} f_t(x)dx + \frac{1}{2} \int_{\mathbb{B}} f_t(x)dx && \text{(grouping terms)} \\ &= \int_{U_+} f_t(x)dx + \frac{1}{2} \int_{U_0} f_t(x)dx + \frac{1}{2} && \text{(inductive hypothesis)} \\ &= \frac{1}{2} + \frac{1}{2} = 1 \end{aligned}$$

where the penultimate inequality is due to the following property which is direct from the definition of the  $\varepsilon$ -window median (Definition 8.3):

$$\int_{U_+} f_t(x)dx = \int_{U_-} f_t(x)dx = \frac{1}{2} - \frac{1}{2} \int_{U_0} f_t(x)dx$$

This concludes our proof. ■

We are now left to bound the regret of Algorithm 8.1.

**Theorem 8.1: Regret of  $\varepsilon$ -Window Median**

The regret of the  $\varepsilon$ -Window Median Algorithm is  $O(C_0 + d \log(1/\varepsilon))$ .

*Proof.* We define a potential function:

$$\Phi_t = \int_{\mathbb{B}(\theta^*, \varepsilon/2)} f_t(x)dx$$

For each round  $t$ , we distinguish the following three cases.

For *Case 1*, if round  $t$  is a corrupted round, then the potential decreases by at most a factor of 2, i.e.,  $\Phi_{t+1} \geq \Phi_t/2$ . This is because regardless of the feedback  $\sigma_t$ :  $f_{t+1}(x) \geq (1/2)f_t(x)$  for all  $x$ . Note that there are at most  $C_0$  such corrupted rounds.

For *Case 2*, assume that round  $t$  is an uncorrupted round in which we pick up a loss of 1. In this case, note that the potential increases by a factor of  $3/2$ , i.e.,  $\Phi_{t+1} = (3/2)\Phi_t$ . To see this, note that since we pick up a loss of 1, then by definition the distance from  $\theta^*$  to the hyperplane  $\{x; \langle u_t, x \rangle = y_t\}$  has to be at least  $\varepsilon$ . As a consequence, the ball  $B(\theta^*, \varepsilon/2)$  has to be inside the halfspace  $\{x; \sigma_t(\langle u_t, x \rangle - y_t) \geq \varepsilon/2\}$  and therefore,  $f_{t+1}(x) = (3/2)f_t(x)$  for all  $x \in B(\theta^*, \varepsilon/2)$ . We denote by  $L$  the total number of such uncorrupted rounds. Note that this  $L$  corresponds also the total loss suffered through these rounds.

For *Case 3*, assume that  $t$  is an uncorrupted round in which we incur a loss of 0. In that case, observe that the potential does not decrease, i.e.,  $\Phi_{t+1} \geq \Phi_t$ . Indeed, since the round is uncorrupted, it must be the case that  $\sigma_t(\langle u_t, \theta^* \rangle - y_t) \geq 0$ . Therefore, for all  $x \in B(\theta^*, \varepsilon/2)$  we must have:  $\sigma_t(\langle u_t, x \rangle - y_t) \geq -\varepsilon/2$ . Hence,  $f_{t+1}(x) \geq f_t(x)$  for all  $x \in B(\theta^*, \varepsilon/2)$ .

Putting it all together and telescoping for  $\Phi_t$  we obtain:

$$\Phi_{T+1} \geq \Phi_1 \cdot \left(\frac{1}{2}\right)^{C_0} \cdot \left(\frac{3}{2}\right)^L$$

Since  $f_t$  is always a density (Lemma 8.2), we have that  $\Phi_{T+1} \leq 1$ . So, taking logarithms for both sides of the above equation, we get:

$$0 \geq \log \Phi_1 + C_0 \log \frac{1}{2} + L \log \frac{3}{2}$$

Reorganizing the terms:

$$L \leq O(C_0 - \log(\text{Vol}(B(\theta^*, \varepsilon)))) = O(C_0 + d \log(1/\varepsilon))$$

Finally, note that the regret from corrupted rounds is at most  $C_0$  and the regret from uncorrupted rounds is  $L$ , so  $\text{Regret} \leq C_0 + L \leq O(C_0 + d \log(1/\varepsilon))$ . ■

**Relation to Multiplicative Weights.** Like the traditional MWU algorithm, we keep a weight over the set of candidate solutions and update it multiplicatively. However, it is worth pointing out some important key differences. First, unlike in experts' or bandits' settings, we do not get to observe the loss (not even an unbiased estimator thereof). We can only observe binary feedback, so it is impossible to update proportionally to the loss in each round. Second, we do not choose an action proportionally to the weights like MWU or EXP3. Instead, we use the  $\varepsilon$ -window-median.

We conclude this section by discussing the running time of the  $\varepsilon$ -window-median algorithm.

**Lemma 8.3** (Running Time). *Algorithm 8.1 has runtime  $O(T^d \cdot \text{poly}(d, T))$ .*

*Proof.* The running time in each step is dominated by the computation of the  $\varepsilon$ -window median. Given an oracle that for each  $u \in \mathbb{R}^d$  and  $y \in \mathbb{R}$  returns the integral  $\int f_t(x) \mathbf{1}\{\langle u, x \rangle \leq y\} dx$ , we can use binary search to determine the  $\varepsilon$ -window median. Observe that the function:

$$\psi(y) = \frac{\int f_t(x) \mathbf{1}\{\langle u, x \rangle \leq y - \varepsilon/2\} dx}{\int f_t(x) \mathbf{1}\{\langle u, x \rangle \geq y + \varepsilon/2\} dx}$$

is monotonically increasing and computing the  $\varepsilon$ -window median is equivalent to finding a value of  $y$  such that  $\psi(y) = 1$ . Note also that the analysis does not require us to query the  $\varepsilon$ -window median exactly. Rather, any point  $y$  with  $\psi(y) \in [1 - \varepsilon, 1 + \varepsilon]$  would lead to the same bound with a change only in the constants.

Next, we discuss how to design an oracle to compute the integral  $\int f_t(x) \mathbf{1}\{\langle u, x \rangle \leq y\} dx$ . Note that  $f_t$  is piecewise constant, where each piece is one the regions in space determined by the hyperplanes  $\{x : \langle u_t, x \rangle = y_t\}$ . The maximum number of regions created by  $T$  hyperplanes in  $\mathbb{R}^d$  is given by the Whitney number, which is at most  $O(T^d)$  ([Stanley et al. \(2004\)](#)). Keeping track of each of these regions explicitly leads to an  $O(T^d \text{poly}(d, T))$  algorithm for computing the integral. ■

## 8.4 AN EFFICIENT $O(C_1 + d \log T)$ ALGORITHM FOR THE ABSOLUTE LOSS

The  $\varepsilon$ -Window Median Algorithm with  $\varepsilon = 1/T$  has regret  $O(C_0 + d \log T)$  for the absolute loss<sup>†</sup>, with a running time that is polynomial in  $T$  but exponential in  $d$ . In this section, we provide an alternative algorithm that has polynomial runtime in both  $T$  and  $d$  and has regret  $O(C_1 + d \log T)$ , i.e., using the more stringent measure of corruption  $C_1 \leq C_0$ . We remark that this is the first result of corruption-robust contextual search against  $C_1$ -type corruptions.

The main idea is to keep a more “structured” density function  $f_t$ . Formally, we make sure that  $f_t$  is a *log-concave density*. We then update it in such a way that the density  $f_t$  remains log-concave throughout the algorithm, which enables efficient sampling from it. The algorithm we use is formally defined below, where we have slightly abused the notation and defined  $\text{cg}_t = \text{cg}(f_t)$ .

---

### ALGORITHM 8.2: Log-Concave Density Algorithm

---

- 1 Initialize  $f_1(x)$  to be the uniform density over  $B(0, 1)$ .
- 2 **for** rounds  $t \in [T]$  **do**
- 3     Observe context  $u_t$ .
- 4     Compute the centroid of distribution  $f_t$ , i.e.,  $\text{cg}_t = \int_B x f_t(x) dx$ .
- 5     Query the projection of the centroid in the direction  $u_t$ :  $y_t = \langle u_t, \text{cg}_t \rangle$ .
- 6     Receive feedback  $\sigma_t$  and update the density as:

$$f_{t+1}(x) = f_t(x) \cdot \left(1 + \frac{1}{3} \cdot \sigma_t \cdot \langle u_t, x - \text{cg}_t \rangle\right) \quad (8.4.1)$$


---

We first argue that the  $f_t$  remains a log-concave density throughout  $T$  rounds.

**Lemma 8.4.** *The function  $f_t$  maintained by Algorithm 8.2 is a log-concave density for all  $t \in [T]$ .*

*Proof.* We proceed with induction. For the base case, note that the lemma holds by definition, since  $f_1$  is the uniform density. Assume now that the lemma holds for some  $t = n$ , i.e.,  $f_t$  is log-concave and  $\int_B f_t(x) dx = 1$ . We now focus on  $f_{t+1}$ . First, note that  $f_{t+1}$  is non-negative, since

$$\left| \frac{1}{3} \cdot \sigma_t \langle u_t, x - \text{cg}_t \rangle \right| \leq \frac{1}{3} (\|x\| + \|\text{cg}_t\|) < 1$$

---

<sup>†</sup>This is because every time the  $(1/T)$ -ball loss is 1, the absolute loss is at most 1. Every time the  $(1/T)$ -ball loss is 0, the absolute loss is at most  $1/T$ . Hence, the absolute loss is at most 1 plus the  $(1/T)$ -ball loss.

and so

$$1 + \frac{1}{3}\sigma_t \langle u_t, x - cg_t \rangle > 0.$$

To see that it integrates to 1:

$$\begin{aligned} \int_{B(0,1)} f_{t+1}(x) dx &= \int_{B(0,1)} f_t(x) dx + \frac{1}{3} \cdot \sigma_t \cdot \left\langle \int_{B(0,1)} x f_t(x) dx - cg_t, u_t \right\rangle \\ &= 1 + \frac{1}{3} \cdot \sigma_t \cdot \left\langle \int_{B(0,1)} x f_t(x) dx - cg_t, u_t \right\rangle && \text{(inductive hypothesis)} \\ &= 1 + 0 && \text{(definition of } cg_t) \end{aligned}$$

We are left to show that  $f_t$  is a log-concave function for all rounds  $g$ . We use again induction. By definition, the uniform density is log-concave (base case). Assume now that  $f_t$  is log-concave for some round  $t = n$ , and let us rewrite it as  $f_t(x) = \exp(-g_t(x))$  for some convex function  $g_t$  (Definition 8.1). Then, for round  $t + 1$ , the density can be written as:

$$f_{t+1}(x) = \exp(-g_{t+1}(x)), \quad \text{for} \quad g_{t+1}(x) = g_t(x) - \log \left( 1 + \frac{1}{3}\sigma_t \langle u_t, x - cg_t \rangle \right)$$

Note that  $g_{t+1}$  is a convex function, since it is a sum of convex functions. As a result, by Definition 8.1,  $f_{t+1}$  is a log-concave function. ■

### Theorem 8.2: Regret of the Log-Concave Density Algorithm

The regret of the Log-Concave Density Algorithm is  $O(C_1 + d \log T)$ .

*Proof.* We define a potential corresponding to the total mass around  $\theta^*$  and argue that loss leads to concentration of measure:

$$\Phi_t = \int_{B(\theta^*, 1/T)} f_t(x) dx$$

Let  $\tilde{y}_t^*$  be the corrupted target  $\tilde{y}_t^* = \langle u_t, \theta^* \rangle + z_t$  and  $y_t^*$  be the uncorrupted target value  $y_t^* = \langle u_t, \theta^* \rangle$ . Similarly, let  $\ell_t$  and  $\tilde{\ell}_t$  be the losses with respect to  $y^*$  and  $\tilde{y}^*$  respectively, i.e.,  $\ell_t = |y_t^* - y_t|$  and  $\tilde{\ell}_t = |\tilde{y}_t^* - y_t|$ . Finally, let  $\sigma_t$  and  $\tilde{\sigma}_t$  denote the feedback obtained in the case where the round is uncorrupted and corrupted respectively, i.e.,  $\sigma_t = \text{sgn}(y_t^* - y_t)$  and  $\tilde{\sigma}_t = \text{sgn}(\tilde{y}_t^* - y_t)$ . We analyze three cases for how the potential changes.

For *Case 1*, if  $\ell_t \leq 1/T$ , then the hyperplane  $\{x; \langle u_t, x - c\mathbf{g}_t \rangle = 0\}$  passes through the ball  $B(\theta^*, 1/T)$ . This means that for all  $x \in B(\theta^*, 1/T)$  we have that

$$\begin{aligned} |\langle u_t, x - c\mathbf{g}_t \rangle| &\leq |\langle u_t, x - \theta^* + \theta^* - c\mathbf{g}_t \rangle| \\ &\leq |\langle u_t, x - \theta^* \rangle| + |\langle u_t, \theta^* - c\mathbf{g}_t \rangle| && \text{(triangle inequality)} \\ &\leq \frac{1}{T} + |\langle u_t, \theta^* - c\mathbf{g}_t \rangle| && (x \in B(\theta^*, 1/T)) \\ &\leq \frac{2}{T} && (\ell_t \leq 1/T, \text{ by assumption for Case 1}) \end{aligned}$$

This means that regardless of the feedback  $\sigma_t$  at this round, from Equation 8.4.1 it holds that  $f_{t+1}(x) \geq f_t(x)(1 - \frac{2}{3T})$ ,  $\forall x$ . In particular,

$$\Phi_{t+1} \geq \left(1 - \frac{2}{3T}\right) \cdot \Phi_t \quad (8.4.2)$$

Let us denote by  $S_1$  the set of rounds  $t \in [T]$  for which case 1 holds, i.e., for which  $\ell_t \leq \frac{1}{T}$ .

For *Case 2*, assume that  $\ell_t > 1/T$  and  $\sigma_t = \tilde{\sigma}_t$ , i.e., the corruption does not change the feedback. Then, for all  $x \in B(\theta^*, 1/T)$  we have that:

$$\begin{aligned} \sigma_t \cdot \langle u_t, x - c\mathbf{g}_t \rangle &= \sigma_t \cdot \langle u_t, x + \theta^* - \theta^* - c\mathbf{g}_t \rangle = \sigma_t \cdot \langle u_t, \theta^* - c\mathbf{g}_t \rangle + \sigma_t \cdot \langle u_t, x - \theta^* \rangle \\ &\geq \sigma_t \cdot \langle u_t, \theta^* - c\mathbf{g}_t \rangle - |\langle u_t, x - \theta^* \rangle| && (\sigma_t \in \{-1, +1\}) \\ &\geq \sigma_t \cdot \langle u_t, \theta^* - c\mathbf{g}_t \rangle - \frac{1}{T} && (x \in B(\theta^*, 1/T)) \\ &= \ell_t - \frac{1}{T} \end{aligned}$$

This means from the update rule in Equation (8.4.1) that for  $x \in B(\theta^*, 1/T)$ :

$$f_{t+1}(x) \geq f_t(x) \left(1 + \frac{1}{3} \left(\ell_t - \frac{1}{T}\right)\right).$$

As a result, the potential becomes:

$$\Phi_{t+1} \geq \Phi_t \cdot \left(1 + \frac{1}{3} \left(\ell_t - \frac{1}{T}\right)\right) \geq \Phi_t \cdot \exp\left(\frac{\ln 2}{3} \cdot \left(\ell_t - \frac{1}{T}\right)\right) \quad (8.4.3)$$

We denote by  $S_2$  the set of rounds  $t \in [T]$  for which case 2 holds, i.e.,  $\ell_t > 1/T$  and  $\sigma_t = \tilde{\sigma}_t$ .

For Case 3, assume that  $\ell_t > 1/T$  but  $\sigma_t = -\tilde{\sigma}_t$ , i.e., the corruption flips the feedback. So, it must be that  $|z_t| \geq \ell_t$ . Then, for all  $x \in B(\theta^*, 1/T)$  we have that:

$$\begin{aligned} \tilde{\sigma}_t \cdot \langle u_t, x - cg_t \rangle &= -\sigma_t \cdot \langle u_t, x - cg_t \rangle \\ &\geq -\sigma_t \cdot \langle u_t, \theta^* - cg_t \rangle - |\langle \theta^* - x \rangle| \\ &\geq -\sigma_t \cdot \langle u_t, \theta^* - cg_t \rangle - \frac{1}{T} && (x \in B(\theta^*, 1/T)) \\ &= -\ell_t - \frac{1}{T} \\ &\geq -|z_t| - \frac{1}{T} && (|z_t| \geq \ell_t) \end{aligned}$$

Plugging this to Equation (8.4.1), this means that the potential becomes:

$$\Phi_{t+1} \geq \Phi_t \cdot \left(1 - \frac{1}{3} \cdot \left(|z_t| + \frac{1}{T}\right)\right) \geq \Phi_t \cdot \exp\left(-\frac{\alpha}{3} \left(|z_t| + \frac{1}{T}\right)\right) \quad (8.4.4)$$

for constant  $\alpha = \frac{3}{2} \ln 3 > 1$  such that  $1 - v \geq \exp(-\alpha v)$  for  $v \in [0, 2/3]$ . We denote by  $S_3$  the set of rounds  $t \in [T]$  for which case 3 holds.

Putting everything together and telescoping the definition of  $\Phi_t$  the potential in the end is:

$$\Phi_{T+1} \geq \Phi_1 \cdot \left(1 - \frac{2}{3T}\right)^{|S_1|} \cdot \exp\left(\frac{\ln 2}{3} \cdot \sum_{t \in S_2} \ell_t - \frac{\alpha}{3} \cdot \sum_{t \in S_3} |z_t| - 2\right) \quad (8.4.5)$$

where the  $-2$  term at the end collects the  $1/T$  terms in  $S_2$  and  $S_3$ . We can further relax Equation (8.4.5) as follows:

$$\begin{aligned} \Phi_{T+1} &\geq \Phi_1 \cdot \left(1 - \frac{2}{3T}\right)^T \cdot \exp\left(\frac{\ln 2}{3} \cdot \sum_{t \in S_2} \ell_t - \frac{\alpha}{3} \cdot \sum_{t \in S_3} |z_t| - 2\right) && ((1 - \frac{2}{3T})^{|S_1|} \geq (1 - \frac{2}{3T})^T) \\ &\geq \Phi_1 \cdot \frac{1}{3} \cdot \exp\left(\frac{\ln 2}{3} \cdot \sum_{t \in S_2} \ell_t - \frac{\alpha}{3} \cdot \sum_{t \in S_3} |z_t| - 2\right) \end{aligned} \quad (8.4.6)$$

where the last inequality is due to the fact that  $(1 - \frac{2}{3T})^T \geq \frac{1}{3}$ . Since  $f_t$  is a valid density function throughout the  $T$  rounds (Lemma 8.4), we have that  $\Phi_{T+1} \leq 1$ . Hence, taking logarithms on both

sides of Equation (8.4.6) and using the fact that  $\Phi_1 = \text{Vol}(\mathcal{B}(\theta^*, 1/T))$  we have that:

$$\sum_{t \in S_2} \ell_t \leq O \left( d \log T + \sum_{t \in S_3} |z_t| \right) \quad (8.4.7)$$

We now turn our attention to the total loss picked up during the rounds that belong in  $S_3$ :

$$\sum_{t \in S_3} \tilde{\ell}_t = \sum_{t \in S_3} |\tilde{y}_t - y_t + z_t| \leq \sum_{t \in S_3} |z_t| \quad (8.4.8)$$

where the last inequality is due to the fact that  $z_t$  and  $\tilde{y}_t - y_t$  have opposite signs, since  $\sigma_t = -\tilde{\sigma}_t$  for  $t \in S_3$ . We can now bound the total regret as follows:

$$\begin{aligned} \text{Regret} &= \sum_{t \in S_1} \tilde{\ell}_t + \sum_{t \in S_2} \tilde{\ell}_t + \sum_{t \in S_3} \tilde{\ell}_t \leq \sum_{t \in A} \left( \frac{1}{T} + |z_t| \right) + O \left( d \log T + \sum_{t \in C} |z_t| \right) + \sum_{t \in B} |z_t| + \sum_{t \in C} |z_t| \\ &\leq \sum_{t \in S_1} (\ell_t + |z_t|) + \sum_{t \in S_2} (\ell_t + |z_t|) + \sum_{t \in S_3} \tilde{\ell}_t \quad (\tilde{\ell}_t \leq \ell_t + |z_t| \text{ for } t \in S_1 \cup S_2) \\ &\leq \sum_{t \in S_1} \frac{1}{T} + \sum_{t \in S_1 \cup S_2} |z_t| + \sum_{t \in S_2} \ell_t + \sum_{t \in S_3} \tilde{\ell}_t \quad (\ell_t \leq 1/T \text{ for } t \in S_1) \\ &\leq 1 + \sum_{t \in S_1 \cup S_2} |z_t| + O \left( d \log T + \sum_{t \in S_3} |z_t| \right) + \sum_{t \in S_3} \tilde{\ell}_t \quad (\text{Eq. (8.4.7)}) \\ &\leq 1 + \sum_{t \in [T]} |z_t| + O \left( \sum_{t \in S_3} |z_t| \right) + O(d \log T) \quad (\text{Eq. (8.4.8)}) \\ &\leq O(C_1 + d \log T) \quad (C_1 = \sum_{t \in [T]} |z_t|) \end{aligned}$$

This concludes our proof. ■

**COMPARISON TO ONLINE GRADIENT DESCENT.** The contextual search problem with a absolute loss function can be treated as a special case of online convex minimization with functions  $h_t(\theta) = |\langle u_t, \theta \rangle - y_t^*|$  since its gradient can be computed just using the feedback:  $\nabla h_t(\theta) = \sigma_t u_t$ . Online gradient descent, therefore, leads to a corruption-tolerant sublinear bound for this problem. The bound however is  $O(\sqrt{T} + C_1)$  instead of logarithmic. We refer to Chapter 7.6 for a discussion of gradient descent in contextual search.

**RUNNING TIME.** The computationally non-trivial step in the Log-Concave Density Update algorithm is the computation of the centroid. This problem boils down to integrating a log-concave function, since its  $i$ -th component is  $\int x_i f_t(x) dx = \int \exp(\log x_i + \log f_t(x)) dx$ . [Applegate and Kannan \(1991\)](#) were the first to show that an  $\varepsilon$ -additive approximation of the integral of a log-concave function over a bounded domain can be obtained in time  $O(\text{poly}(d, 1/\varepsilon))$ . See the book by [Lee and Vempala \(2021\)](#) for algorithms with an improved running time.

The proof of Theorem 8.2 can be adapted to work with an approximate centroid as follows. Let  $\tilde{\mathbf{cg}}_t$  be an approximate centroid of  $f_t(x)$ , i.e., the point  $\tilde{\mathbf{cg}}_t$  such that:

$$\left\| \tilde{\mathbf{cg}}_t - \frac{\int_{B(0,1)} x f_t(x) dx}{\int_{B(0,1)} f_t(x) dx} \right\| \leq \delta. \quad (8.4.9)$$

Then, the update defined in Equation (8.4.1) no longer keeps  $f_t$  a density, but it still keeps it an approximate density as follows:

$$\int_{B(0,1)} f_{t+1}(x) dx = \int_{B(0,1)} f_t(x) dx \cdot \left( 1 + \frac{\sigma_t}{3} \cdot \left\langle \frac{\int_{B(0,1)} x f_t(x) dx}{\int_{B(0,1)} f_t(x) dx} - \tilde{\mathbf{cg}}_t, u_t \right\rangle \right)$$

Note that this is indeed an “approximate density”, since:

$$1 - \frac{\delta}{3} \leq \frac{\int_{B(0,1)} f_{t+1}(x) dx}{\int_{B(0,1)} f_t(x) dx} \leq 1 + \frac{\delta}{3} \quad (8.4.10)$$

Setting  $\delta = 1/T$  in Equation (8.4.10) and telescoping for  $\int_{B(0,1)} f_t(x) dx$  we get:

$$\frac{1}{e} \leq \int_{B(0,1)} f_{t+1}(x) dx \leq e \quad (8.4.11)$$

Finally, the only thing that these derivations change with respect to the regret proof of Theorem 8.2 is that instead of having  $\Phi_{T+1} \leq 1$  now we can only guarantee that  $\Phi_{T+1} \leq e$ . This only affects the constants in the final regret bound.

## 8.5 DISCUSSION AND OPEN QUESTIONS

In this chapter, we continued our study on corruption-robust contextual search and we obtained significantly optimized regret and runtime results. Concretely, we provided the  $\varepsilon$ -window median algorithm with optimal regret  $O(C_0 + d \log(1/\varepsilon))$  for the  $\varepsilon$ -ball loss. We also provided an algorithm based on log-concave density functions with nearly optimal regret  $O(C_1 + d \log T)$  for the absolute loss while running in polynomial time. This is the first polynomial time algorithm for corruption-robust contextual search. Additionally, it is the first algorithm to obtain regret bounds with respect to  $C_1$ , which is a harder benchmark for corruptions than  $C_0$ .

Although this chapter addressed many open questions from Chapter 7, it also gave rise to many more avenues for future work. We highlight the three biggest open questions stemming from this chapter next. First, although the  $\varepsilon$ -window-median algorithm obtains the optimal regret bounds for the  $\varepsilon$ -ball loss, we have only been able to prove that (in the worst case) it runs in time exponential in the dimension  $d$ . We conjecture that this dependence is tight, but this is yet to be proven. More importantly, it is unclear if one can obtain a fully polynomial runtime algorithm for this problem while also attaining the optimal  $O(C_0 + d \log(1/\varepsilon))$ . Second, it is an intriguing open question to identify a corruption-robust algorithm for the absolute loss that in the absence of corruptions (i.e., when  $C_1 \rightarrow 0$ ) obtains the optimal dependence on the dimension  $d$  (i.e., whether the regret bound can become  $O(C_1 + d \log d)$ ). Finally, although in Chapter 7 we were able to present a unified framework that provided regret bounds for corruption-robust contextual pricing as well, this has not been possible in the present chapter. Due to its potential for applicability in real-life scenarios, obtaining tighter algorithms for the pricing loss is one of the most exciting open questions.

# **Part IV**

## **Fairness Considerations in Incentive-Aware ML**

# 9

## Information Discrepancy in Incentive-Aware Learning

### 9.1 CHAPTER OVERVIEW

In this chapter, we look at the problem of incentive-aware ML from the societal lens. Our goal is to study the societal impact that a benevolent decision-maker (henceforth referred to as *principal*) has on the welfare of the different subgroups given the information discrepancy that arises due to observation learning, which as we explained in Chapter 1 naturally arises as a result of the population being clustered and not having exact perfect information regarding the rules that the decision maker uses. We think of the welfare as a measure of the quality improvement that the agents ex-

perience (e.g., the amount by which their creditworthiness actually increases by participating in this game). The present chapter introduces the first framework to study information discrepancies in the context of incentive-aware learning, concurrently and independently with [Ghalme et al. \(2021\)](#). The considerations of the two works are completely orthogonal, please see the Related Work (Chapter 9.1.1) for more details. Our results can be summarized as follows.

**Equilibrium Model.** We propose a novel model for the interaction of the principal and the agents from different subgroups when there is information discrepancy on the principal’s scoring rule (Chapter 9.2). Based on our model, we show how the agents of different subgroups use observation learning to learn from their peer-networks and compute the closed-form solutions for them and the principal (Chapter 9.3).

**Cross-Subgroup Improvement.** Our study focuses on measuring the disparities in the true quality improvement across subgroups in the equilibrium where the principal optimizes the average welfare across subgroups. We begin by proving a surprising and worrisome negative externality; even when the principal optimizes for the sum of the subgroups’ average welfare, some of the subgroups may in fact suffer a *deterioration* in their true “quality” or label, even after best responding. Fortunately, we show that true improvement is guaranteed for all subgroups under *moderate* conditions, e.g. when they have similar—but *not necessarily identical*—costs for effort exertion or when the informational overlap between them is minimal (Chapter 9.4.1). Next, we show that abstracting away from cost discrepancies in altering their features, the cross-subgroup total improvement disparity is in the worst case upper bounded by the subgroups’ informational overlap on the principal’s decision rule and prove the conditions for this disparity to vanish (Chapter 9.4.2). To further address cases where this disparity does not vanish, we show that under moderate conditions we can guarantee that each subgroups reaches their *optimal* improvement (Chapter 9.4.3). We conclude our theoretical analysis by showing how similar conclusions can be drawn even if the principal is a learner who does not originally know the properties of the agents’ subgroups but has to *learn* them instead (Chapter E.1).

**Empirical Evaluation.** We empirically evaluate our findings on two real-world datasets that are commonly used in the literature; the TAIWAN-CREDIT and the ADULT dataset. Our experiments com-

plement and validate our theoretical results in settings not fully characterized by our theoretical conditions on subgroups' informational overlap (Chapter 9.5).

### 9.1.1 RELATED WORK

The results of this chapter are primarily related to three strands of literature. The first one advocates that changes in the original inputs of the agents are considered "gaming", hence the learner wants to construct algorithms that are robust to such behavior. This perspective has been extensively discussed in Part I and Chapter 5 for strategic classification specifically. In the present chapter, there is no "robustness" consideration. In a concurrent and independent work, [Ghalme et al. \(2021\)](#) also study non-transparency in strategic classification; they define and characterize the "price of opacity", and show conditions under which fully transparent classifiers are the recommended policy. Our results are orthogonal in that we study the *societal implications* of "opacity". When the learner optimizes for robustness to strategic behavior, the deployed algorithm has disparate impact on different subgroups of the population ([Hu et al., 2019](#), [Milli et al., 2019](#)). [Braverman and Garg \(2020\)](#) consider the societal impact of introducing randomized or noisy classifiers to mitigate inequalities, but only focus on a single-dimensional case. In this chapter, we also consider the disparate impact of the algorithm to the different subgroups, but the primary source of inequality is the information discrepancy on the principal's algorithm, which in turn depends on the agent's subgroup.

The second strand of literature advocates that machine learning algorithms should incentivize "good" strategic behavior (*aka improvements*) ([Kleinberg and Raghavan, 2020](#), [Ustun et al., 2019](#), [Khajehnejad et al., 2019](#), [Tsirtsis and Rodriguez, 2020](#), [Liu et al., 2020](#), [Haghtalab et al., 2020](#), [Alon et al., 2020](#), [Chen et al., 2020a](#), [Gupta et al., 2019b](#)). The results of this chapter are most closely related to ([Liu et al., 2020](#), [Haghtalab et al., 2020](#), [Gupta et al., 2019b](#)). [Liu et al. \(2020\)](#) study the long-term impact of incentive-aware learning to different subgroups, but they focus on decision rules that are fully known to the agents. [Haghtalab et al. \(2020\)](#) study social welfare maximization when the learner does not have full knowledge of the feature space of the agents, contrary to our model where the information discrepancies appear on the agents' side. [Gupta et al. \(2019b\)](#) minimize the difference in recourse across subgroups, while maintaining accuracy guarantees for the classifier, whereas we focus on a principal optimizing the social welfare.

The third strand concerns causality in incentive-aware learning ((Miller et al., 2020, Bechavod et al., 2021, Shavit et al., 2020) and more broadly (Perdomo et al., 2020)), where the learner wishes to learn the causal relationship between the agents’ features and their labels/scores by leveraging the agents’ strategic behavior. Importantly, in our setting, even if the principal knows the causal relationship perfectly, the disparate impact from the algorithm may still be unavoidable.

Finally, the present chapter has connections with works studying social welfare in the fairness literature (Heidari et al., 2018, 2019, Hu and Chen, 2020). Heidari et al. (2018) propose incorporating social welfare considerations to the standard loss minimization goal of machine learning. We instead give theoretical guarantees on the disparate impact that information discrepancy has, even if the principal is *optimizing* for the social welfare. Hu and Chen (2020) study the social welfare implications that result from a fair classification algorithm. Heidari et al. (2019) also study how agents in different subgroups invest their efforts through observation learning within their groups. In our case, the agents perform a different type of observation learning, as they try to infer the parameters in the deployed scoring rule. In comparison, there is a model behavior within each subgroup that the agents imitate. Additionally, our focus is on the improvement across different groups, whereas their focus is on how effort gets distributed unevenly across groups.

## 9.2 MODEL AND PRELIMINARIES

We study a Stackelberg game between a *principal* and a population of *agents* comprised of  $m$  subgroups with different distributions over the feature space  $\mathcal{X} \subseteq \mathbb{R}^d$ . We focus on the case  $m = 2$  for clarity, but our results extend to arbitrary  $m$ , as outlined in Appendix E.4. Let the subgroups be  $G_1$  and  $G_2$ , with associated distributions of feature vectors  $\mathcal{D}_1$  and  $\mathcal{D}_2$  respectively over  $\mathcal{X}$ . Let  $\mathcal{S}_1, \mathcal{S}_2$  be the subspaces defined by the supports of  $\mathcal{D}_1, \mathcal{D}_2$ . Let  $\Pi_1, \Pi_2 \in \mathbb{R}^{d \times d}$  be the orthogonal projection matrices onto subspaces  $\mathcal{S}_1, \mathcal{S}_2$  respectively. Let  $\mathbf{w}^* \in \mathbb{R}^d$  denote the *ground truth*<sup>\*</sup> linear assessment rule (which is *known* to the principal through past observations): i.e., for a feature vector  $\mathbf{x}$ , the corresponding agent’s expected *true* “quality” is given by  $\mathbb{E}[y | \mathbf{x}] = \langle \mathbf{w}^*, \mathbf{x} \rangle$ .

The principal deploys a linear scoring rule  $\mathbf{w} \in \mathbb{R}^d$ . Agent  $i$  from subgroup  $g$  draws private feature vector  $\mathbf{x}_{g,i} \sim \mathcal{D}_g$ . Initially, agents from both subgroups have no information regarding

---

<sup>\*</sup>While  $\mathbf{w}^*$  is optimal for prediction accuracy, it may not be the one maximizing the welfare across groups.

$\mathbf{w}$ , so they simply report  $x_{g,i}$  to the principal and receive scores  $\hat{y}_{g,i} = \langle \mathbf{w}, x_{g,i} \rangle$ . After enough agents from both subgroups have received scores for their reported features, the remaining agents use this past information (i.e., feature-predicted score tuples) to appropriately alter their feature vectors from  $\mathbf{x} \sim \mathcal{D}_g$  to  $\hat{\mathbf{x}}(\mathbf{x}; g)$ . Knowing that the ground truth assessment rule together with the scoring rule that the principal deploys are *linear*, and given the fact that they are risk-averse, agents perform empirical risk minimization (ERM) on the peer-dataset comprised of the first unmodified  $N_g \in \mathbb{R}_+$  samples  $S_g = \{(\mathbf{x}_{g,i}, \hat{y}_{g,i})\}_{i \in [N_g]}$  to compute an estimate  $\mathbf{w}_{\text{est}}(g)$  of the deployed scoring rule  $\mathbf{w}$ . Note that the use of ERM is a natural choice given that the agents are risk-averse, fully rational, and have no other information.

Given original features  $\mathbf{x}$  and estimation rule  $\mathbf{w}_{\text{est}}(g)$ , each (myopically rational) agent chooses  $\hat{\mathbf{x}}(\mathbf{x}; g)$  as the  $\mathbf{x}'$  that optimizes their underlying utility function, which similarly to standard models in strategic classification (Dong et al., 2018, Chen et al., 2020b) is defined as

$$u(\mathbf{x}, \mathbf{x}'; g) = \text{Score}(\mathbf{x}'; g) - \text{Cost}(\mathbf{x}, \mathbf{x}'; g) \quad (9.2.1)$$

where  $\text{Score}(\mathbf{x}'; g) = \langle \mathbf{w}_{\text{est}}(g), \mathbf{x}' \rangle$  is the *estimate* value the agent derives for reporting feature vector  $\mathbf{x}'$  and  $\text{Cost}(\mathbf{x}, \mathbf{x}'; g) = \frac{1}{2}(\mathbf{x}' - \mathbf{x})^\top A_g(\mathbf{x}' - \mathbf{x})$  is the agent's cost for modifying vector  $\mathbf{x}$  into  $\mathbf{x}'$ . Note that the actual value that the agent derives by reporting  $\mathbf{x}'$  is the outcome  $\langle \mathbf{w}^*, \mathbf{x}' \rangle$ . But  $\mathbf{w}^*$  is never revealed to the agent; the only information that she has is the estimate for the principal's  $\mathbf{w}$ . Regarding the cost function now, we call  $A_g \in \mathbb{R}^{d \times d}$  the *cost matrix* for group  $g$ , and assume it is *positive definite* (PD).<sup>†</sup> Due to not restricting  $A_g$  further, this cost function family is rather large and encapsulates some cost functions used in the literature on strategic classification (see e.g., (Dong et al., 2018, Ahmadi et al., 2021)). This functional form is one of the simplest ways to model important practical situations in which features can be modified in a correlated manner, and investing in one feature may lead to changes in other features.

At a high level, the utility in Eq. (9.2.1) captures the “net gains” that an agent obtains from spending effort to report  $\mathbf{x}'$ , rather than  $\mathbf{x}$ . Since  $\hat{\mathbf{x}}(\mathbf{x}; g)$  is the best response coming from Eq. (9.2.1), then  $\hat{\mathbf{x}}(\mathbf{x}; g) = \arg \max_{\mathbf{x}' \in \mathcal{X}} u(\mathbf{x}, \mathbf{x}'; g)$ . As we show in Chapter 9.3 the best response  $\hat{\mathbf{x}}(\mathbf{x}; g)$  takes

---

<sup>†</sup>In turn, one can write  $\text{Cost}(\mathbf{x}, \mathbf{x}'; g) = \frac{1}{2} \|\sqrt{A_g}(\mathbf{x}' - \mathbf{x})\|_2^2$  noting that  $\sqrt{A_g}$  is well-defined.

the form  $\mathbf{x} + \Delta_g(\mathbf{w})$  for a “movement” function<sup>†</sup>  $\Delta_g(\mathbf{w})$  to be specified shortly. Putting everything together, Protocol 9.1 summarizes the principal-agent interaction. When it comes to the principal’s

---

**Protocol 9.1:** Principal-Agent Interaction Protocol

---

- 1 Nature decides the ground truth assessment function  $\mathbf{w}^*$ .
  - 2 Learner deploys score rule  $\mathbf{w} \in \mathbb{R}^d$  (Eq. (9.2.2)), but does *not* reveal it to the agents.
  - 3 Agents from subgroup  $g \in \{1, 2\}$  draw their (private) feature vectors  $\mathbf{x} \sim \mathcal{D}_g$ .
  - 4 Given peer-dataset  $S_g$ , (private) feature vector  $\mathbf{x}$ , utility function  $u(\mathbf{x}, \mathbf{x}'; g)$ , the agents *best-respond* with feature vector  $\hat{\mathbf{x}}(\mathbf{x}; g) = \arg \max_{\mathbf{x}' \in \mathcal{X}} u(\mathbf{x}, \mathbf{x}'; g)$ .
- 

behavior, we posit that the principal’s objective is to maximize the agents’ average social welfare across groups (“social welfare” for short), defined as the sum over groups of the average (over agents) and expected (over the randomness of the labels) improvement of their true (as measured by  $\mathbf{w}^*$ ) labels, after they best-respond. Formally, the principal deploys the equilibrium rule  $\mathbf{w}_{SW}$ :

$$\mathbf{w}_{SW} = \arg \max_{\mathbf{w}' : \|\mathbf{w}'\|_2 \leq 1} SW(\mathbf{w}') = \arg \max_{\mathbf{w}' : \|\mathbf{w}'\|_2 \leq 1} \sum_{g \in \{1, 2\}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_g} [\langle \hat{\mathbf{x}}(\mathbf{x}; g), \mathbf{w}^* \rangle] \quad (9.2.2)$$

In section 9.4, we additionally consider a joint objective by the principal, who attempts to optimize for both social welfare and predictive accuracy.

Our goal in this chapter is to study the *true improvement* among subgroups when there is information discrepancy, *at a Stackelberg equilibrium* of our game. In other words, the principal and the agents best respond to each other, with the principal acting first and committing to a rule in anticipation of the strategic best responses of the agents. We quantify our study of true improvement based on two notions: *total* true improvement and *per-unit* true improvement.

**Definition 9.1.** For rule  $\tilde{\mathbf{w}} \in \mathbb{R}^d$ , we define the total true improvement for subgroup  $g$  as:

$$\mathcal{I}_g(\tilde{\mathbf{w}}) = \langle \hat{\mathbf{x}}(\mathbf{x}; g), \mathbf{w}^* \rangle - \langle \mathbf{x}, \mathbf{w}^* \rangle = \langle \mathbf{x} + \Delta_g(\tilde{\mathbf{w}}), \mathbf{w}^* \rangle - \langle \mathbf{x}, \mathbf{w}^* \rangle = \langle \Delta_g(\tilde{\mathbf{w}}), \mathbf{w}^* \rangle.$$

For the same rule, we define the per-unit true improvement for subgroup  $g$  as:

$$u\mathcal{I}_g(\tilde{\mathbf{w}}) = \mathcal{I}_g\left(\frac{\Pi_g \tilde{\mathbf{w}}}{\|\Pi_g \tilde{\mathbf{w}}\|_2}\right) = \left\langle \Delta_g\left(\frac{\Pi_g \tilde{\mathbf{w}}}{\|\Pi_g \tilde{\mathbf{w}}\|_2}\right), \mathbf{w}^* \right\rangle.$$

---

<sup>†</sup>The use of  $\mathbf{w}$  as an argument of  $\Delta_g(\cdot)$  is a slight abuse of notation, which is only used in our analysis. The agents do not have direct access to  $\mathbf{w}$ .

The usefulness of defining the total true improvement as one of our measures is clear. The per-unit true improvement only considers the part of the deployed scoring rule that belongs in the relevant subspace of each subgroup, and measures how efficient the direction of this rule projected onto the relevant subspace is at inducing improvement for the subgroup.

We focus on three objectives for the two subgroups: *do-no-harm*, *equality*, and *optimality*.

**Definition 9.2** (Do-No-Harm). *A rule  $\tilde{\mathbf{w}}$  causes no harm for subgroup  $g$  if  $\mathcal{I}_g(\tilde{\mathbf{w}}) \geq 0$ .*

**Definition 9.3** (Equality). *A rule  $\tilde{\mathbf{w}}$  enforces subgroup-equality if:  $\mathcal{I}_1(\tilde{\mathbf{w}}) = \mathcal{I}_2(\tilde{\mathbf{w}})$ .*

**Definition 9.4** (Optimality). *A rule  $\mathbf{w}'$  enforces  $g$ 's subgroup-optimality if:  $\mathbf{w}' = \arg \max_{\tilde{\mathbf{w}}} \mathbf{u} \mathcal{I}_g(\tilde{\mathbf{w}})$ .*

Optimality of the per-unit true improvement is equivalent to guaranteeing for a rule  $\mathbf{w}$  that no other  $\mathbf{w}'$  for which  $\|\Pi_g \mathbf{w}'\|_2 \leq \|\Pi_g \mathbf{w}\|_2$ , can induce greater improvement than  $\mathbf{w}$  does in group  $g$ .

Based on these objectives, we quantify how much the equilibrium play in this strategic interaction exacerbates inequalities across groups due to their information discrepancies, even in the best-case scenario, where the principal is optimizing the population's average welfare across groups.

### 9.3 EQUILIBRIUM COMPUTATION

In this section, we compute the equilibrium strategies. We first compute the agents' estimate rules ( $\mathbf{w}_{\text{est}}(g)$ ), given limited amount of information from their own subgroup. Then, we derive the closed form of the agents' best-response  $\hat{\mathbf{x}}(\mathbf{x}; g)$ . Finally, we explain how  $\mathbf{w}_{\text{est}}(g)$  and  $\hat{\mathbf{x}}(\mathbf{x}; g)$  affect the principal's optimization problem of Eq. (9.2.2).

#### 9.3.1 COMPUTING AN ESTIMATE FOR THE PRINCIPAL'S SCORE RULE

Recall that the agents run ERM on the peer dataset  $S_g$  to derive proxy  $\mathbf{w}_{\text{est}}$ . We posit that the agents are *risk averse*; i.e., they prefer "certain" outcomes, rather than betting on uncertain ones. In our setting, this means that the agents take the *minimum norm* ERM to break ties, since the agents only want to move in the direction that can surely improve their outcome. In contrast, if the agents invest their effort according to an estimate that is not the minimum norm ERM, then they may not

improve their outcome further but still incur a cost. Formally, the agents compute  $\mathbf{w}_{\text{est}}(g)$  as:

$$\mathbf{w}_{\text{est}}(g) = \arg \min_{\mathbf{w} \in W} \|\tilde{\mathbf{w}}\|_2^2, \text{ such that } W = \left\{ w : w = \arg \min_{\mathbf{w}'} \sum_{i \in [N_g]} (\mathbf{x}_{g,i}^\top \mathbf{w}' - \hat{y}_{g,i})^2 \right\} \quad (9.3.1)$$

When agents use ERM, we can state in closed form their estimate rule.

**Lemma 9.1.** *Agents from subgroup  $g$  using ERM compute the estimate rule  $\mathbf{w}_{\text{est}}(g) = \Pi_g \mathbf{w}$ .*

*Proof.* We first identify the rules  $\tilde{\mathbf{w}}$  that are the solutions of the error minimization part of Equation (9.3.1):

$$\tilde{\mathbf{w}} = \arg \min_{\mathbf{w}'} \sum_{i \in [N_g]} (\mathbf{x}_{g,i}^\top \mathbf{w}' - \hat{y}_{g,i})^2 = \arg \min_{\mathbf{w}'} \underbrace{\sum_{i \in [N_g]} ((\Pi_g \mathbf{x}_{g,i})^\top \mathbf{w}' - \hat{y}_{g,i})^2}_{f(\mathbf{w}')}, \quad (9.3.2)$$

where the second equation is due to the fact that since  $\forall \mathbf{x}_{g,i} \sim \mathcal{D}_g$ , then  $\Pi_g \mathbf{x}_{g,i} = \mathbf{x}_{g,i}$ . To solve the minimization problem of Eq. (9.3.2), we take the first order conditions, so at the optimal  $\tilde{\mathbf{w}}$ :

$$\nabla f(\tilde{\mathbf{w}}) = 0 \Leftrightarrow 2 \sum_{i \in [N_g]} (\Pi_g \mathbf{x}_{g,i}) \left( (\Pi_g \mathbf{x}_{g,i})^\top \tilde{\mathbf{w}} - \hat{y}_{g,i} \right) = 0 \quad (9.3.3)$$

Now,  $\tilde{\mathbf{w}} = \Pi_g \mathbf{w}$  is one of the solutions of Eq. (9.3.3), since  $\hat{y}_{g,i} = (\Pi_g \mathbf{x}_{g,i})^\top \mathbf{w} = (\Pi_g \mathbf{x}_{g,i})^\top \Pi_g \mathbf{w}$ . Next, we argue that due to the norm-minimization rule we use for tie-breaking, it is also the *unique* solution. To do so, let  $\tilde{\mathbf{w}}$  be a norm-minimizing solution of Eq. (9.3.3), and write  $\tilde{\mathbf{w}} = \Pi_g \mathbf{w} + \mathbf{x}'$ , where  $\mathbf{x}'$  is an arbitrary vector; note that this is without loss of generality. We can write  $\tilde{\mathbf{w}} = \Pi_g \mathbf{w} + \Pi_g \mathbf{x}' + \Pi_g^\perp \mathbf{x}'$  (where  $\Pi_g^\perp \mathbf{x}'$  is the projection of  $\mathbf{x}'$  in the orthogonal subspace of  $\mathcal{S}_g$ ). Now, note that  $\Pi_g \tilde{\mathbf{w}} = \Pi_g \mathbf{w} + \Pi_g \mathbf{x}' + \Pi_g \Pi_g^\perp \mathbf{x}' = \Pi_g \mathbf{w} + \Pi_g \mathbf{x}'$  is also a solution to Eq. (9.3.3), as

$$\begin{aligned} \sum_{i \in [N_g]} (\Pi_g \mathbf{x}_{g,i}) \left( (\Pi_g \mathbf{x}_{g,i})^\top \Pi_g \tilde{\mathbf{w}} - \hat{y}_{g,i} \right) &= \sum_{i \in [N_g]} (\Pi_g \mathbf{x}_{g,i}) \left( \mathbf{x}_{g,i}^\top \Pi_g^\top \Pi_g \tilde{\mathbf{w}} - \hat{y}_{g,i} \right) \\ &= \sum_{i \in [N_g]} (\Pi_g \mathbf{x}_{g,i}) \left( \mathbf{x}_{g,i}^\top \Pi_g \tilde{\mathbf{w}} - \hat{y}_{g,i} \right), \end{aligned}$$

where the last step is due to the fact that  $\Pi_g$  represents an orthogonal projection, hence  $\Pi_g^\top = \Pi_g$

and  $\Pi_g \Pi_g = \Pi_g$ . Further, if  $\Pi_g^\perp \mathbf{x}' \neq 0$ , we have that

$$\|\tilde{\mathbf{w}}\|_2 = \|\Pi_g \mathbf{w} + \Pi_g \mathbf{x}'\|_2 + \|\Pi_g^\perp \mathbf{x}'\|_2 > \|\Pi_g \mathbf{w} + \Pi_g \mathbf{x}'\|_2 = \|\Pi_g \tilde{\mathbf{w}}\|_2$$

by orthogonality of  $\Pi_g(\mathbf{w} + \mathbf{x}')$  and  $\Pi_g^\perp \mathbf{x}'$ . This contradicts the fact that  $\tilde{\mathbf{w}}$  is a norm-minimizing solution of Eq. (9.3.3). Therefore, we have  $\tilde{\mathbf{w}} = \Pi_g \mathbf{w} + \Pi_g \mathbf{x}'$ .

Using this together with  $\hat{y}_{g,i} = (\Pi_g \mathbf{x}_{g,i})^\top \Pi_g \mathbf{w}$ , the left-hand side of Eq. (9.3.3) becomes:

$$\begin{aligned} & \sum_{i \in [N_g]} (\Pi_g \mathbf{x}_{g,i}) \left( (\Pi_g \mathbf{x}_{g,i})^\top \Pi_g \mathbf{w} + (\Pi_g \mathbf{x}_{g,i})^\top \Pi_g \mathbf{x}' - (\Pi_g \mathbf{x}_{g,i})^\top \Pi_g \mathbf{w} \right) \\ &= \sum_{i \in [N_g]} (\Pi_g \mathbf{x}_{g,i}) (\Pi_g \mathbf{x}_{g,i})^\top \mathbf{x}' \end{aligned} \quad (9.3.4)$$

where the last equality comes from the fact that  $\Pi_g \Pi_g^\perp = 0_{d \times d}$ . We next prove a technical lemma.

**Lemma 9.2.** Let  $Z \triangleq \sum_{i \in [N_g]} (\Pi_g \mathbf{x}_{g,i}) (\Pi_g \mathbf{x}_{g,i})^\top$  be full-rank in subspace  $\mathcal{S}_g$ . Then, for any vector  $\mathbf{x}' \in \mathbb{R}^d$  it holds that  $Z(\Pi_g \mathbf{x}') = 0$  if and only if  $\Pi_g \mathbf{x}' = 0$ .

*Proof.* If  $\Pi_g \mathbf{x}' = 0$  then, it holds that  $Z(\Pi_g \mathbf{x}') = 0$ . So, assume that  $Z(\Pi_g \mathbf{x}') = 0$ .

Let  $r = \text{rank}(\mathcal{S}_g)$  (hence,  $r = \text{rank}(Z)$ ). Let us denote by  $v_1, \dots, v_r$  the eigenvectors of  $Z$  corresponding to eigenvalues  $\lambda_1, \dots, \lambda_r$  for which  $\lambda_i > 0, \forall i \in [r]$ ; note that  $(v_1, \dots, v_r)$  span  $\mathcal{S}_g$ . For the rest of the eigenvalues (i.e.,  $i \in [r+1, d], \lambda_i = 0$ ) the remaining eigenvectors are denoted as  $v_{r+1}, \dots, v_d$ . Without loss of generality, we take  $v_i$  to have norm 1 for all  $i$ ; since  $Z$  is a symmetric matrix,  $(v_1, \dots, v_r)$  is an orthonormal basis for  $\mathcal{S}_G$  and  $(v_1, \dots, v_d)$  is an orthonormal basis for  $\mathbb{R}^d$ .

Let  $V$  denote the  $d \times d$  matrix  $[v_1^\top \ v_2^\top \ \cdots \ v_d^\top]$  that is the change of basis that transforms the standard basis into  $(v_1, \dots, v_d)$ . By orthonormality of  $(v_1, \dots, v_d)$ :  $V$  is such that  $V^\top V = \mathbb{I}$ . In turn,

$$Z = (\Pi_g \mathbf{x}') = V^\top V Z V^\top V \Pi_g \mathbf{x}' \quad (9.3.5)$$

Let us define matrices  $P_1 = V Z V^\top$  and  $P_2 = V \Pi_g \mathbf{x}'$ .  $P_1$  is a diagonal matrix having  $\lambda_1, \dots, \lambda_d$  on the diagonal (and hence, it only has positive values until row  $r$  and 0's for rows in  $\{r+1, d\}$ ). Also,

$$P_2 = V \Pi_g \mathbf{x}' = [a_1 \ \cdots \ a_r \ 0 \ \cdots \ 0]^\top, \text{ where } a_i = v_i^\top (\Pi_g \mathbf{x}').$$

Substituting the values of  $P_1, P_2$  in Eq. (9.3.5) we have that:

$$Z(\Pi_g \mathbf{x}') = V^\top [\lambda_1 a_1 \ \dots \ \lambda_r a_r \ 0 \ \dots \ 0]^\top$$

But  $Z(\Pi_g \mathbf{x}') = 0$  if and only if  $\lambda_i a_i = 0, \forall i \in [r]$ , because  $V$  is invertible. Since  $\lambda_i > 0$  for  $i \in [r]$ , it must be that  $a_i = 0$ . Since then,  $V\Pi_g \mathbf{x}' = 0$  and  $V$  is invertible, this implies that  $\Pi_g \mathbf{x}' = 0$ .  $\blacksquare$

Defining  $Z$  as  $Z = \sum_{i \in [N_g]} (\Pi_g \mathbf{x}_{g,i}) (\Pi_g \mathbf{x}_{g,i})^\top$ ,  <sup>$\S$</sup>  then from Lemma 9.2, Eq. (9.3.4) is equal to 0 if and only if  $\Pi_g \mathbf{x}' = 0$ . This directly yields  $\tilde{\mathbf{w}} = \Pi_g \mathbf{w} + \Pi_g \mathbf{x}' = \Pi_g \mathbf{w}$ .  $\blacksquare$

### 9.3.2 CLOSED FORM OF THE AGENT'S BEST-RESPONSE

Slightly abusing notation, the agents' value function becomes:  $\text{Score}(\mathbf{x}, \mathbf{x}'; g) = \langle \mathbf{w}_{\text{est}}(g), \mathbf{x}' \rangle$ , which is equal to  $\langle \Pi_g \mathbf{w}, \mathbf{x}' \rangle$  from Lemma 9.1. So their utility function (Eq. (9.2.1)) takes the form:

$$u(\mathbf{x}, \mathbf{x}'; g) = \langle \Pi_g \mathbf{w}, \mathbf{x}' \rangle - \frac{1}{2} \left\| \sqrt{A_g} (\mathbf{x}' - \mathbf{x}) \right\|^2 \quad (9.3.6)$$

**Lemma 9.3.** *The best-response of an agent from subgroup  $g$  with feature vector  $\mathbf{x}$  is:  $\hat{\mathbf{x}}(\mathbf{x}; g) = \mathbf{x} + A_g^{-1} \Pi_g \mathbf{w} \triangleq \mathbf{x} + \Delta_g(\mathbf{w})$ .*

*Proof.* The function in Eq. (9.3.6) is concave. At the optimum  $\mathbf{x}'$  from the first order conditions we have that  $\nabla u(\mathbf{x}, \mathbf{x}'; g) = \Pi_g \mathbf{w} - A_g(\mathbf{x}' - \mathbf{x}) = 0$ . Solving the latter in terms of  $\mathbf{x}'$  and using the fact that matrix  $A_g$  is positive definite (hence also *invertible*) gives us the result.  $\blacksquare$

In equilibrium, the principal *knows* that the agent's best-response as a function of their private  $\mathbf{x}$  is given by Lemma 9.3. We use this when solving the principal's optimization problem.

### 9.3.3 THE PRINCIPAL'S CHOSEN SCORING RULE IN EQUILIBRIUM

Using the fact that the principal can compute  $\Delta_g(\mathbf{w})$  for any subgroup  $g$ , we can obtain a closed form solution for the principal's chosen rule  $\mathbf{w}$  (i.e., the solution to Eq. (9.2.2)).

---

<sup>$\S$</sup> Given enough samples from the peer dataset (i.e., a large enough  $N_g$ ), one guarantees that  $Z$  is full rank.

**Lemma 9.4.** *The principal's score rule that maximizes the social welfare in equilibrium is:*

$$\mathbf{w}_{\text{SW}} = \frac{(\Pi_1 A_1^{-1} + \Pi_2 A_2^{-1}) \mathbf{w}^*}{\|(\Pi_1 A_1^{-1} + \Pi_2 A_2^{-1}) \mathbf{w}^*\|} \quad (9.3.7)$$

*Proof.* We first note a useful lemma (which we formally state and prove in Lemma E.1), namely that if  $Q \in \mathbb{R}^{d \times d}$  is a symmetric PD matrix and  $c$  a vector in  $\mathbb{R}^d$  then the solution of the optimization problem  $\max_x c^\top x$  such that  $x^\top Qx \leq b$  has unique solution  $x = \frac{b Q^{-1} c}{\sqrt{c^\top Q^{-1} c}}$ .

Using the closed-form of the agents' best-response from Lemma 9.3 in Eq. (9.2.2) we get that:

$$\begin{aligned} \mathbf{w}_{\text{SW}} &= \arg \max_{\mathbf{w}' : \|\mathbf{w}'\|_2 \leq 1} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_1} [\langle \mathbf{w}^*, \hat{\mathbf{x}}(\mathbf{x}; 1) \rangle] + \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_2} [\langle \mathbf{w}^*, \hat{\mathbf{x}}(\mathbf{x}; 2) \rangle] \\ &= \arg \max_{\mathbf{w}' : \|\mathbf{w}'\|_2 \leq 1} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_1} [\langle \mathbf{w}^*, \mathbf{x} + \Delta_1(\mathbf{w}) \rangle] + \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_2} [\langle \mathbf{w}^*, \mathbf{x} + \Delta_2(\mathbf{w}) \rangle] \\ &= \arg \max_{\mathbf{w}' : \|\mathbf{w}'\|_2 \leq 1} \langle (A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2) \mathbf{w}', \mathbf{w}^* \rangle \end{aligned} \quad (9.3.8)$$

We rewrite the objective function to be optimized above as:  $[\mathbf{w}^* \top (A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)] \mathbf{w} = [(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*]^\top \mathbf{w}$  and the constraint for  $\mathbf{w}$  remains:  $\mathbf{w}^\top \mathbf{w} \leq 1$ . This problem is an instance of the problem solved in Lemma E.1 for  $c = (A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*$ ,  $b = 1$  and  $Q$  the identity matrix. Substituting  $c, Q, b$  in the solution of Lemma E.1 gives the result. ■

Using the same techniques, we can also make the following claim for the principal, which is useful in connecting the true improvement that a subgroup experiences when the principal chooses  $\mathbf{w}$  to maximize the social welfare of a particular subgroup  $g$ .

**Lemma 9.5.** *The score rule maximizing the social welfare of subgroup  $g$  is:  $\mathbf{w}_g = \frac{(A_g^{-1} \Pi_g)^\top \mathbf{w}^*}{\|(A_g^{-1} \Pi_g)^\top \mathbf{w}^*\|}$ .*

## 9.4 EQUILIBRIUM ANALYSIS

In this section, we study the societal impact of the equilibrium strategies of the principal and the agents computed in Section 9.3, in terms of the disparities in the resulting improvement across the different subgroups. We do so by examining the feasibility of each of the objectives of cross-subgroup improvement previously introduced in Chapter 9.2. Throughout, we make the technical

assumption that  $(A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)^\top \mathbf{w}^* \neq 0$ , since if the latter is an equality, then the objective of Eq. (9.3.8) is always 0.

#### 9.4.1 Do-No-HARM

When the benevolent principal deploys an equilibrium rule maximizing the social welfare of the population, one could expect that this rule does not cause any negative externality (i.e., outcome deterioration). Surprisingly, this is not the case in general, as we present in the following example.<sup>\ddagger</sup>

**Example 9.1.** Assume that the cost and the projection matrices for the two subgroups are:

$$A_1 = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}, \quad A_2 = 4 \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}, \quad \Pi_1 = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{and} \quad \Pi_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

Note that  $A_1, A_2$  are symmetric and PD, as their eigenvalues are  $\lambda^1 = 3 \pm 2\sqrt{2}$  and  $\lambda^2 = 2(3 \pm \sqrt{5})$  respectively. Further,  $\Pi_1, \Pi_2$  are orthogonal projections, as  $\Pi_g^2 = \Pi_g = \Pi_g^\top$ . Finally, assume that  $\mathbf{w}^{*\top} = [0 \ \sqrt{a}]$ , for scalar  $a > 0$ . Then, for the numerator of  $\mathcal{I}_2(\mathbf{w}^*)$  we have that:

$$\mathbf{w}^{*\top} \left( A_1^{-1}\Pi_1\Pi_2A_2^{-1\top} + A_2^{-1}\Pi_2A_2^{-1\top} \right) \mathbf{w}^* = \mathbf{w}^{*\top} \begin{bmatrix} 2 & 1 \\ -5/8 & -5/16 \end{bmatrix} \mathbf{w}^* = -\frac{5}{16}a < 0 \quad (a > 0).$$

This example conveys the tension that arises as a result of the different costs that the subgroups incur for altering their feature vectors, which in the worst case can result in a negative externality for one of the two subgroups. Luckily, this is not always true; if we abstract away from factors that affect the cost functions (e.g., consider cost functions that differ among subgroups by only a constant<sup>\|\</sup>) then the only discrepancy between the two groups is due to the different, partial information that they get from the principal's assessment rule. This is enough to guarantee that for *any* such  $\Pi_1, \Pi_2$  the principal is guaranteed to cause no negative externality to any of the subgroups in equilibrium.

---

<sup>\ddagger</sup>A similar example can be proved for the case that the product matrix  $A_1A_2$  is not PSD.

<sup>\|\</sup>This covers most of the cost functions considered in prior work (Hardt et al., 2016, Dong et al., 2018, Chen et al., 2020b, Ahmadi et al., 2021), where the cost matrices are diagonal with identical coefficients for all agents.

**Proposition 9.1.** *There is no negative externality for any of the subgroups in equilibrium, when the cost matrices are proportional to each other, i.e.,  $A_1 = c \cdot A_2$  for a scalar  $c > 0$ .*

*Proof.* Fix a group  $g \in \{1, 2\}$  (wlog, let  $g = 1$ ), and let  $\bar{A} = A_1^{-1}$ . Then, from Theorem 9.1 no negative externality for subgroup  $g$  is guaranteed if and only if:

$$\begin{aligned} \left\langle \left( \bar{A}\Pi_1\bar{A} + \frac{1}{c}\bar{A}\Pi_2\Pi_1\bar{A} \right), \mathbf{w}^* \right\rangle \geq 0 &\Leftrightarrow \left\langle \left( \bar{A}\Pi_1\bar{A} + \frac{1}{c}\bar{A}\Pi_2\Pi_1\bar{A} \right)^\top \mathbf{w}^* \right\rangle^\top \mathbf{w}^* \geq 0 \\ &\Leftrightarrow \mathbf{w}^{*\top} \left( \bar{A}\Pi_1\bar{A} + \frac{1}{c^2}\bar{A}\Pi_2\Pi_1\bar{A} \right) \mathbf{w}^* \geq 0 \end{aligned} \quad (9.4.1)$$

Eq. (9.4.1) is true if and only if matrix  $\bar{A}\Pi_1\bar{A} + \bar{A}\Pi_2\Pi_1\bar{A}$  is PSD. Matrix  $\Pi_1$  is by definition PSD. Matrix  $A$  is PD, hence its inverse,  $\bar{A}$ , is also PD. As a result, matrix  $\bar{A}\Pi_1\bar{A}$  is PSD. We shift our attention to matrix  $\bar{A}\Pi_2\Pi_1\bar{A}$  now. Since  $\Pi_1, \Pi_2$  are projection matrices, then the eigenvalues of their product  $\Pi_2\Pi_1$  are non-negative (Anderson Jr et al., 1985). Recall that a matrix is PSD if and only if its eigenvalues are non-negative. As a result, matrix  $\Pi_2\Pi_1$  is PSD. Using the same property as above (i.e., that if matrices  $A, B$  are PSD, then so is matrix  $ABA$ ) we can conclude that  $\bar{A}\Pi_2\Pi_1\bar{A}$  is PSD. If matrices  $A, B$  are PSD, then so is matrix  $A + B$ . Hence, matrix  $\bar{A}\Pi_2\bar{A} + \bar{A}\Pi_2\Pi_1\bar{A}$  is PSD, i.e., by definition that for any vector  $z$  we have that:  $z^\top (\bar{A}\Pi_1\bar{A} + \bar{A}\Pi_2\Pi_1\bar{A}/c)z \geq 0$ . ■

No negative externality is also experienced in equilibrium (even with arbitrarily different  $A_g$ 's) when the subspaces  $\mathcal{S}_1, \mathcal{S}_2$  are orthogonal. Intuitively, this happens when the two subgroups have no information overlap, as this implies that there is no trade-off for the improvements across them.

**Proposition 9.2.** *There is no negative externality in equilibrium, if subspaces  $\mathcal{S}_1, \mathcal{S}_2$  are orthogonal.*

*Proof.* Fix a group  $g \in \{1, 2\}$  (wlog let  $g = 1$ ). From Theorem 9.1 we need:

$$\left\langle (A_1^{-1}\Pi_1^2A_1^{-1} + A_2^{-1}\Pi_2\Pi_1A_1^{-1})^\top \mathbf{w}^*, \mathbf{w}^* \right\rangle \geq 0 \Leftrightarrow \left\langle (A_1^{-1}\Pi_1A_1^{-1})^\top \mathbf{w}^*, \mathbf{w}^* \right\rangle \geq 0 \quad (9.4.2)$$

where for the last inequality we used  $\Pi_2\Pi_1 = 0$  (as subspaces  $\mathcal{S}_1, \mathcal{S}_2$  are orthogonal) and  $\Pi_g^2 = \Pi_g$  (as orthogonal projection matrices). Eq. (9.4.2) holds since matrices  $\Pi_1$  and  $A_1$  are PSD. ■

More generally, the following is necessary and sufficient for causing no negative externality:

### Theorem 9.1

In equilibrium, there is no negative externality for subgroup  $g$  and any  $\mathbf{w}^*$  if and only if for both  $g \in \{1, 2\}$ , the matrix  $(A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)\Pi_g A_g^{-1} + A_g^{-1}\Pi_g(\Pi_1 A_1^{-1} + \Pi_2 A_2^{-1})$  is PSD.

*Proof.* By Definition 9.1, having no negative externality in equilibrium translates to:

$$\begin{aligned}
 \forall g : \mathcal{I}_g(\mathbf{w}) \geq 0 &\Leftrightarrow \left\langle A_g^{-1}\Pi_g \frac{(A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)^\top \mathbf{w}^*}{\|(A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)^\top \mathbf{w}^*\|_2}, \mathbf{w}^* \right\rangle \geq 0 && \text{(Lemma 9.4)} \\
 &\Leftrightarrow \left\langle A_g^{-1}\Pi_g (A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)^\top \mathbf{w}^*, \mathbf{w}^* \right\rangle \geq 0 \\
 &\Leftrightarrow \left\langle (\Pi_g^\top A_g^{-1})^\top (A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)^\top \mathbf{w}^*, \mathbf{w}^* \right\rangle \geq 0 && ((AB)^\top = B^\top A^\top \text{ and } A^{\top\top} = A) \\
 &\Leftrightarrow \left\langle ((A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2) (\Pi_g^\top A_g^{-1}))^\top \mathbf{w}^*, \mathbf{w}^* \right\rangle \geq 0 && ((AB)^\top = B^\top A^\top) \\
 &\Leftrightarrow \left\langle (A_1^{-1}\Pi_1 \Pi_g^\top A_g^{-1\top} + A_2^{-1}\Pi_2 \Pi_g^\top A_g^{-1\top})^\top \mathbf{w}^*, \mathbf{w}^* \right\rangle \geq 0 && ((A+B)C = AC + BC)
 \end{aligned}$$

Using the fact that  $\Pi_g^\top = \Pi_g$  (as orthogonal projection matrices) and that  $A_g^{-1\top} = A_g^{-1}$  (as  $A_g$  is a symmetric matrix), we obtain the condition  $q(\mathbf{w}^*) \geq 0$  where  $q(\mathbf{w}^*) = (\mathbf{w}^*)^\top M \mathbf{w}^*$  is a quadratic form with  $M = A_1^{-1}\Pi_1 \Pi_g A_g^{-1} + A_2^{-1}\Pi_2 \Pi_g A_g^{-1}$ . By standard linear algebra arguments, noting that  $(\mathbf{w}^*)^\top M \mathbf{w}^* = ((\mathbf{w}^*)^\top M \mathbf{w}^*)^\top = (\mathbf{w}^*)^\top M^\top \mathbf{w}^*$ , we can rewrite  $q(\mathbf{w}^*) = \frac{1}{2}(\mathbf{w}^*)^\top (M + M^\top) \mathbf{w}^*$ . The condition then holds for all  $\mathbf{w}^*$  if and only if

$$M + M^\top = A_1^{-1}\Pi_1 \Pi_g A_g^{-1} + A_2^{-1}\Pi_2 \Pi_g A_g^{-1} + A_g^{-1}\Pi_g \Pi_1 A_1^{-1} + A_g^{-1}\Pi_g \Pi_2 A_2^{-1}$$

is PSD. This concludes the proof. ■

### JOINT ACCURACY–SOCIAL WELFARE OBJECTIVE

As in many practical settings, the principal is expected to take into account a joint objective of predictive accuracy and social welfare. We show next that under mild conditions, deploying any combination of the social-welfare maximizing solution and the true underlying predictor results in inheriting the Do-No-Harm guarantee of the social welfare maximizer.

### Theorem 9.2

Assume that for all  $g \in G$ ,  $A_g^{-1}$  and  $\Pi_g$  commute. Then, if Do-No-Harm is guaranteed when the principal deploys the social-welfare maximizing solution, it is also guaranteed for any positive linear combination of  $\mathbf{w}_{\text{SW}}$  and  $\mathbf{w}^*$ .

*Proof.* Let  $\mathbf{w}^{\text{Joint}} := \alpha\mathbf{w}^* + \beta\mathbf{w}_{\text{SW}}$  for  $\alpha, \beta \geq 0$ . We have, for all  $g$ ,

$$\begin{aligned}
\mathcal{I}_g(\mathbf{w}^{\text{Joint}}) &:= \mathcal{I}_g(\alpha\mathbf{w}^* + \beta\mathbf{w}_{\text{SW}}) \\
&= \langle A_g^{-1}\Pi_g(\alpha\mathbf{w}^* + \beta\mathbf{w}_{\text{SW}}), \mathbf{w}^* \rangle \quad (\text{Definition 9.1}) \\
&= \alpha \langle A_g^{-1}\Pi_g\mathbf{w}^*, \mathbf{w}^* \rangle + \beta \langle A_g^{-1}\Pi_g\mathbf{w}_{\text{SW}}, \mathbf{w}^* \rangle \quad (\text{Linearity}) \\
&= \alpha\mathbf{w}^{*\top} \Pi_g A_g^{-1} \mathbf{w}^* + \beta\mathbf{w}_{\text{SW}}^{*\top} \Pi_g A_g^{-1} \mathbf{w}^* \\
&\geq 0 \quad (\Pi_g A_g^{-1} \succcurlyeq 0, \text{Do-No-Harm for } \mathbf{w}_{\text{SW}})
\end{aligned}$$

■

#### 9.4.2 TOTAL IMPROVEMENT

We shift now our attention to studying a stronger objective: equal total improvement across subgroups. Equality of total improvement is a strictly stronger objective than do-no-harm. Indeed, achieving equal total improvement guarantees that there exists no negative externality, since the social welfare is always non-negative. In order to single out the effects of information discrepancy we assume throughout this subsection that  $A_1 = A_2 = \mathbb{I}_{d \times d}$ . We begin by introducing a measure to help us in quantifying the difference in total improvement between subgroups:

**Definition 9.5.** Given a scoring rule  $\mathbf{w} \in \mathbb{R}^d$  and projections  $\Pi_1, \Pi_2 \in \mathbb{R}^{d \times d}$ , we define the overlap proxy between subgroups  $G_1, G_2$  with respect to  $\mathbf{w}$  to be  $r_{1,2}(\mathbf{w}) := \|\Pi_1\mathbf{w} - \Pi_2\mathbf{w}\|_2$ .

We next show an interesting insight - that the overlap proxy with respect to the underlying scoring rule  $\mathbf{w}^*$  in fact characterizes the difference of improvement in the worst case.

**Lemma 9.6.** Let  $\text{diff}_{1,2}(\mathbf{w}) \triangleq |\mathcal{I}_1(\mathbf{w}) - \mathcal{I}_2(\mathbf{w})|$  be the disparity in total improvement across subgroups when the principal's rule is  $\mathbf{w}$ . In equilibrium, if  $A_1 = A_2 = \mathbb{I}_{d \times d}$ , then:  $\text{diff}_{1,2}(\mathbf{w}_{\text{SW}}) \leq r_{1,2}(\mathbf{w}^*)$ . Further,

equality holds if and only if  $\Pi_1 \mathbf{w}^*$  and  $\Pi_2 \mathbf{w}^*$  are co-linear.

*Proof.* Note that

$$\begin{aligned}\mathcal{I}_1\left(\frac{(\Pi_1 + \Pi_2) \mathbf{w}^*}{\|(\Pi_1 + \Pi_2) \mathbf{w}^*\|}\right) - \mathcal{I}_2\left(\frac{(\Pi_1 + \Pi_2) \mathbf{w}^*}{\|(\Pi_1 + \Pi_2) \mathbf{w}^*\|}\right) &= \left\langle (\Pi_1 - \Pi_2) \frac{(\Pi_1 + \Pi_2) \mathbf{w}^*}{\|(\Pi_1 + \Pi_2) \mathbf{w}^*\|}, \mathbf{w}^* \right\rangle \\ &= \frac{1}{\|(\Pi_1 + \Pi_2) \mathbf{w}^*\|} \cdot \langle (\Pi_1 + \Pi_2) \mathbf{w}^*, (\Pi_1 - \Pi_2) \mathbf{w}^* \rangle.\end{aligned}$$

By Cauchy-Schwarz, we have that

$$\begin{aligned}&\left| \frac{1}{\|(\Pi_1 + \Pi_2) \mathbf{w}^*\|} \cdot \langle (\Pi_1 + \Pi_2) \mathbf{w}^*, (\Pi_1 - \Pi_2) \mathbf{w}^* \rangle \right| \\ &\leq \frac{1}{\|(\Pi_1 + \Pi_2) \mathbf{w}^*\|} \cdot \|(\Pi_1 + \Pi_2) \mathbf{w}^*\| \cdot \|(\Pi_1 - \Pi_2) \mathbf{w}^*\| \\ &= \|(\Pi_1 - \Pi_2) \mathbf{w}^*\| \\ &= r_{1,2}(\mathbf{w}^*),\end{aligned}$$

with equality if and only if  $(\Pi_1 + \Pi_2) \mathbf{w}^*$  and  $(\Pi_1 - \Pi_2) \mathbf{w}^*$  are colinear, i.e., there exists  $\alpha \in \mathbb{R}$  such that  $\alpha (\Pi_1 + \Pi_2) \mathbf{w}^* = (\Pi_1 - \Pi_2) \mathbf{w}^*$ , which can be equivalently written as  $(1 - \alpha)\Pi_1 \mathbf{w}^* = (1 + \alpha)\Pi_2 \mathbf{w}^*$ , i.e.,  $\Pi_1 \mathbf{w}^*$  and  $\Pi_2 \mathbf{w}^*$  are colinear. ■

**Remark 9.1.** In particular, note that the bound is tight when  $\Pi_1 = \Pi_2$  (perfect overlap) and  $\Pi_1 = \mathbb{I}_{d \times d}$ ,  $\Pi_2 = 0$  (maximum informational disparities across groups).

We state next the necessary and sufficient conditions for equal total improvement across groups.

### Theorem 9.3

In equilibrium, the subgroups obtain equal total improvement for all  $\mathbf{w}^*$  if and only if

$$A_1^{-1} \Pi_1 A_1^{-1} = A_2^{-1} \Pi_2 A_2^{-1}.$$

*Proof.* Equal total outcome improvement across subgroups is guaranteed in equilibrium if and only

if the following holds:

$$\begin{aligned}
& \mathcal{I}_1(\mathbf{w}_{SW}) - \mathcal{I}_2(\mathbf{w}_{SW}) = 0 \Leftrightarrow && \text{(Definition 9.1)} \\
& \Leftrightarrow \left\langle A_1^{-1}\Pi_1 \frac{(A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)^\top \mathbf{w}^*}{\|(A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)^\top \mathbf{w}^*\|_2} - A_2^{-1}\Pi_2 \frac{(A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)^\top \mathbf{w}^*}{\|(A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)^\top \mathbf{w}^*\|_2}, \mathbf{w}^* \right\rangle = 0 \\
& \Leftrightarrow \left\langle A_1^{-1}\Pi_1 (A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)^\top \mathbf{w}^* - A_2^{-1}\Pi_2 (A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)^\top \mathbf{w}^*, \mathbf{w}^* \right\rangle = 0 \\
& \Leftrightarrow \left\langle \left[ A_1^{-1}\Pi_1 (A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)^\top - A_2^{-1}\Pi_2 (A_1^{-1}\Pi_1 + A_2^{-1}\Pi_2)^\top \right] \mathbf{w}^*, \mathbf{w}^* \right\rangle = 0 \\
& \Leftrightarrow \left\langle \left( A_1^{-1}\Pi_1 \Pi_1^\top A_1^{-1\top} + A_2^{-1}\Pi_2 \Pi_1^\top A_1^{-1\top} - A_1^{-1}\Pi_1 \Pi_2^\top A_2^{-1\top} - A_2^{-1}\Pi_2 \Pi_2^\top A_2^{-1\top} \right)^\top \mathbf{w}^*, \mathbf{w}^* \right\rangle = 0 \\
& \Leftrightarrow \left\langle \left( A_1^{-1}\Pi_1 A_1^{-1} + A_2^{-1}\Pi_2 \Pi_1 A_1^{-1\top} - A_1^{-1}\Pi_1 \Pi_2 A_2^{-1} - A_2^{-1}\Pi_2 A_2^{-1} \right)^\top \mathbf{w}^*, \mathbf{w}^* \right\rangle = 0
\end{aligned}$$

where the second transition is due to Lemma 9.4, the fourth is due to  $Av - Bv = (A - B)v$ , the second-to-last one is due to  $(AB)^\top = B^\top A^\top$  and  $A^{\top\top} = A$ , and the last one is due to the fact that  $\Pi_g = \Pi_g^\top$ ,  $\Pi_g^2 = \Pi_g$  and  $A_g^{-1} = A_g^{-1\top}$ . Let  $M \triangleq A_1^{-1}\Pi_1 A_1^{-1} + A_2^{-1}\Pi_2 \Pi_1 A_1^{-1\top} - A_1^{-1}\Pi_1 \Pi_2 A_2^{-1} - A_2^{-1}\Pi_2 A_2^{-1\top}$ , the above can be written as the quadratic form  $q(\mathbf{w}^*) = (\mathbf{w}^*)^\top M \mathbf{w}^*$ . In turn,  $q(\mathbf{w}^*) = 0$  simultaneously for all  $\mathbf{w}^*$ , i.e  $q = 0$ , if and only  $M$  is skew-symmetric, which means  $M + M^\top = 0$ .

This can be rewritten as

$$\begin{aligned}
0 &= A_1^{-1}\Pi_1 A_1^{-1} + A_2^{-1}\Pi_2 \Pi_1 A_1^{-1\top} - A_1^{-1}\Pi_1 \Pi_2 A_2^{-1} - A_2^{-1}\Pi_2 A_2^{-1} \\
&\quad + A_1^{-1}\Pi_1 A_1^{-1} + A_1^{-1}\Pi_1 \Pi_2 A_1^{-1\top} - A_2^{-1}\Pi_2 \Pi_1 A_1^{-1} - A_2^{-1}\Pi_2 A_2^{-1},
\end{aligned}$$

or equivalently

$$2A_1^{-1}\Pi_1 A_1^{-1} - 2A_2^{-1}\Pi_2 A_2^{-1} = 0.$$

This concludes the proof. ■

#### 9.4.3 PER-UNIT IMPROVEMENT

We begin by highlighting an important insight - the following proposition shows that, even when the difference between improvement across subgroups is arbitrarily large, the deployed scoring rule may still, in fact, induce optimal per-unit improvement within each group. We also note that

separately but similarly to the objective of equal true improvement, the fulfillment of the objective of optimal per-unit improvement across subgroups implies no negative externality.

**Proposition 9.3.** *Let  $\alpha > 0$  be arbitrarily small. In equilibrium we may see simultaneously:*

- *arbitrarily different total improvement across subgroups:  $\mathcal{I}_1(\mathbf{w}_{\text{SW}}) < \alpha \cdot \mathcal{I}_2(\mathbf{w}_{\text{SW}})$ .*
- *optimal per-unit improvement in all subgroups, i.e.,  $\mathbf{u}\mathcal{I}_g(\mathbf{w}_{\text{SW}}) = \mathbf{u}\mathcal{I}_g(\mathbf{w}_g), \forall g$ .*

*Proof.* We focus on a two-dimensional example for clarity of exposition. To abstract away from discrepancies in the cost matrices, we assume that  $A_1, A_2 = \mathbb{I}_{2 \times 2}$ , and that the projection matrices of the two subgroups are

$$\Pi_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \Pi_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Next, we select  $\varepsilon > 0$  such that  $\frac{\varepsilon^2}{1-\varepsilon^2} < \alpha$ , and assume that the distribution we face is such that the optimal outcome-decision rule is:  $\mathbf{w}^* = [\varepsilon, \sqrt{1-\varepsilon^2}]^\top$ . From Lemma 9.4, substituting the values of  $A_1, A_2, \Pi_1, \Pi_2$  as defined above, we have that the  $\mathbf{w}$  maximizing the social welfare satisfies:

$$\mathbf{w}_{\text{SW}} = \frac{\mathbf{w}^*}{\|\mathbf{w}^*\|_2} = \left[ \varepsilon, \sqrt{1-\varepsilon^2} \right]^\top.$$

Substituting  $A_1, A_2, \Pi_1, \Pi_2$  in  $\Delta_g(\mathbf{w}) = A_g^{-1} \Pi_g \mathbf{w}$  we have:  $\mathcal{I}_1(\mathbf{w}_{\text{SW}}) = \varepsilon^2$  and  $\mathcal{I}_2(\mathbf{w}_{\text{SW}}) = 1 - \varepsilon^2$ .

Next, we compute  $\mathbf{u}\mathcal{I}_1(\mathbf{w}_1)$  and  $\mathbf{u}\mathcal{I}_2(\mathbf{w}_2)$ . Substituting  $A_1, A_2, \Pi_1, \Pi_2$  in the definition of  $\Delta_g(\mathbf{w})$  and by Lemma 9.5, we have:  $\mathbf{w}_1 = [1, 0]^\top, \mathbf{w}_2 = [0, 1]^\top$ . Finally:

$$\mathbf{u}\mathcal{I}_1(\mathbf{w}) = \mathbf{u}\mathcal{I}_1 \left( \frac{\Pi_1 \left[ \varepsilon, \sqrt{1-\varepsilon^2} \right]^\top}{\left\| \Pi_1 \left[ \varepsilon, \sqrt{1-\varepsilon^2} \right]^\top \right\|_2} \right) = \mathbf{u}\mathcal{I}_1 \left( \frac{[\varepsilon, 0]^\top}{\left\| [\varepsilon, 0]^\top \right\|_2} \right) = \mathbf{u}\mathcal{I}_1 \left( [1, 0]^\top \right) = \varepsilon = \max_{\mathbf{w}'} \mathbf{u}\mathcal{I}_1(\mathbf{w}')$$

$$\begin{aligned} \mathbf{u}\mathcal{I}_2(\mathbf{w}) &= \mathbf{u}\mathcal{I}_1 \left( \frac{\Pi_2 \left[ \varepsilon, \sqrt{1-\varepsilon^2} \right]^\top}{\left\| \Pi_2 \left[ \varepsilon, \sqrt{1-\varepsilon^2} \right]^\top \right\|_2} \right) = \mathbf{u}\mathcal{I}_2 \left( \frac{\left[ 0, \sqrt{1-\varepsilon^2} \right]^\top}{\left\| \left[ 0, \sqrt{1-\varepsilon^2} \right]^\top \right\|_2} \right) = \mathbf{u}\mathcal{I}_2 \left( [0, 1]^\top \right) = \sqrt{1-\varepsilon^2} \\ &= \mathbf{u}\mathcal{I}_2(\mathbf{w}_2). \end{aligned}$$

However, for the total outcome improvement:  $\frac{\mathcal{I}_1(\mathbf{w})}{\mathcal{I}_2(\mathbf{w})} = \frac{\varepsilon^2}{1-\varepsilon^2} < \alpha$ , which concludes the proof. ■

Next, we identify two cases where we can guarantee optimal per-unit improvement. Intuitively, the first case occurs when the direction of the optimized solution in each subspace is not affected by the other subgroups, which happens when their information regarding the decision rule does not overlap.

**Proposition 9.4.** *In equilibrium, optimal per-unit improvement across subgroups is guaranteed when the subspaces  $\mathcal{S}_1, \mathcal{S}_2$  are orthogonal.*

The second case is when subgroups have the same information regarding the decision rule, and their feature modification costs are proportional to one another.

**Proposition 9.5.** *In equilibrium, optimal per-unit improvement across subgroups is guaranteed when the cost matrices are proportional to each other and  $\Pi_1 = \Pi_2$ .*

Propositions 9.4 and 9.5 can be proven as corollaries from the following general theorem, which derives the necessary and sufficient conditions for optimal per-unit improvement.

#### Theorem 9.4

In equilibrium, subgroup  $g$  gets optimal per-unit improvement if and only if:

$$\left\langle A_g^{-1} \frac{\Pi_g A_g^{-1} \mathbf{w}^*}{\|\Pi_g A_g^{-1} \mathbf{w}^*\|_2} - A_g^{-1} \frac{\Pi_g (\Pi_1 A_1^{-1} + \Pi_2 A_2^{-1}) \mathbf{w}^*}{\|\Pi_g (\Pi_1 A_1^{-1} + \Pi_2 A_2^{-1}) \mathbf{w}^*\|_2}, \mathbf{w}^* \right\rangle = 0.$$

*Proof.* Using Lemmas 9.4, 9.5 and Definition 9.1 we get that  $\mathbf{w}_{SW}$  induces optimal per-unit outcome

improvement if and only if:

$$\begin{aligned}
\mathbf{w}_{\text{SW}} = \mathbf{w}_g &= \arg \max_{\mathbf{w}'} \text{u}\mathcal{I}_g(\mathbf{w}') \Leftrightarrow \\
\Leftrightarrow \text{u}\mathcal{I}_g(\mathbf{w}_g) - \text{u}\mathcal{I}_g(\mathbf{w}) &= 0 \Leftrightarrow \\
\Leftrightarrow \mathcal{I}_g\left(\frac{\Pi_g \mathbf{w}_g}{\|\Pi_g \mathbf{w}_g\|_2}\right) - \mathcal{I}_g\left(\frac{\Pi_g \mathbf{w}}{\|\Pi_g \mathbf{w}\|_2}\right) &= 0 \quad (\text{Definition of } \text{u}\mathcal{I}_g(\cdot))
\end{aligned}$$

$$\Leftrightarrow \left\langle A_g^{-1} \Pi_g \frac{\frac{\Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^*}{\|(A_g^{-1} \Pi_g)^\top \mathbf{w}^*\|_2}}{\left\| \frac{\Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^*}{\|(A_g^{-1} \Pi_g)^\top \mathbf{w}^*\|_2} \right\|_2} - A_g^{-1} \Pi_g \frac{\frac{\Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*}{\|(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*\|_2}}{\left\| \frac{\Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*}{\|(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*\|_2} \right\|_2}, \mathbf{w}^* \right\rangle = 0 \quad (4)$$

$$\Leftrightarrow \left\langle A_g^{-1} \Pi_g \frac{\frac{\Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^*}{\|(A_g^{-1} \Pi_g)^\top \mathbf{w}^*\|_2}}{\left\| \frac{\Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^*}{\|(A_g^{-1} \Pi_g)^\top \mathbf{w}^*\|_2} \right\|_2} - A_g^{-1} \Pi_g \frac{\frac{\Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*}{\|(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*\|_2}}{\left\| \frac{\Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*}{\|(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*\|_2} \right\|_2}, \mathbf{w}^* \right\rangle = 0 \quad (5)$$

$$\Leftrightarrow \left\langle A_g^{-1} \Pi_g \frac{\frac{\Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^*}{\|\Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^*\|_2}}{\left\| \frac{\Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^*}{\|\Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^*\|_2} \right\|_2} - A_g^{-1} \Pi_g \frac{\frac{\Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*}{\|\Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*\|_2}}{\left\| \frac{\Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*}{\|\Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*\|_2} \right\|_2}, \mathbf{w}^* \right\rangle = 0$$

$$\Leftrightarrow \left\langle A_g^{-1} \Pi_g \frac{\frac{\Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^*}{\|\Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^*\|_2}}{\left\| \frac{\Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^*}{\|\Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^*\|_2} \right\|_2} - A_g^{-1} \Pi_g \frac{\frac{\Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*}{\|\Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*\|_2}}{\left\| \frac{\Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*}{\|\Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*\|_2} \right\|_2}, \mathbf{w}^* \right\rangle = 0 \quad (7)$$

where the transitions are given by: (4) Lemmas 9.4 and 9.5, (5)  $\|\frac{\mathbf{v}}{c}\|_2 = \frac{\|\mathbf{v}\|_2}{c}$  for any scalar  $c$ , and

(7)  $\Pi_g \Pi_g = \Pi_g$  as they are orthogonal projections.  $\blacksquare$

**Corollary 9.1.** *In equilibrium, optimal per-unit outcome improvement is guaranteed if there exists  $c_g > 0$ , such that:*

$$\Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^* = c_g \Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*.$$

*Proof.* Assume the condition in the statement holds and denote

$$v = \Pi_g(A_g^{-1} \Pi_g)^\top \mathbf{w}^* = c_g \Pi_g(A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*.$$

By Theorem 9.4, we know that for any subgroup  $g \in \{1, 2\}$ , we are guaranteed Optimal per-unit

outcome Improvement if and only if the following holds:

$$\left\langle A_g^{-1} \frac{\Pi_g (A_g^{-1} \Pi_g)^\top \mathbf{w}^*}{\|\Pi_g (A_g^{-1} \Pi_g)^\top \mathbf{w}^*\|_2} - A_g^{-1} \frac{\Pi_g (A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*}{\|\Pi_g (A_1^{-1} \Pi_1 + A_2^{-1} \Pi_2)^\top \mathbf{w}^*\|_2}, \mathbf{w}^* \right\rangle = 0.$$

Which is in our case equivalent to requiring

$$\left\langle A_g^{-1} \frac{v}{\|v\|_2} - A_g^{-1} \frac{\frac{v}{c_g}}{\left\| \frac{v}{c_g} \right\|_2}, \mathbf{w}^* \right\rangle = 0.$$

Equivalently, this can be written as

$$\left\langle A_g^{-1} \frac{v}{\|v\|_2} - A_g^{-1} \frac{v}{\|v\|_2}, \mathbf{w}^* \right\rangle = 0.$$

This concludes the proof. ■

## 9.5 EXPERIMENTS

Here, we empirically evaluate the impact of information disparities at equilibrium on two real-world datasets that pertain to our setting: the TAIWAN-CREDIT and ADULT datasets.<sup>\*\*</sup>

**Experimental Setup.** For TAIWAN-CREDIT  $d = 24$  and ADULT  $d = 14$ . In order to guarantee *numerical* (rather than categorical) feature values, we pre-processed the ADULT dataset to transform the categorical ones to integers. Specifically, for the features for which there was a *clear* hierarchical ordering (e.g., the “Education” feature, where we could order agents in terms of their highest education level reached), we reflected this ordering in the assignment of numerical values to these categories. For the TAIWAN-CREDIT dataset, no pre-processing was needed.

In both cases, we ran ERM in order to identify  $\mathbf{w}^*$  and we assumed that costs are  $A_1 = A_2 = I_{d \times d}$ . In Appendix E.3 we present additional experimental results for cost matrices  $A_1, A_2$  that differ from one another. After the pre-processing step, we created the groups of the population based on categories that intuitively “define” peer-networks. Our judgment for picking these categories

---

<sup>\*\*</sup><https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>,  
<https://archive.ics.uci.edu/ml/datasets/adult>.

Table 9.1: Groups for the TAIWAN-CREDIT dataset.

|                                  | <b>Age</b>        | <b>Education</b>     | <b>Marriage</b> |
|----------------------------------|-------------------|----------------------|-----------------|
| $G_1$                            | $\leq 25$ yrs old | gradschool & college | married         |
| $G_2$                            | $> 25$ yrs old    | high school          | not-married     |
| <b>diff(<math>w_{SW}</math>)</b> | 0.34              | 0.05                 | 0.23            |
| $r_{1,2}(w^*)$                   | 0.5               | 0.52                 | 0.48            |

Table 9.2: Groups for the ADULT dataset.

|                                  | <b>Age</b>        | <b>Country</b> | <b>Education</b>           |
|----------------------------------|-------------------|----------------|----------------------------|
| $G_1$                            | $\leq 35$ yrs old | western world  | degrees $\geq$ high school |
| $G_2$                            | $> 35$ yrs old    | everyone else  | degrees $<$ high-school    |
| <b>diff(<math>w_{SW}</math>)</b> | 0.15              | 0.66           | 0.20                       |
| $r_{1,2}(w^*)$                   | 0.29              | 0.89           | 0.77                       |

is based on folklore ideas about how people choose their network and social circles. For the TAIWAN-CREDIT dataset, we use the following categories: Age, Education, and Marriage, while for the ADULT dataset, we use: Age, Country, Education, and the final groups are in Tables 9.1 and 9.2. In order to obtain the projection matrices  $\Pi_1, \Pi_2$ , we ran SVD on the points inside of  $G_1, G_2$ . To be more precise, let  $X_i \in \mathbb{R}^{|G_i| \times d}$  be the matrix having as rows the vectors  $x^\top, \forall x \in G_i$ . Then, running SVD on  $X$  produces three matrices:  $X_g = U D V_g^\top$ , where  $V \in \mathbb{R}^{d \times r}$  and  $r = \text{rank}(X_g)$ . Let  $V_{g,5}$  correspond to the matrix having as columns the eigenvectors corresponding to the 5 top eigenvalues and zeroed out all other  $d - 5$  columns. Then, the projection matrix  $\Pi_g$  is defined as  $\Pi_g = V_{g,5} V_{g,5}^\top$ . Effectively, we focus the feature space on the directions corresponding the top 5 eigenvalues found in each group's data, as per traditional principal component analysis.

**Results.** In summary, our experimental results validate our theoretical analysis, and extend our insights to when the projection matrices do not satisfy the exact conditions required by the formal statements of Chapter 9.4.

First, we see that in both datasets, the principal's rule that optimizes the social welfare does not cause *any* negative externality when  $A_1 = A_2 = I_{d \times d}$  (that said, we do observe outcome deteriorations when the cost matrices differ from one another – see Appendix E.3). In fact, we observe *strict* improvement, i.e.,  $\mathcal{I}_g(w_{SW}) > 0$  for all groups  $g$ .

Second, neither the total nor the per-unit improvements are equal. In terms of total improvements, we in fact see *significant* disparities when the groups are defined based on their Age or their Marital Status in the TAIWAN-CREDIT dataset and based on every categorization in the ADULT dataset. These significant disparities for the particular groups we created match our intuition: we expect that people from significantly different age groups or countries to have very different understandings of the scoring rule, in turn leading to possibly very disparate total improvements.

We note also that in both datasets the difference in the total improvements of the groups is upper bounded by the overlap proxy (i.e.,  $\text{diff}(\mathbf{w}_{\text{SW}}) \leq r_{1,2}(\mathbf{w}^*)$ ), as expected from Lemma 9.6. That said, the gap between the two quantities can be rather large. This is because the magnitude of the overlap is *not* the only factor controlling  $\text{diff}(\mathbf{w}_{\text{SW}})$ . Rather, other factors (e.g., the direction of the overlap or how it compares to  $\mathbf{w}^*$ ) also matter significantly.

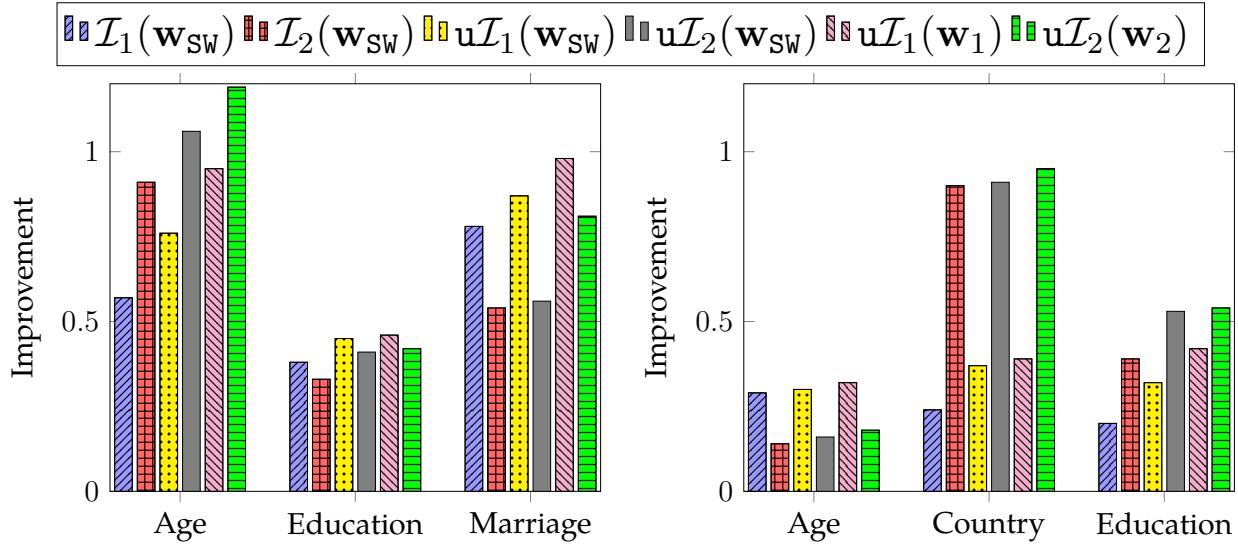


Figure 9.1: Left, Right: evaluation on the TAIWAN-CREDIT and ADULT dataset respectively. Tables 9.1 and 9.2 contain the characteristics of groups  $G_1, G_2$ . Recall that  $\mathcal{I}_g(\mathbf{w}_{\text{SW}})$ ,  $u\mathcal{I}_g(\mathbf{w}_{\text{SW}})$ , denote the total and per-unit improvement for group  $g$  in equilibrium respectively, while  $u\mathcal{I}_g(\mathbf{w}_g)$  denotes the optimal per-unit improvement for group  $g$ .

The *optimal per-unit* improvement can be very different across groups; an extreme example is the Country categorization in ADULT. It is surprising, however, that for Education in TAIWAN-CREDIT the optimal per-unit improvement is almost identical. Another interesting observation is that the *per-unit* improvement is close to (or almost the same as!) the optimal per-unit improvement for all groups in ADULT. We suspect that this is due to having very different projection matrices  $\Pi_1, \Pi_2$ .

## 9.6 DISCUSSION AND OPEN QUESTIONS

In this chapter, we have taken a first step towards modeling and developing a richer understanding of strategic learning and its implications in the context of inaccessible decision rules. Our results establish a connection between the information overlap and the ability to achieve cross-subgroup improvement, which seems to favor two extreme cases - either full informational discrepancy or full informational overlap accompanied by proportional modification costs. We further note that part of our resulting characterizations still holds even with fewer assumptions than we initially placed. Namely, we were able to give both do-no-harm and equal improvement guarantees which hold even under a complete lack of knowledge of the principal regarding the underlying scoring rule. Finally, our experiments complement our results by showing that under more general conditions on the informational overlap, significant disparities in improvements can arise across subgroups.

We discuss next some limitations of the results of the present chapter and avenues for future research. First, our model incorporates a linearity assumption regarding the form of the decision rule; we believe this linear assumption is a natural and simple choice for a first model that studies the phenomenon of information discrepancy, but an interesting future direction would be to understand which of our insights translate to a non-linear model, and what new insights arise. Second, we assume agents best-respond perfectly to the principal's choices.

While one can weaken the best-response assumption, this may affect the sharpness of our results. We note that some form of assumption regarding how agents respond to the model is natural and reflects many real-life situations. Finally, we take the point of view of a benevolent principal that aims to maximize social welfare, and discards any accuracy objective. We do so to study disparities in improvements in the “best” possible case for agents, when the principal’s incentives align with those of the agents. More broadly, we believe that this opens an avenue for future work on strategic machine learning research, by looking at the behavior of a principal that aims to jointly optimize the social welfare and his accuracy objective, and trades off true improvements to the label with the accuracy of the predicted labels. If the social welfare of the subgroups can increase and genuine effort can be incentivized for the price of a small accuracy loss, then scoring rules should strive for this *combined* desideratum, instead of just accuracy.

## **Part V**

# **Incentive-Aware Machine Learning**

## **Beyond Myopic Agents**

# 10

## The User Perspective: Learning to Bid without Knowing your Value

### 10.1 CHAPTER OVERVIEW

So far in this dissertation, we have taken the viewpoint of either the institution, or society. In this chapter we focus on online ad auctions and we take the perspective of the bidder/advertiser.

A standard assumption in the majority of the literature on auction theory and mechanism design (which we have also adopted in this dissertation apart from Chapter 7 where we allowed for irrational agents) is that participants that arrive in the market have a clear assessment of their valuation for the goods at sale. This assumption may seem acceptable in small markets with infrequent

auction occurrences and amplitude of time for participants to do market research on the goods. However, it is an assumption that is severely violated in the context of the digital economy.

In settings like online advertisement auctions or eBay auctions, bidders participate very frequently in auctions that they have very little knowledge about the good at sale, e.g. the value produced by a user clicking on an ad. It is unreasonable, therefore, to believe that the participant has a clear picture of this value. However, the inability to pre-assess the value of the good before arriving to the market is alleviated by the fact that due to the large volume of auctions in the digital economy, participants can employ *learning-by-doing* approaches.

In this chapter we address the question of *how would you learn to bid approximately optimally in a repeated auction setting where you do not know your value for the good at sale and where that value could potentially be changing over time*. The setting of learning in auctions with an unknown value poses an interesting interplay between exploration and exploitation that is not standard in the online learning literature: in order for the bidder to get feedback on her value she has to bid high enough to win the good with higher probability and hence, receive some information about that underlying value. However, the latter requires paying a higher price. Thus, there is an inherent trade-off between value-learning (exploration) and cost (exploitation). The main point of this chapter is to address the problem of learning how to bid in such unknown valuation settings with partial *win-only feedback*, so as to minimize regret with respect to the best fixed bid in hindsight.

Balancing the exploration-exploitation trade-off that is born, one can treat the problem as a Multi-Armed Bandit (MAB) problem, where each possible bid that the bidder could submit (e.g. any multiple of a cent between 0 and some upper bound on her value) is treated as an arm. Then, standard MAB algorithms (see e.g. ([Bubeck and Cesa-Bianchi, 2012](#)), and Chapter 2 in the present dissertation) can achieve regret rates that scale *linearly* with the number of such discrete bids. The latter can be very slow and does not leverage the structure of utilities and the form of partial feedback that arises in online auction markets. Recently, the authors in ([Weed et al., 2016](#)) addressed learning with such type of partial feedback in the context of repeated single-item second-price auctions. However, their approach does not address more complex auctions and is tailored to the second-price auction.

**Contributions.** This chapter addresses learning with a unique form of partial feedback in gen-

eral mechanism design environments. Importantly, we allow for randomized auctions with probabilistic outcomes, encompassing the case of sponsored search auctions, where the outcome of the mechanism (i.e., getting a click) is inherently randomized.

Our first main contribution is to introduce a novel online learning setting with partial feedback, which we call *learning with outcome-based feedback* and which could be of independent interest. We show that our setting captures online learning in many repeated auction scenarios including all types of single-item auctions, value-per-click sponsored search auctions, value-per-impression sponsored search auctions, and multi-item auctions.

Our setting generalizes the setting of learning with feedback graphs [Mannor and Shamir \(2011\)](#), [Alon et al. \(2013\)](#), in a way that is crucial for applying it to the auction settings of interest. At a high level, the auction setting is defined as follows: the learner chooses an action  $b \in B$  (e.g. a bid in an auction). The adversary chooses an *allocation function*  $x_t$ , that maps an action to a distribution over a set of potential outcomes  $O$  (e.g. the probability of getting a click) and a *reward function*  $r_t$  that maps an action-outcome pair to a reward (utility conditional on getting a click with a bid of  $b$ ). Then, an outcome  $o_t$  is chosen based on distribution  $x_t(b)$  and a reward  $r_t(b, o_t)$  is observed. The learner also gets to observe the function  $x_t$  and the reward function  $r_t(\cdot, o_t)$  for the realized outcome  $o_t$  (i.e. in our auction setting: she learns the probability of a click, the expected payment as a function of her bid and, *if she gets clicks*, her value).

Our second main contribution is an algorithm which we call WIN-EXP, which achieves regret  $O\left(\sqrt{T|O|\log(|B|)}\right)$ . The latter is inherently better than the generic multi-armed bandit regret of  $O\left(\sqrt{T|B|}\right)$ , since in most of our applications  $|O|$  will be a small constant (e.g.  $|O| = 2$  in sponsored search) and takes advantage of the particular feedback structure. Our algorithm is a variant of the EXP3 algorithm ([Auer et al., 2002b](#)), with a carefully crafted unbiased estimate of the utility of each action, which has lower variance than the unbiased estimate used in the standard EXP3 algorithm. This result could also be of independent interest and applicable beyond learning in auction settings. Our approach is similar to the importance weighted sampling approach used in EXP3 so as to construct unbiased estimates of the utility of each possible action. Our main technical insight is how to incorporate the allocation function feedback that the bidder receives to construct unbiased estimates with small variance, leading to dependence only in the number of outcomes

and not the number of actions.

This setting engulfs learning in many auctions of interest where bidders learn their value for a good only when they win the good and where the good which is allocated to the bidder is determined by some randomized allocation function. For instance, when applied to the case of single-item first-price, second-price or all-pay auctions, our setting corresponds to the case where the bidders observe their value for the item auctioned at each iteration only when they win the item. Moreover, after every iteration, they observe the critical bid they would have needed to submit to win (for instance, by observing the bids of others or the clearing price). The latter is typically the case in most government auctions or in settings like eBay.

Our flagship application is that of value-per-click sponsored search auctions. These are auctions where bidders repeatedly bid in an auction for a slot in a keyword impression on a search engine. The complexity of the sponsored search ecosystem and the large volume of repeated auctions has given rise to a plethora of automated bidding tools (see e.g. [Wordstream \(2018\)](#)) and has made sponsored search an interesting arena for automated learning agents. Our framework captures the fact that in this setting the bidders observe their value for a click only when they get clicked. Moreover, it assumes that the bidders also observe the average probability of click and the average cost per click for any bid they could have submitted. The latter is exactly the type of feedback that the automated bidding tools can receive via the use of *bid simulators* offered by both major search engines [Google \(2018a,b,c\)](#), [Microsoft \(2018\)](#). In Figure 10.1 we portray example interfaces from these tools, where we see that the bidders can observe exactly these allocation and payment curves assumed by our outcome-based-feedback formulation. Not using this information seems unreasonable and a waste of available information. This chapter shows how one can utilize this partial feedback given by the auction systems to provide improved learning guarantees over what would have been achieved if one took a fully bandit approach. In the experimental section, we also show that our approach outperforms that of the bandit approach even if the allocation and payment curves provided by the system have some error that could stem from errors in the machine learning models used in the calculation of these curves by the search engines. Hence, even when these curves are not fully reliable our approach can offer improvements in the learning rate.

We also extend our results to cases where the space of actions is a continuum (e.g. all bids in



Figure 10.1: Example interfaces of bid simulators of two major search engines, Google Adwords (left) and BingAds (right), that enables learning the allocation and the payment function. (sources [Standard \(2014\)](#), [Land \(2014\)](#))

an interval  $[0, 1]$ ). We show that in many auction settings, under appropriate assumptions on the utility functions, a regret of  $O\left(\sqrt{T \log(T)}\right)$  can be achieved by simply discretizing the action space to a sufficiently small uniform grid and running our WIN-EXP algorithm. This result encompasses the results of [Weed et al. \(2016\)](#) for second price auctions, learning in first-price and all-pay auctions, as well as learning in sponsored search with smoothness assumptions on the utility function. We also show how smoothness of the utility can easily arise due to the inherent randomness that exists in the mechanism run in sponsored search.

Finally, we provide two further extensions: *switching regret* and *feedback-graphs over outcomes*. The former adapts our algorithm to achieve low regret against a sequence of bids rather than a fixed bid. The latter has implications on faster convergence to approximate efficiency of the outcome (price of anarchy). Feedback graphs address the idea that in many cases the learner could be receiving information about other items other than the item he won (through correlations in the values for these items). This essentially corresponds to adding a feedback graph over outcomes and when outcome  $o_t$  is chosen, then the learner learns the reward function  $r_t(\cdot, o)$  for all neighboring outcomes  $o$  in the feedback graph. We provide improved results that mainly depend on the dependence number of the graph rather than the number of possible outcomes.

### 10.1.1 RELATED WORK

The results of this chapter lie on the intersection of two main areas: No regret learning in Game Theory and Mechanism Design and Contextual Bandits.

**NO REGRET LEARNING IN GAME THEORY AND MECHANISM DESIGN.** No regret learning has received a lot of attention in the Game Theory and Mechanism Design literature ([Chawla et al., 2014](#)). Most of the existing literature, however, focuses on the problem from the side of the auctioneer, who tries to maximize revenue through repeated rounds without knowing a priori the valuations of the bidders ([Amin et al., 2015, 2014](#), [Blum et al., 2004, 2015](#), [Cesa-Bianchi et al., 2015](#), [Cole and Roughgarden, 2014](#), [Dhangwatnotai et al., 2015](#), [Kanoria and Nazerzadeh, 2014](#), [Mohri and Munoz, 2014](#), [Ostrovsky and Schwarz, 2011](#), [Munoz and Vassilvitskii, 2017](#), [Feldman et al., 2016](#), [Koren et al., 2017](#)). These works are centered around different auction formats like the sponsored search ad auctions, the pricing of inventory and the single-item auctions. This chapter is mostly related to [Weed et al. \(2016\)](#), who adopt the point of view of the bidders in repeated second-price auctions and who also analyze the case when the true valuation of the item is revealed to the bidders only when they win the item. Their setting falls into the family of settings for which our novel and generic WIN-EXP algorithm produces good regret bounds and as a result, we are able to fully retrieve the regret that their algorithms yield, up to a tiny increase in the constants. Hence, we give an easier way to recover their results. Closely related to the results of this chapter are the works of [Dikkala and Tardos \(2013\)](#) and ([Balseiro and Gur, 2019](#)). [Dikkala and Tardos \(2013\)](#) analyze a setting where bidders have to experiment in order to learn their valuations, and show that the seller can increase revenue by offering an initial credit to them, in order to give them incentives to experiment. [Balseiro and Gur \(2019\)](#) introduce a family of dynamic bidding strategies in repeated second-price auctions, where advertisers adjust their bids throughout the campaign. They analyze both regret minimization and market stability. There are two key differences from our setting; first, Balseiro and Gur consider the case where the goal of the bidders is the expenditure rate in a way that guarantees that the available campaign budget will be spent in an optimal *pacing* way and second, because of their target being the expenditure rate at every round  $t$ , they assume that the bidders get information about the value of the slot being auctioned and based on this information they decide how to adjust their bid. Moreover, several works analyze the properties of auctions when bidders adopt a no-regret learning strategy ([Blum et al., 2008](#), [Caragiannis et al., 2015](#), [Roughgarden, 2009](#)). None of these works, however, addresses the question of learning more efficiently in the unknown valuation model and either invokes generic MAB algorithms or develops tailored full information

algorithms when the bidder knows his value. Another line of research takes a Bayesian approach to learn in repeated auctions and makes large market assumptions, analyzing learning to bid with an unknown value under a Mean Field Equilibrium condition ([Adlakha and Johari, 2013](#), [Iyer et al., 2011](#), [Balseiro et al., 2015](#))<sup>\*</sup>.

**LEARNING WITH PARTIAL FEEDBACK.** This chapter is also related to the literature in *learning with partial feedback* [Agarwal et al. \(2014\)](#), [Bubeck et al. \(2012\)](#). To establish this connection we observe that the *policies* and the *actions* in contextual bandit terminology translate into *discrete bids* and *groups of bids for which we learn the rewards* in our this chapter. The difference between these two is the fact that for each *action* in contextual bandits we get a single reward, whereas for our setting we observe a *group* of rewards; one for each action in the group. Moreover, the fact that we allow for randomized outcomes adds extra complication, non existent in contextual bandits. In addition, this chapter is closely related to the literature in *online learning with feedback graphs* ([Alon et al., 2015](#), [2013](#), [Cohen et al., 2016](#), [Mannor and Shamir, 2011](#)). In fact, we propose a new setting in online learning, namely, *learning with outcome-based feedback*, which is a generalization of learning with feedback graphs and is essential when applied to a variety of auctions which include sponsored search, single-item second-price, single-item first-price and single-item all-pay auctions. Moreover, the fact that the learner only learns the probability of each outcome and not the actual realization of the randomness, is similar in nature to a feedback graph setting, but where the bidder does not observe the whole graph. Rather, he observes a distribution over feedback graphs and for each bid he learns with what probability each feedback graph would arise. For concreteness, consider the case of sponsored search and suppose for now that the bidder gets even more information than what we assume and also observes the bids of his opponents. He still does not observe whether he would get a click if he falls on the slot below but only the probability with which he would get a click in the slot below. If he could observe whether he would still get a click in the slot below, then we could in principle construct a feedback graph that would say that for all bids were the bidder gets a slot his reward is revealed, and for every bid that he does not get a click, his reward is not revealed. However, this is not the structure that we have and essentially this corresponds to the

---

<sup>\*</sup>No-regret learning is complementary and orthogonal to the mean field approach, as it does not impose any stationarity assumption on the evolution of valuations of the bidder or the behavior of his opponents.

case where the feedback graph is not revealed, as analyzed in Cohen et al. (2016) and for which no improvement over full bandit feedback is possible. However, we show that this impossibility is amended by the fact that the learner observes the probability of a click and hence for each possible bid, he observes the probability with which each feedback graph would have happened. This is enough for a low variance unbiased estimate.

## 10.2 MODEL AND PRELIMINARIES

For simplicity of exposition, we start with a simple single-dimensional mechanism design setting, but our results extend to multi-dimensional (multi-item) mechanisms, as we will see in Section 10.4. Let  $n$  be the number of bidders. Each bidder has a value  $v_i \in [0, 1]$  *per-unit of a good* and submits a bid  $b_i \in B$ , where  $B$  is a discrete set of bids (e.g. a uniform  $\varepsilon$ -grid of  $[0, 1]$ ). Given the bid profile of all bidders, the auction allocates a unit of the good to the bidders. The allocation rule for bidder  $i$  is given by  $X_i(b_i, b_{-i})$ . Moreover, the mechanism defines a per-unit payment function  $P_i(b_i, b_{-i}) \in [0, 1]$ . The overall utility of the bidder is quasi-linear, i.e.  $u_i(b_i, b_{-i}) = (v_i - P_i(b_i, b_{-i})) \cdot X_i(b_i, b_{-i})$ .

**ONLINE LEARNING WITH PARTIAL FEEDBACK.** The bidders participate in this mechanism repeatedly. At each iteration, each bidder has some value  $v_{it}$  and submits a bid  $b_{it}$ . The mechanism has some time-varying allocation function  $X_{it}(\cdot, \cdot)$  and payment function  $P_{it}(\cdot, \cdot)$ . We assume that the bidder does *not* know his value  $v_{it}$ , nor the bids of his opponents  $b_{it}$ , nor the allocation and payment functions, *before* submitting a bid.

At the end of each iteration, he gets an item with probability  $X_{it}(b_{it}, b_{-i,t})$  and observes his value  $v_{it}$  for the item only when he gets one (e.g. in sponsored search, the good allocated is the probability of getting clicked, and you only observe your value if you get clicked). Moreover, we assume that he gets to observe his allocation and payment functions for that iteration, i.e. he gets to observe two functions  $x_{it}(\cdot) = X_{it}(\cdot, b_{-i,t})$  and  $p_{it}(\cdot) = P_{it}(\cdot, b_{-i,t})$ . Finally, he receives utility  $(v_{it} - p_{it}(b_{it})) \cdot \mathbf{1}\{\text{item is allocated to him}\}$  or in other words expected utility  $u_{it}(b_{it}) = (v_{it} - p_{it}(b_{it})) \cdot x_{it}(b_{it})$ . Given that we focus on learning from the perspective of a single bidder we will drop the index  $i$  from all notation and instead write,  $x_t(\cdot)$ ,  $p_t(\cdot)$ ,  $u_t(\cdot)$ ,  $v_t$ , etc. The goal of the bidder is to achieve small expected regret with respect to any fixed bid in hindsight:  $R(T) =$

$$\sup_{b^* \in B} \mathbb{E} \left[ \sum_{t=1}^T (u_t(b^*) - u_t(b_t)) \right] \leq o(T).$$

### 10.3 ABSTRACTION: LEARNING WITH WIN-ONLY FEEDBACK

We abstract further the learner's problem to a setting that could be of interest beyond auctions.

**LEARNING WITH WIN-ONLY FEEDBACK.** Every day a learner picks an action  $b_t$  from a finite set  $B$ . The adversary chooses a reward function  $r_t : B \rightarrow [-1, 1]$  and an allocation function  $x_t : B \rightarrow [0, 1]$ . The learner wins a reward  $r_t(b)$  with probability  $x_t(b)$ . Let  $u_t(b) = r_t(b)x_t(b)$  be the learner's expected utility from action  $b$ . After each iteration, if he won the reward then he learns the whole reward function  $r_t(\cdot)$ , while he *always* learns the allocation function  $x_t(\cdot)$ .

*Can the learner achieve regret  $O(\sqrt{T \log(|B|)})$  rather than bandit-feedback regret  $O(\sqrt{T|B|})$ ?*

In the case of the auction learning problem, the reward function  $r_t(b)$  takes the parametric form  $r_t(b) = v_t - p_t(b)$  and the learner needs to learn  $v_t$  and  $p_t(\cdot)$  at the end of each iteration, when he wins the item. This is inline with the feedback structure we described in the previous section.

We consider the following adaptation of the EXP3 algorithm with unbiased estimates based on the information received. It is also notationally useful throughout the section to denote with  $A_t$  the event of *winning a reward at time t*. Then, we can write:  $\Pr[A_t | b_t = b] = x_t(b)$  and  $\Pr[A_t] = \sum_{b \in B} \pi_t(b)x_t(b)$ , where with  $\pi_t(\cdot)$  we denote the multinomial distribution from which bid  $b$  is drawn. With this notation we define our WIN-EXP algorithm in Algorithm 10.1. We note here that our generic family of the WIN-EXP algorithms can be parametrized by the step-size  $\eta$ , the estimate of the utility  $\tilde{u}_t$  that the learner gets at each round and the feedback structure that he receives.

**BOUNDRING THE REGRET.** We first bound the first and second moment of the unbiased estimates built at each iteration in the WIN-EXP algorithm.

**Lemma 10.1.** *At each iteration  $t$ , for any action  $b \in B$ , the random variable  $\tilde{u}_t(b)$  is an unbiased estimate of the true expected utility  $u_t(b)$ , i.e.:  $\forall b \in B : \mathbb{E}[\tilde{u}_t(b)] = u_t(b) - 1$  and has expected second moment bounded by:  $\forall b \in B : \mathbb{E}[(\tilde{u}_t(b))^2] \leq \frac{4\Pr[A_t | b_t=b]}{\Pr[A_t]} + \frac{\Pr[\neg A_t | b_t=b]}{\Pr[\neg A_t]}$ .*

---

**Algorithm 10.1: WIN-EXP algorithm for learning with win-only feedback**


---

```

1 Let  $\pi_1(b) = \frac{1}{|B|}$  for all  $b \in B$  (i.e. the uniform distribution over bids),  $\eta = \sqrt{\frac{2 \log(|B|)}{5T}}$ .
2 for each iteration  $t$  do
3   Draw a bid  $b_t$  from the multinomial distribution based on  $\pi_t(\cdot)$ .
4   Observe  $x_t(\cdot)$  and if reward is won also observe  $r_t(\cdot)$ .
5   Compute estimate of utility:
6   if reward is won then
7     
$$\tilde{u}_t(b) = \frac{(r_t(b)-1)\Pr[A_t|b_t=b]}{\Pr[A_t]}$$

8   else
9     
$$\tilde{u}_t(b) = -\frac{\Pr[\neg A_t|b_t=b]}{\Pr[\neg A_t]}$$

10  Update  $\pi_t(\cdot)$  as in Exponential Weights Update:  $\forall b \in B : \pi_{t+1}(b) \propto \pi_t(b) \cdot \exp \{ \eta \cdot \tilde{u}_t(b) \}$ .

```

---

*Proof.* Let  $A_t$  denote the event that the reward was won. We have:

$$\begin{aligned}
\mathbb{E} [\tilde{u}_t(b)] &= \mathbb{E} \left[ \frac{(r_t(b) - 1) \cdot \Pr[A_t|b_t=b]}{\Pr[A_t]} \mathbf{1}\{A_t\} - \frac{\Pr[\neg A_t|b_t=b]}{\Pr[\neg A_t]} \mathbf{1}\{\neg A_t\} \right] \\
&= (r_t(b) - 1) \Pr[A_t|b_t=b] - \Pr[\neg A_t|b_t=b] \\
&= r_t(b) \Pr[A_t|b_t=b] - 1 = u_t(b) - 1
\end{aligned}$$

Similarly for the second moment:

$$\begin{aligned}
\mathbb{E} [\tilde{u}_t(b)^2] &= \mathbb{E} \left[ \frac{(r_t(b) - 1)^2 \cdot \Pr[A_t|b_t=b]^2}{\Pr[A_t]^2} \mathbf{1}\{A_t\} + \frac{\Pr[\neg A_t|b_t=b]^2}{\Pr[\neg A_t]^2} \mathbf{1}\{\neg A_t\} \right] \\
&= \frac{(r_t(b) - 1)^2 \cdot \Pr[A_t|b_t=b]^2}{\Pr[A_t]} + \frac{\Pr[\neg A_t|b_t=b]^2}{\Pr[\neg A_t]} \leq \frac{4\Pr[A_t|b_t=b]}{\Pr[A_t]} + \frac{\Pr[\neg A_t|b_t=b]}{\Pr[\neg A_t]}
\end{aligned}$$

where the last inequality holds since  $r_t(\cdot) \in [-1, 1]$  and  $x_t(\cdot) \in [0, 1]$ . ■

We are now ready to prove our main theorem:

**Theorem 10.1: Regret of WIN-EXP**

The regret of the WIN-EXP algorithm with the aforementioned unbiased estimates and step size  $\sqrt{\frac{2 \log(|B|)}{5T}}$  is:  $4\sqrt{T \log(|B|)}$ .

*Proof.* Observe that regret with respect to utilities  $u_t(\cdot)$  is equal to regret with respect to the translated utilities  $u_t(\cdot) - 1$ . We use the fact that the exponential weights update with an unbiased

estimate  $\tilde{u}_t(\cdot) \leq 0$  of the true utilities, achieves expected regret of the form<sup>†</sup>:

$$R(T) \leq \frac{\eta}{2} \sum_{t=1}^T \sum_{b \in B} \pi_t(b) \cdot \mathbb{E} [(\tilde{u}_t(b))^2] + \frac{1}{\eta} \log(|B|)$$

Invoking the bound on the second moment by Lemma 10.1, we get:

$$R(T) \leq \frac{\eta}{2} \sum_{t=1}^T \sum_{b \in B} \pi_t(b) \cdot \left( \frac{4\Pr[A_t | b_t = b]}{\Pr[A_t]} + \frac{\Pr[\neg A_t | b_t = b]}{\Pr[\neg A_t]} \right) + \frac{1}{\eta} \log(|B|) \leq \frac{5}{2}\eta T + \frac{1}{\eta} \log(|B|)$$

Picking  $\eta = \sqrt{\frac{2\log(|B|)}{5T}}$ , we get the theorem. ■

#### 10.4 BEYOND BINARY OUTCOMES: OUTCOME-BASED FEEDBACK

In the win-only feedback framework there were two possible outcomes that could happen: either you win the reward or not. We now consider a more general problem, where there are more than two outcomes and you learn your reward function for the outcome you won. Moreover, the outcome that you won is also a probabilistic function of your action.

**LEARNING WITH OUTCOME-BASED FEEDBACK.** Every day a learner picks an action  $b_t$  from a finite set  $B$ . There is a set of payoff-relevant outcomes  $O$ . The adversary chooses a reward function  $r_t : B \times O \rightarrow [-1, 1]$ , which maps an action and an outcome to a reward and he also chooses an allocation function  $x_t : B \rightarrow \Delta(O)$ , which maps an action to a distribution over the outcomes. Let  $x_t(b, o)$  be the probability of outcome  $o$  under action  $b$ . An outcome  $o_t \in O$  is chosen based on distribution  $x_t(b_t)$ . The learner wins reward  $r_t(b_t, o_t)$  and observes the whole outcome-specific reward function  $r_t(\cdot, o_t)$ . He *always* learns the allocation function  $x_t(\cdot)$  after the iteration. Let  $u_t(b) = \sum_{o \in O} r_t(b, o) \cdot x_t(b, o)$  be the expected utility from action  $b$ .

We consider the following adaptation of the EXP3 algorithm with unbiased estimates based on the information received. It is also notationally useful throughout the section to consider  $o_t$  as the random variable of the outcome chosen at time  $t$ . Then, we can write:  $\Pr_t[o_t | b] = x_t(b, o_t)$  and  $\Pr_t[o_t] = \sum_{b \in B} \pi_t(b) \Pr_t[o_t | b] = \sum_{b \in B} \pi_t(b) \cdot x_t(b, o_t)$ . With this notation and based on the

---

<sup>†</sup>A detailed proof of this claim can be found in Appendix F.2.

feedback structure, we define our WIN-EXP algorithm for learning with outcome-based feedback in Algorithm 10.2.

---

**Algorithm 10.2: WIN-EXP algorithm for learning with outcome-based feedback**


---

1 Let  $\pi_1(b) = \frac{1}{|B|}$  for all  $b \in B$  (i.e. the uniform distribution over bids),  $\eta = \sqrt{\frac{\log(|B|)}{2T|O|}}$ .

2 **for each iteration  $t$  do**

3     Draw an action  $b_t$  from the multinomial distribution based on  $\pi_t(\cdot)$ .

4     Observe  $x_t(\cdot)$ , observe chosen outcome  $o_t$  and associated reward function  $r_t(\cdot, o_t)$ .

5     Compute estimate of utility:

$$\tilde{u}_t(b) = \frac{(r_t(b, o_t) - 1)\Pr_t[o_t|b]}{\Pr_t[o_t]} \quad (10.4.1)$$

6     Update  $\pi_t(\cdot)$  based on the Exponential Weights Update:

$$\forall b \in B : \pi_{t+1}(b) \propto \pi_t(b) \cdot \exp\{\eta \cdot \tilde{u}_t(b)\} \quad (10.4.2)$$


---

**Theorem 10.2: Regret of WIN-EXP with outcome-based feedback**

The regret of Algorithm 10.2 with  $\tilde{u}_t(b) = \frac{(r_t(b, o_t) - 1)\Pr_t[o_t|b]}{\Pr_t[o_t]}$  and  $\eta = \sqrt{\frac{\log(|B|)}{2T|O|}}$  is:

$$R(T) \leq 2\sqrt{2T|O|\log(|B|)}.$$

We first give a lemma that bounds the moments of our utility estimate.

**Lemma 10.2.** At each iteration  $t$ , for any action  $b \in B$ , the random variable  $\tilde{u}_t(b)$  is an unbiased estimate of the true expected utility  $u_t(b)$ , i.e.:  $\forall b \in B : \mathbb{E}[\tilde{u}_t(b)] = u_t(b) - 1$  and has expected second moment bounded by:  $\forall b \in B : \mathbb{E}[(\tilde{u}_t(b))^2] \leq 4 \sum_{o \in O} \frac{\Pr_t[o|b]}{\Pr_t[o]}$ .

*Proof of Lemma 10.2.* According to the notation we introduced before we have:

$$\begin{aligned} \mathbb{E}[\tilde{u}_t(b)] &= \mathbb{E}_{o_t} \left[ \frac{(r_t(b, o_t) - 1) \cdot \Pr_t[o_t|b]}{\Pr_t[o_t]} \right] = \sum_{o \in O} \frac{(r_t(b, o) - 1) \cdot \Pr_t[o|b]}{\Pr_t[o]} \Pr_t[o] \\ &= \sum_{o \in O} r_t(b, o) \Pr_t[o|b] - 1 = u_t(b) - 1 \end{aligned}$$

Similarly for the second moment:

$$\begin{aligned}\mathbb{E} [\tilde{u}_t(b)^2] &\leq \mathbb{E}_{o_t} \left[ \frac{(r_t(b, o_t) - 1)^2 \Pr_t[o|b]^2}{\Pr_t[o_t]^2} \right] = \sum_{o \in O} \frac{(r_t(b, o) - 1)^2 \Pr_t[o|b]^2}{\Pr_t[o]^2} \Pr_t[o] \\ &\leq \sum_{o \in O} \frac{4 \Pr_t[o|b]}{\Pr_t[o]}\end{aligned}$$

where the last inequality holds since  $r_t(\cdot, \cdot) \in [-1, 1]$ . ■

*Proof of Theorem 10.2.* Observe that regret with respect to utilities  $u_t(\cdot)$  is equal to regret with respect to the translated utilities  $u_t(\cdot) - 1$ . We use the fact that the exponential weight updates with an unbiased estimate  $\tilde{u}_t(\cdot) \leq 0$  of the true utilities, achieves expected regret of the form:

$$R(T) \leq \frac{\eta}{2} \sum_{t=1}^T \sum_{b \in B} \pi_t(b) \cdot \mathbb{E} [(\tilde{u}_t(b))^2] + \frac{1}{\eta} \log(|B|)$$

For a detailed proof of the above, we refer the reader to Appendix F.2. Invoking the bound on the second moment by Lemma 10.2, we get:

$$\begin{aligned}R(T) &\leq 2\eta \sum_{t=1}^T \sum_{b \in B} \pi_t(b) \cdot \sum_{o \in O} \frac{\Pr_t[o|b]}{\Pr_t[o]} + \frac{1}{\eta} \log(|B|) \\ &= 2\eta \sum_{t=1}^T \sum_{o \in O} \sum_{b \in B} \pi_t(b) \cdot \frac{\Pr_t[o|b]}{\Pr_t[o]} + \frac{1}{\eta} \log(|B|) \\ &\leq 2\eta T |O| + \frac{1}{\eta} \log(|B|)\end{aligned}$$

Picking  $\eta = \sqrt{\frac{\log(|B|)}{2T|O|}}$ , we get the theorem. ■

**APPLICATIONS TO LEARNING IN AUCTIONS.** We present a series of applications of the result of this section to several learning in auction settings, even beyond single-item or single-dimensional ones.

**Example 10.1** (Second-price auction). *Suppose that the mechanism ran at each iteration is just the second price auction. Then, we know that the allocation function  $X_i(b_i, b_{-i})$  is simply of the form:  $\mathbb{1}\{b_i \geq \max_{j \neq i} b_j\}$  and the payment function is simply the second highest bid. In this case, observing the allocation and payment functions at the end of the auction simply boils down to observing the highest other bid. In fact, in this case we have a trivial setting where the bidder gets an allocation of either 0 or 1 and if we*

let  $B_t = \max_{j \neq i} b_{jt}$ , then the unbiased estimate of the utility takes the simpler form (assuming the bidder always loses in case of ties) of:  $\tilde{u}_t(b) = \frac{(v_{it} - B_t - 1)\mathbb{1}\{b > B_t\}}{\sum_{b' > B_t} \pi_t(b')}$  if  $b_t > B_t$  and  $\tilde{u}_t(b) = \frac{\mathbb{1}\{b \leq B_t\}}{\sum_{b' \leq B_t} \pi_t(b')}$  in any other case. Our main theorem gives regret  $R(T) \leq \sqrt{T \log(|B|)}$ . We note that this theorem recovers exactly the results of [Weed et al. \(2016\)](#), by simply using as  $B$  a uniform  $1/\Delta^o$  discretization of the bidding space, for an appropriately defined constant  $\Delta^o$ .

**COMPARISON WITH RESULTS IN WEED ET AL. (2016)** We note that our result in Example 10.1 also recovers the results of [Weed et al. \(2016\)](#), who work in the continuous bid setting (i.e.  $b \in [0, 1]$ ). In order to describe their results, consider the grid  $\mathcal{L}_T$  formed by the maximum bids from other bidders  $m_t = \max_{j \neq i} b_{jt}$  for all the rounds. Let  $l^o = (m_t, m_{t'})$  be the widest interval in  $\mathcal{L}_T$ , that contains an optimal fixed bid in hindsight and let  $\Delta^o$  denote its length. [Weed et al. \(2016\)](#) provide an algorithm for learning the valuation, which yields regret  $4\sqrt{T \log(1/\Delta^o)}$ .

The same regret can be achieved, if we simply consider a partition of the bidding space  $[0, 1]$  into  $\frac{1}{\varepsilon}$  intervals of equal length  $\varepsilon$ , for  $\varepsilon < \Delta^o$ , and run our algorithm on this discretized bid space  $B$ . If  $l^o$  contains an optimal bid, then any bid  $b \in l^o$  is also optimal in-hindsight, since all such bids achieve the same utility. Since  $\Delta^o > \varepsilon$ , there must exist a discretized bid  $b_\varepsilon^* \in B \cap l^o$ . Thus,  $b_\varepsilon^*$  is also optimal in hindsight. Hence, regret against the best fixed bid in  $[0, 1]$  is equal to regret against the best fixed discretized bid in  $B$ . By our Theorem 10.2, the latter regret is  $4\sqrt{T \log(1/\varepsilon)}$ , which can be made arbitrarily close to the regret bound achieved by [Weed et al. \(2016\)](#), who use a more intricate adaptive discretization. Similar to [Weed et al. \(2016\)](#), knowledge of  $\Delta^o$  can be bypassed by instead defining  $\Delta^o$  as the length of the smallest interval in  $\mathcal{L}_T$  and then using the standard doubling trick, i.e.: keep an estimate of  $\Delta^o$  and once this estimate is violated, divide  $\Delta^o$  in half and re-start your algorithm. The latter only increases the regret by a constant factor.

**Example 10.2** (Value-per-click auctions). *This is a variant of the binary outcome case analyzed in Section 10.3, where  $O = \{A, \neg A\}$ , i.e. get clicked or not. Hence,  $|O| = 2$ , and  $r_t(b, A) = v_t - p_t(b)$ , while  $r_t(b, \neg A) = 0$ . Our main theorem gives regret  $R(T) \leq 4\sqrt{T \log(|B|)}$ .*

**Example 10.3** (Unit-demand multi-item auctions). *Consider the case of  $K$  items at an auction where the bidder has value  $v_k$  for only one item  $k$ . Given a bid  $b$ , the mechanism defines a probability distribution over the items that the bidder will be allocated and also defines a payment function, which depends on the bid*

of the bidder and the item allocated. When a bidder gets allocated an item  $k$  he gets to observe his value  $v_{kt}$  for that item. Thus, the set of outcomes is equal to  $O = \{1, \dots, K+1\}$ , with outcome  $K+1$  associated with not getting any item. The rewards are also of the form:  $r_t(b, k) = v_{kt} - p_t(b, k)$  for some payment function  $p_t(b, k)$  dependent on the auction format. Our main theorem then gives regret  $2\sqrt{2(K+1)T \log(|B|)}$ .

#### 10.4.1 BATCH REWARDS PER-ITERATION AND SPONSORED SEARCH AUCTIONS

We now consider the case of sponsored search auctions, where the learner participates in multiple auctions per-iteration. At each of these auctions he has a chance to win and get feedback on his value. To this end, we abstract the *learning with win-only feedback* setting to a setting where multiple rewards are awarded per-iteration. The allocation function remains the same throughout the iteration but the reward functions can change.

**OUTCOME-BASED FEEDBACK WITH BATCH REWARDS.** Every iteration  $t$  is associated with a set of *reward contests*  $I_t$ . The learner picks an action  $b_t$ , which is used at *all* reward contests. For each  $\tau \in I_t$  the adversary picks an outcome specific reward function  $r_\tau : B \times O \rightarrow [-1, 1]$ . Moreover, the adversary chooses an allocation function  $x_t : B \rightarrow \Delta(O)$ , which is not  $\tau$ -dependent. At each  $\tau$ , an outcome  $o_\tau$  is chosen based on distribution  $x_t(b_t)$  and independently. The learner receives reward  $r_\tau(b_t, o_\tau)$  from that contest. The overall realized utility from that iteration is the average reward:  $\frac{1}{|I_t|} \sum_{\tau \in I_t} r_\tau(b_t, o_\tau)$ , while the expected utility from any bid  $b$  is:  $u_t(b) = \frac{1}{|I_t|} \sum_{\tau \in I_t} \sum_{o \in O} r_\tau(b, o) \cdot x_t(b, o)$ . We assume that at the end of each iteration the learner receives as feedback the average reward function conditional on each realized outcome, i.e. if we let  $I_{to} = \{\tau \in I_t : o_\tau = o\}$ , then the learner learns the function:  $Q_t(b, o) = \frac{1}{|I_{to}|} \sum_{\tau \in I_{to}} r_\tau(b, o)$  (with the convention that  $Q_t(b, o) = 0$  if  $|I_{to}| = 0$ ) as well as the realized frequencies  $f_t(o) = \frac{|I_{to}|}{|I_t|}$  for all outcomes  $o$ .

With this at hand we can define the *batch-analogue* of our unbiased estimates of the previous section. To avoid any confusion we define:  $\Pr_t[o|b] = x_t(b, o)$  and  $\Pr_t[o] = \sum_{b \in B} \pi_t(b) \Pr_t[o|b]$ , to denote that these probabilities only depend on  $t$  and not on  $\tau$ . The estimate of the utility will be:

$$\tilde{u}_t(b) = \sum_{o \in O} \frac{\Pr_t[o|b]}{\Pr_t[o]} f_t(o) (Q_t(b, o) - 1) \quad (10.4.3)$$

We show the full algorithm with outcome-based batch-reward feedback in Algorithm 10.3.

---

**Algorithm 10.3: WIN-EXP algorithm for learning with outcome-based batch-reward feedback**


---

- 1 Let  $\pi_1(b) = \frac{1}{|B|}$  for all  $b \in B$  (i.e. the uniform distribution over bids),  $\eta = \sqrt{\frac{\log(|B|)}{2T|O|}}$ .
- 2 **for each iteration  $t$  do**
- 3     Draw an action  $b_t$  from the multinomial distribution based on  $\pi_t(\cdot)$ .
- 4     Observe  $x_t(\cdot)$ , chosen outcomes  $o_\tau, \forall \tau \in I_t$ , average reward function conditional on each realized outcome  $Q_t(b, o)$  and the realized frequencies for each outcome  $f_t(o) = \frac{|I_{to}|}{|I_t|}$ .
- 5     Compute estimate of utility:

$$\tilde{u}_t(b) = \sum_{o \in O} \frac{\Pr_t[o|b]}{\Pr_t[o]} f_t(o) (Q_t(b, o) - 1) \quad (10.4.4)$$

- 6 Update  $\pi_t(\cdot)$  based on the Exponential Weights Update:

$$\forall b \in B : \pi_{t+1}(b) \propto \pi_t(b) \cdot \exp \{ \eta \cdot \tilde{u}_t(b) \} \quad (10.4.5)$$


---

**Corollary 10.1.** *The WIN-EXP algorithm with the latter unbiased utility estimates and step size  $\sqrt{\frac{\log(|B|)}{2T|O|}}$ , achieves regret in the outcome-based feedback with batch rewards setting at most:  $2\sqrt{2T|O|\log(|B|)}$ .*

**Lemma 10.3.** *At each iteration  $t$ , for any action  $b \in B$ , the random variable  $\tilde{u}_t(b)$  is an unbiased estimate of  $u_t(b) - 1$  and can actually be constructed based on the feedback that the learner receives:  $\forall b \in B : \tilde{u}_t(b) = \sum_{o \in O} \frac{\Pr_t[o|b]}{\Pr_t[o]} f_t(o) (Q_t(b, o) - 1)$  and has expected second moment bounded by:  $\forall b \in B : \mathbb{E} [(\tilde{u}_t(b))^2] \leq 4 \sum_{o \in O} \frac{\Pr_t[o|b]}{\Pr_t[o]}$ .*

*Proof.* For the estimate of the utility it holds that:

$$\begin{aligned} \tilde{u}_t(b) &= \frac{1}{|I_t|} \sum_{\tau \in I_t} \frac{(r_\tau(b, o_\tau) - 1)\Pr_t[o_\tau|b]}{\Pr_t[o_\tau]} \\ &= \frac{1}{|I_t|} \sum_{o \in O} \sum_{\tau \in I_{to}} \frac{(r_\tau(b, o) - 1)\Pr_t[o|b]}{\Pr_t[o]} \\ &= \sum_{o \in O: |I_{to}| > 0} \frac{\Pr_t[o|b]}{\Pr_t[o]} f_t(o) \frac{1}{|I_{to}|} \sum_{\tau \in I_{to}} (r_\tau(b, o) - 1) \\ &= \sum_{o \in O} \frac{\Pr_t[o|b]}{\Pr_t[o]} f_t(o) (Q_t(b, o) - 1) \end{aligned} \quad (10.4.6)$$

From the first equation it follows along identical lines, that this is an unbiased estimate, while from the last equation it is easy to see that this unbiased estimate can be constructed based on the feedback that the learner receives.

Moreover, we can also bound the second moment of these estimates by a similar quantity as in the previous section:

$$\begin{aligned}
\mathbb{E}[\tilde{u}_t(b)^2] &= \sum_{b_t \in B} \mathbb{E} \left[ \left( \sum_{o \in O} \frac{\Pr_t[o|b]}{\Pr_t[o]} f_t(o) (Q_t(b, o) - 1) \right)^2 \middle| b_t \right] \pi_t(b_t) \\
&\leq \sum_{b_t \in B} \mathbb{E} \left[ \sum_{o \in O} \left( \frac{\Pr_t[o|b]}{\Pr_t[o]} (Q_t(b, o) - 1) \right)^2 f_t(o) \middle| b_t \right] \pi_t(b_t) \quad (\text{By Jensen's inequality}) \\
&= \sum_{b_t \in B} \sum_{o \in O} \left( \frac{\Pr_t[o|b]}{\Pr_t[o]} (Q_t(b, o) - 1) \right)^2 \mathbb{E}[f_t(o)|b_t] \cdot \pi_t(b_t) \\
&= \sum_{o \in O} \left( \frac{\Pr_t[o|b]}{\Pr_t[o]} (Q_t(b, o) - 1) \right)^2 \sum_{b_t \in B} \mathbb{E}[f_t(o)|b_t] \cdot \pi_t(b_t) \\
&= \sum_{o \in O} \left( \frac{\Pr_t[o|b]}{\Pr_t[o]} (Q_t(b, o) - 1) \right)^2 \Pr_t[o] \leq 4 \sum_{o \in O} \frac{\Pr_t[o|b]}{\Pr_t[o]}
\end{aligned}$$

■

Then following the same techniques in Theorem 10.2, it is straightforward to conclude the proof of the corollary.

It is also interesting to note that the same result holds if instead of using  $f_t(o)$  in the expected utility (Equation (10.4.6)), we used its *mean value*, which is  $x_t(o, b_t) = \Pr_t[o|b_t]$ . This would not change any of the derivations above. The nice property of this alternative is that the learner does not need to learn the realized fraction of each outcome, but only the expected fraction of each outcome. This is already contained in the function  $x_t(\cdot, \cdot)$ , which we assumed was given to the learner at the end of each iteration. Thus, with these new estimates, the learner does not need to observe  $f_t(o)$ . In Appendix F.1 we also discuss the case where different periods can have different number of rewards and how to extend our estimate to that case. The batch rewards setting finds an interesting application in the case of learning in sponsored search, as we describe below.

**Example 10.4** (Sponsored Search). *In the case of sponsored search auctions, the latter boils down to learning the average value  $\hat{v} = \frac{1}{\#clicks} \sum_{clicks} v_{click}$  for the clicks that were generated, as well as the cost-*

per-click function  $p_t(b)$ , which is assumed to be constant throughout the period  $t$ . Given these quantities, the learner can compute:  $Q(b, A) = \hat{v} - p_t(b)$  and  $Q(b, \neg A) = 0$ . An advertiser can keep track of the traffic generated by a search engine ad and hence, can keep track of the number of clicks from the search engine and the value generated by each of these clicks (conversion). Thus, he can estimate  $\hat{v}$ . Moreover, he can elicit the probability of click (aka click-through-rate or CTR) curves  $x_t(\cdot)$  and the cost-per-click (CPC) curves  $p_t(\cdot)$  over relatively small periods of time of about a few days. See for instance the Adwords bid simulator tools offered by Google, which exactly enable a bidder to elicit these curves [Google \(2018a,b,c\)](#), [Microsoft \(2018\)](#)<sup>‡</sup>.

Thus, with these at hand we can apply our batch reward outcome based feedback algorithm and get regret that does not grow linearly with  $|B|$ , but only as  $4\sqrt{T \log(|B|)}$ . Our main assumption is that the expected CTR and CPC curves during this relatively small period of a few days remains approximately constant. The latter holds if the distribution of click-through-rates does not change within these days and if the bids of opponent bidders also do not significantly change. This is a reasonable assumption when feedback can be elicited relatively frequently, which is the case in practice.

## 10.5 CONTINUOUS ACTIONS WITH PIECEWISE-LIPSCHITZ REWARDS

In this section, we extend our discussions to continuous action spaces; that is, we allow the action of each bidder to lie in a continuous action space  $B$  (e.g. a uniform interval in  $[0, 1]$ ). To assist us in our analysis, we are going to use the following discretization result by Kleinberg [Kleinberg \(2005\)](#)<sup>§</sup>. For what follows in this section, let  $R(T, B) = \sup_{b^* \in B} \mathbb{E} \left[ \sum_{t=1}^T (u_t(b^*) - u_t(b_t)) \right]$  be the regret of the bidder, after  $T$  rounds with respect to an action space  $B$ . Moreover, for any pairs of action spaces  $B$  and  $B$  we let:  $DE(B, B) = \sup_{b \in B} \sum_{t=1}^T u_t(b) - \sup_{b' \in B} \sum_{t=1}^T u_t(b')$ , denote the discretization error incurred by optimizing over  $B$  instead of  $B$ .

**Lemma 10.4.** ([Kleinberg \(2005\)](#), [Kleinberg et al. \(2008a\)](#)) Let  $B$  be a continuous action space and  $B$  a discretization of  $B$ . Then:

$$R(T, B) \leq R(T, B) + DE(B, B)$$

---

<sup>‡</sup>One could argue that the CTRs that the bidder gets in this case are not accurate enough. Nevertheless, even if they have random perturbations, we show in our experimental results that for reasonable noise assumptions, WIN-EXP is more robust compared to EXP3.

<sup>§</sup>In [Kleinberg \(2005\)](#) Kleinberg discusses the uniform discretization of continuum-armed bandits and in [Kleinberg et al. \(2008a\)](#) the authors extend the results for the case of Lipschitz-armed bandits.

Observe now that in the setting of Weed et al. (2016) the discretization error was:  $DE(B, \mathcal{B}) = 0$  if  $\varepsilon < \Delta^o$ , as we discussed in Section 10.4 and that was *the key* that allowed us to recover this result, without adding an extra  $\varepsilon T$  in the regret of the bidder. To achieve that, we construct the following general class of utility functions:

**Definition 10.1** ( $\Delta^o$ -Piecewise Lipschitz Average Utilities). *A learning setting with action space  $B = [0, 1]^d$ , is said to have  $\Delta^o$ -Piecewise Lipschitz Cumulative Utilities if the average utility function  $\frac{1}{T} \sum_{t=1}^T u_t(b)$  satisfies the following conditions: the bidding space  $[0, 1]^d$  is divided into  $d$ -dimensional cubes with edge length at least  $\Delta^o$  and within each cube the utility is  $L$ -Lipschitz with respect to the  $\ell_\infty$  norm. Moreover, for any boundary point there exists a sequence of non-boundary points whose limit cumulative utility is at least as large as the cumulative utility of the boundary point.*

**Lemma 10.5** (Discretization Error for Piecewise Lipschitz). *Let  $\mathcal{B} = [0, 1]^d$  be a continuous action space and  $B$  a uniform  $\varepsilon$ -grid of  $[0, 1]^d$ , such that  $\varepsilon < \Delta^o$  (i.e.  $B$  consists of all the points whose coordinates are multiples of a given  $\varepsilon$ ). Assume that the average utility function is  $\Delta^o$ -Piecewise  $L$ -Lipschitz. Then, the discretization error of  $B$  is bounded as:  $DE(B, \mathcal{B}) \leq \varepsilon LT$ .*

*Proof.* Let  $\text{OPT} = \arg \sup_{b \in \mathcal{B}} \sum_{t=1}^T u_t(b)$  be the best fixed action in the continuous action space  $B$  in hindsight. Since  $\varepsilon < \Delta^o$ , then  $b^*$  must belong to some  $d$ -dimensional  $\varepsilon$ -cube, either as an interior point or as a limit of interior points, as expressed by Definition 10.1. The utility is  $L$ -Lipschitz within this  $\varepsilon$ -cube and since  $\varepsilon < \Delta^o$ , each cube contains at least one point in the discretized space  $B$ . For the case where  $\text{OPT}$  is achieved as the limit of interior points then for every  $\delta > 0$  there exist an interior point of some cube  $\tilde{b}$ , such that  $\sum_{t=1}^T u_t(\tilde{b}) \geq \text{OPT} - \delta$ . The same obviously holds when  $\text{OPT}$  is achieved by an interior point. Let  $\hat{b}$  be the closest discretized point to  $\tilde{b}$  that lies in the same cube as  $\tilde{b}$ . Since  $\|\hat{b} - \tilde{b}\|_\infty \leq \varepsilon$ , by the Lipschitzness of the average reward function within each cube, we get:

$$\text{OPT} \leq \sum_{t=1}^T u_t(\tilde{b}) + \delta \leq \sum_{t=1}^T u_t(\hat{b}) + \delta + \varepsilon LT \leq \sup_{b \in B} \sum_{t=1}^T u_t(\hat{b}) + \delta + \varepsilon LT$$

Since we can take  $\delta$  as close to zero as we want, we get the lemma. ■

If we know the Lipschitzness constant  $L$  mentioned above, the time horizon  $T$  and  $\Delta^o$ , then our

WIN-EXP algorithm for Outcome-Based Feedback with Batch Rewards yields regret as defined by the following theorem. Subsequently, we show how to deal with unknown parameters  $L$ ,  $T$  and  $\Delta^o$  by applying a standard doubling trick.

### Theorem 10.3

Let  $B = [0, 1]^d$  be the action space as defined in our model and let  $B$  be a uniform  $\varepsilon$ -grid of  $B$ .

The WIN-EXP algorithm with unbiased estimates given by Equation 10.4.6 on space  $B$  with step size  $\sqrt{\frac{\log(|B|)}{2T|O|}}$  and  $\varepsilon = \min\left\{\frac{1}{LT}, \Delta^o\right\}$  achieves expected regret at most

$$2\sqrt{2T|O|d\log\left(\max\left\{\frac{1}{\Delta^o}, LT\right\}\right)} + 1$$

in the outcome-based feedback with batch rewards and  $\Delta^o$ -Piecewise  $L$ -Lipschitz average utilities <sup>a</sup>.

<sup>a</sup>Interestingly, the above regret bound can help to retrieve two familiar expressions for the regret. First, when  $L = 0$  (i.e. when the function is constant within each cube), which is the case for the second price auction analyzed by Weed et al. (2016),  $R(T) = 2\sqrt{2dT|O|\log\left(\frac{1}{\Delta^o}\right)} + 1$ . Hence, we recover the bounds from the prior sections up to a tiny increase. Second, when  $\Delta^o \rightarrow \infty$ , then we have functions that are  $L$ -Lipschitz in the whole space  $B$  and the regret bound that we retrieve is:  $R(T) = 2\sqrt{2dT|O|\log(LT)} + 1$ , which is of the type achieved in continuous lipschitz bandit settings.

*Proof.* From Lemma 10.5 we know that for  $\varepsilon < \Delta^o$ , the discretization error is  $DE(B, B) \leq \varepsilon LT$ . Combining Lemma 10.4 and Corollary 10.1, we have

$$\begin{aligned} R(T, B) &\leq R(T, B) + DE(B, B) = 2\sqrt{2T|O|\log(|B|)} + \varepsilon LT \\ &= 2\sqrt{2T|O|\log\left(\frac{1}{\varepsilon^d}\right)} + \varepsilon LT \\ &= 2\sqrt{2dT|O|\log\left(\frac{1}{\varepsilon}\right)} + \varepsilon LT \\ &= 2\sqrt{2dT|O|\log\left(\max\left\{LT, \frac{1}{\Delta^o}\right\}\right)} + \min\left\{\frac{1}{LT}, \Delta^o\right\} \\ &\leq 2\sqrt{2dT|O|\log\left(\max\left\{LT, \frac{1}{\Delta^o}\right\}\right)} + 1 \end{aligned}$$

■

**Example 10.5** (First Price and All-Pay Auctions). Consider the case of learning in first price or all-pay auctions. In the former, the highest bidder wins and pays his bid, while in the latter the highest bidder wins and every player pays his bid whether he wins or loses. Let  $B_t$  be the highest other bid at time  $t$ . Then the average hindsight utility of the player in each auction is <sup>T</sup>:

$$\frac{1}{T} \sum_{t=1}^T u_t(b) = \frac{1}{T} \sum_{t=1}^T v_t \cdot \mathbb{1}\{b > B_t\} - b \cdot \frac{1}{T} \sum_{t=1}^T \mathbb{1}\{b > B_t\} \quad (\text{first price})$$

$$\frac{1}{T} \sum_{t=1}^T u_t(b) = \frac{1}{T} \sum_{t=1}^T v_t \cdot \mathbb{1}\{b > B_t\} - b \quad (\text{all-pay})$$

Let  $\Delta^o$  be the smallest difference between the highest other bid at any two iterations  $t$  and  $t'$  <sup>H</sup>. Then observe that the average utilities in this setting are  $\Delta^o$ -Piecewise 1-Lipschitz: Between any two highest other bids, the average allocation,  $\frac{1}{T} \sum_{t=1}^T v_t \cdot \mathbb{1}\{b > B_t\}$ , of the player remains constant and the only thing that changes is his payment which grows linearly. Hence, the derivative at any bid between any two such highest other bids is upper bounded by 1. Hence, by applying Theorem 10.3, our WIN-EXP algorithm with a uniform discretization on a  $\varepsilon$ -grid, for  $\varepsilon = \min\{\Delta^o, \frac{1}{T}\}$ , achieves regret  $4\sqrt{T \log(\max\{\frac{1}{\Delta^o}, T\})} + 1$ , where we used that  $|O| = 2$  and  $d = 1$  for any of these auctions.

**UNKNOWN LIPSCHITZNESS CONSTANT.** In Theorem 10.3 the discretization parameter  $\varepsilon$  depends on the prior knowledge of the Lipschitzness constant,  $L$ , the number of rounds,  $T$  and the minimum edge length of each  $d$ -dimensional cube,  $\Delta^o$ . In order to address the problem that in general we do not know any of those constants a priori, we will apply a standard doubling trick ([Auer et al. \(2002b\)](#)) to remove this dependence. We assume that  $T$  is upper bounded by a constant  $T_M$  and similarly we also assume that  $\log(\max\{LT, \frac{1}{\Delta^o}\})$  is upper bounded by a constant.

We will then initialize two bounds:  $B_T = 1$  and  $B_{\Delta^o, LT} = 1$  and run the WIN-EXP algorithm with step size  $\sqrt{\frac{\log(1/\varepsilon)}{2B_T|O|}}$  and  $\varepsilon = \min\{\frac{1}{LT}, \Delta^o\}$  until  $t \leq B_T$  or  $\log(\max\{tL, \frac{1}{\Delta^o}\}) \leq B_{\Delta^o, LT}$  fails to hold. If one of these discriminants fails, then we double the bound and restart the algorithm. This modified strategy only increases the regret by a constant factor.

**Corollary 10.2.** The WIN-EXP algorithm run with the above doubling trick achieves an expected regret

---

<sup>T</sup>For simplicity assume the player loses in case of ties, though we can handle arbitrary random tie-breaking rules.

<sup>H</sup>This is an analogue of the  $\Delta^o$  used by [Weed et al. \(2016\)](#) in second price auctions.

$$\text{bound } \mathcal{R}(T) \leq 25\sqrt{2dT|O|\log(\max\{LT, \frac{1}{\Delta^o}\})} + 1$$

*Proof of Corollary 10.2.* Based on the doubling trick that we described above, we divide the algorithm into stages in which  $B_T$  and  $B_{\Delta^o, LT}$  are constants. Let  $B_L^*$  and  $B_{\Delta^o, LT}^*$  be the values of  $B_L$  and  $B_{\Delta, LT}$  respectively when the algorithm terminates. There is at most a total of  $\log(B_T^*) + \log(B_{\Delta^o, LT}^*) + 1$  stages in this doubling process. Since the actual expected regret is bounded by the sum of the regret of each stage, following the result of Theorem 10.3, we have

$$\begin{aligned} R(T) &\leq \sum_{i=0}^{\lceil \log(B_T^*) \rceil} \sum_{j=0}^{\lceil \log(B_{\Delta^o, LT}^*) \rceil} \left( 2\sqrt{2d2^i|O|2^j} \right) + \log(B_T^*) + \log(B_{\Delta^o, LT}^*) + 1 \\ &= \sum_{i=0}^{\lceil \log(B_T^*) \rceil} \sum_{j=0}^{\lceil \log(B_{\Delta^o, LT}^*) \rceil} \left( 2\sqrt{2d|O|2^i \cdot 2^j} \right) + \log(B_T^*B_{\Delta^o, LT}^*) + 1 \\ &= \left[ \sum_{i=0}^{\lceil \log(B_T^*) \rceil} (\sqrt{2})^i \right] \cdot \left[ \sum_{j=0}^{\lceil \log(B_{\Delta^o, LT}^*) \rceil} (\sqrt{2})^j \right] 2\sqrt{2d|O|} + \log(B_T^*B_{LT, \Delta^o}^*) + 1 \\ &= \frac{1 - \sqrt{2}^{\lceil \log(B_T^*) \rceil + 1}}{1 - \sqrt{2}} \cdot \frac{1 - \sqrt{2}^{\lceil \log(B_{\Delta^o, LT}^*) \rceil + 1}}{1 - \sqrt{2}} \cdot 2\sqrt{2d|O|} + \log(B_T^*B_{\Delta^o, LT}^*) + 1 \\ &\leq \left( \frac{\sqrt{2}}{\sqrt{2} - 1} \right)^2 \sqrt{B_T^*B_{\Delta^o, LT}^*} \cdot 2\sqrt{2d|O|} + \log(B_T^*B_{\Delta^o, LT}^*) + 1 \\ &= \left( \frac{\sqrt{2}}{\sqrt{2} - 1} \right)^2 \cdot 2\sqrt{2d|O|B_T^*B_{\Delta^o, LT}^*} + \log(B_T^*B_{\Delta^o, LT}^*) + 1 \\ &\leq 25\sqrt{2d|O|B_T^*B_{\Delta^o, LT}^*} + 1 \end{aligned}$$

Combining the fact that  $B_T^* \leq T$  and  $B_{\Delta^o, LT}^* \leq \log(\max\{LT, \frac{1}{\Delta^o}\})$  as well as the above inequalities, we complete the proof. ■

### 10.5.1 SPONSORED SEARCH WITH LIPSCHITZ UTILITIES

In this subsection, we extend our analysis of learning in the sponsored search auction model (Example 10.4) to the continuous bid space case, i.e., each bidder can submit a bid  $b \in [0, 1]$ . As a reminder, the utility function is:  $u_t(b) = x_t(b)(\hat{v}_t - p_t(b))$ , where  $b \in [0, 1]$ ,  $\hat{v}_t \in [0, 1]$  is the average value for the clicks at iteration  $t$ ,  $x_t(\cdot)$  is the CTR curve and  $p_t(\cdot)$  is the CPC curve. These curves

are implicitly formed by running some form of a Generalized Second Price auction (GSP) at each iteration to determine the allocation and payment rules.

We show in this section that the form of the GSP ran in reality gives rise to Lipschitz utilities, under some minimal assumptions. Therefore, we can apply the results in Section 10.5 to get regret bounds even with respect to the continuous bid space  $B = [0, 1]$  \*\*. We begin by providing a brief description of the type of Generalized Second Price auction ran in practice.

**Definition 10.2** (Weighted-GSP). *Each bidder  $i$  is assigned a quality score  $s_i \in [0, 1]$ . Bidders are ranked according to their score-weighted bid  $s_i \cdot b_i$ , typically called the rank-score. Every bidder whose rank-score does not pass a reserve  $r$  is discarded. Bidders are allocated slots in decreasing order of rank-score. Each bidder is charged per-click the lowest bid he could have submitted and maintained the same slot. Hence, if a bidder  $i$  is allocated a slot  $k$  and  $\rho_{k+1}$  is the rank-score of the bidder in slot  $k + 1$ , then he is charged  $\rho_{k+1}/s_i$  per-click. We denote with  $U_i(\mathbf{b}, \mathbf{s}, r)$ , the utility of bidder  $i$  under a bid profile  $\mathbf{b}$  and score profile  $\mathbf{s}$ .*

The quality scores are typically highly random and dependent on the features of the advertisement and the user that is currently viewing the page. Hence, a reasonable modeling assumption is that the scores  $s_i$  at each auction are drawn i.i.d. from some distribution with CDF  $F_i$ . We now show that if the CDF  $F_i$  is Lipschitz (i.e. admits a bounded density), then the utilities of the bidders are also Lipschitz.

#### Theorem 10.4: Lipschitzness of the utility of Weighted GSP

Suppose that the score  $s_i$  of each bidder  $i$  in a weighted GSP is drawn independently from a distribution with an  $L$ -Lipschitz CDF  $F_i$ . Then, the expected utility

$$u_i(b_i, \mathbf{b}_{-i}, r) = \mathbb{E}_{\mathbf{s}} [U_i(b_i, \mathbf{b}_{-i}, \mathbf{s}, r)]$$

is  $\frac{2nL}{r}$ -Lipschitz wrt  $b_i$ .

*Proof.* Consider a player  $i$ . Observe that conditional on the player's score  $s_i$ , his utility remains constant if he is allocated the same slot. Moreover, when the slots are different, then the difference

---

\*\*The aforementioned Lipschitzness is also reinforced by real world data sets from Microsoft's sponsored search auction system.

in utilities is at most 2, since utilities lie in  $[-1, 1]$ . Moreover, because the slots are allocated in decreasing order of rank scores, the slot allocation of a player is different under  $b_i$  and  $b'_i$  only if there exists a player  $j$ , who passes the rank-score reserve (i.e.  $s_j \cdot b_j \geq r$ ) and whose rank-score  $s_j \cdot b_j$  lies in the interval  $[s_i \cdot b_i, s_i \cdot (b_i + \varepsilon)]$ . Hence, conditional on  $s_i$ , the absolute difference between the player's expected utility when he bids  $b_i$  and when he bids  $b_i + \varepsilon$ , with  $\varepsilon > 0$ , is upper bounded by:

$$2 \cdot \Pr [\exists j \neq i \text{ s.t. } s_j \cdot b_j \in [s_i \cdot b_i, s_i \cdot (b_i + \varepsilon)] \text{ and } s_j \cdot b_j \geq r \mid s_i]$$

By a union bound the latter is at most:

$$2 \cdot \sum_{j \neq i} \Pr \left[ s_j \in \left[ \frac{s_i b_i}{b_j}, \frac{s_i (b_i + \varepsilon)}{b_j} \right] \text{ and } s_j \cdot b_j \geq r \mid s_i \right]$$

Since  $s_j \in [0, 1]$ , the previous quantity is upper bounded by replacing the event  $s_j \cdot b_j \geq r$  by  $b_j \geq r$ . This event is independent of the scores and we can then write the above bound as:

$$2 \cdot \sum_{j \neq i \text{ s.t. } b_j \geq r} \Pr \left[ s_j \in \left[ \frac{s_i b_i}{b_j}, \frac{s_i (b_i + \varepsilon)}{b_j} \right] \mid s_i \right]$$

Since each quality score  $s_j$  is drawn independently from an  $L$ -Lipschitz CDF  $F_j$ , we can further simplify the bound by:

$$2 \cdot \sum_{j \neq i \text{ s.t. } b_j \geq r} \left[ F_j \left( \frac{s_i (b_i + \varepsilon)}{b_j} \right) - F_j \left( \frac{s_i b_i}{b_j} \right) \right] \leq 2 \cdot \sum_{j \neq i \text{ s.t. } b_j \geq r} L \frac{s_i \varepsilon}{b_j} \leq 2 \cdot \sum_{j \neq i \text{ s.t. } b_j \geq r} L \frac{s_i \varepsilon}{r} \leq \frac{2nL}{r} \varepsilon$$

Since the absolute difference of utilities between these two bids is upper bounded conditional on  $s_i$ , by the triangle inequality it is also upper bounded even unconditional on  $s_i$ , which leads to the Lipschitz property we want:

$$|u_i(b_i, \mathbf{b}_{-i}, r) - u_i(b_i + \varepsilon, \mathbf{b}_{-i}, r)| \leq \frac{2nL}{r} \varepsilon \quad (10.5.1)$$

■

Thus, we see that when the quality scores in sponsored search are drawn from  $L$ -Lipschitz CDFs

$F_i, \forall i \in n$  and the reserve is lower bounded by  $\delta > 0$ , then the utilities are  $\frac{2nL}{\delta}$ -Lipschitz and we can achieve good regret bounds by using the WIN-EXP algorithm with batch rewards, with action space  $B$  being a uniform  $\varepsilon$ -grid,  $\varepsilon = \frac{\delta}{2nLT}$  and unbiased estimates given by Equation (10.4.6) or Equation (10.4.3). In the case of sponsored search the second unbiased estimate takes the following simple form:

$$\tilde{u}_t(b) = \frac{x_t(b) \cdot x_t(b_t)}{\sum_{b' \in B} \pi_t(b') x_t(b')} (\hat{v}_t - p_t(b) - 1) - \frac{(1-x_t(b)) \cdot (1-x_t(b_t))}{\sum_{b' \in B} \pi_t(b') (1-x_t(b'))} \quad (10.5.2)$$

where  $\hat{v}_t$  is the average value from the clicks that happened during iteration  $t$ ,  $x_t(\cdot)$  is the CTR curve,  $b_t$  is the realized bid that the bidder submitted and  $\pi_t(\cdot)$  is the distribution over discretized bids of the algorithm at that iteration. We can then apply Theorem 10.3 to get the following guarantee:

**Corollary 10.3.** *The WIN-EXP algorithm run on a uniform  $\varepsilon$ -grid with  $\varepsilon = \frac{\delta}{2nLT}$ , step size  $\sqrt{\frac{\log(1/\varepsilon)}{4T}}$  and unbiased estimates given by Equation (10.4.6) or Equation (10.4.3), when applied to the sponsored search auction setting with quality scores drawn independently from distributions with  $L$ -Lipschitz CDFs, achieves regret at most:  $4\sqrt{T \log(\frac{2nLT}{\delta})} + 1$ .*

## 10.6 FURTHER EXTENSIONS

In this section, we discuss an extension to switching regret and the implications on Price of Anarchy and one to the feedback graphs setting.

### 10.6.1 SWITCHING REGRET AND IMPLICATIONS FOR PRICE OF ANARCHY

We show below that actually our results can be extended to capture the case where, instead of having just one optimal bid  $b^*$ , there is a sequence of  $C \geq 1$  switches in the optimal bids. Using the results presented in Gyorgy et al. (2012) and adapting them for our setting we get the following corollary.

**Corollary 10.4.** *Let  $C \geq 0$  be the number of times that the optimal bid  $b^* \in B$  switches in a horizon of  $T$  rounds. Then, using Algorithm 2 in Gyorgy et al. (2012) with  $\mathcal{A} \equiv$  WIN-EXP and any  $\alpha \in (0, 1)$  we can achieve expected switching regret at most:  $O\left(\sqrt{(C+1)^2 \left(2 + \frac{1}{\alpha}\right) 2d|O|T \log(\max\{LT, \frac{1}{\Delta^\alpha}\})}\right)$*

*Proof.* We first observe that the results proven in [Gyorgy et al. \(2012\)](#) for a prediction algorithm  $\mathcal{A}$  with *regret* upper bounded by  $\rho(T)$  hold also for algorithms  $\mathcal{A}$  for which we know upper bound of their expected regrets. Specifically, if algorithm  $\mathcal{A}$  has an upper bound of  $\rho(T)$  for its expected regret, where  $\rho(T)$  is a concave, non-decreasing,  $[0, +\infty) \rightarrow [0, +\infty)$  function, then Lemma 1 from [Gyorgy et al. \(2012\)](#) holds for *expected* regret. With that in mind, we can directly apply the *Randomized Tracking Algorithm* and get expected switching regret upper bounded by:

$$(C(TP) + 1) L_{C(TP),T} \rho \left( \frac{T}{(C(TP) + 1) L_{C(TP),T}} \right) + \sum_{t=1}^T \frac{\eta_t}{8} + \frac{r_T ((C(TP) + 1) L_{C(TP),T-1} - 1)}{\eta_T} \quad (10.6.1)$$

where  $TP$  is the switching path of the optimal bids and  $C(TP)$  is the number of switches in the optimal bid according to this path.

We proceed by making sure that the conditions for the upper bound of the expected regret of WIN-EXP satisfy the conditions required by algorithm  $\mathcal{A}$  in [Gyorgy et al. \(2012\)](#). Indeed, the upper bound of the expected regret of our algorithm,  $\sqrt{2dT|O| \log (\max \{LT, \frac{1}{\Delta_o}\})} + 1$ , is non decreasing in  $T$ . Also, at timestep  $t = 0$ , we incur no regret. We also apply the following slight modifications in Algorithm 2 in [Gyorgy et al. \(2012\)](#) so as to match the nature of our problem. First, instead of computing the expected loss at each timestep  $t$ , we will now compute the expected outcome-based utility, i.e.  $\bar{u}_t(\pi_t) = \sum_{b \in B} \pi_t(b) \mathbb{E}_{o_t} [\tilde{u}_t(b)]$ . Second, instead of the cumulative loss of their algorithm  $\mathcal{A}$  we will now use the cumulative outcome-based expected utility of WIN-EXP, i.e.  $\bar{U}_t(\text{WIN-EXP}, T) = \sum_{c=0}^C \bar{U}_{\text{WIN-EXP}}(t_c, t_{c+1})$ , where

$$\bar{U}_{\text{WIN-EXP}}(t_c, t_{c+1}) = \sum_{s=t_c}^{t_{c+1}-1} \bar{u}_s(\pi_{\text{WIN-EXP},s}(t_c))$$

is the cumulative outcome-based expected utility gained from our WIN-EXP algorithm in the time interval  $[t_c, t_{c+1}]^{\dagger\dagger}$  with respect to  $\bar{u}_s$  for  $s \in [t_c, t_{c+1}]$ . Now, we are computing the regret components of [Gyorgy et al. \(2012\)](#) so as to achieve the desired result.

Before we show the specifics of the computation, we note here that  $g > 0$  is a *parameter* of the Tracking Regret algorithm presented by [Gyorgy et al. \(2012\)](#) and can be set a priori from the de-

---

<sup>††</sup>We clarify here that these time intervals are with respect to the switching bids.

signer of the algorithm. The complexity of  $g$  affects the computational complexity of the algorithm and there is a tradeoff between the computational complexity and the regret of the algorithm. For our computations here, we will set

$$g + 1 = \left( \frac{T}{C(TP) + 1} \right)^\alpha \quad (10.6.2)$$

where  $0 < \alpha < 1$  is a constant. Now, we are ready to compute the components of the regret:

$$\begin{aligned} A &= L_{C(TP),T} (C(TP) + 1) R_{\text{WIN-EXP}} \left( \frac{T}{L_{C(TP),T} (C(TP) + 1)} \right) \\ &\leq 25 \left( \frac{\log \left( \frac{T}{C(TP)+1} \right)}{\log(g+1)} + 2 \right) (C(TP) + 1) \left( \sqrt{2d|O| \frac{T \log(g+1) \log(m)}{\log \left( \frac{T}{C(TP)+1} \right) + 2 \log(g+1)}} + 1 \right) \\ &= 50 \cdot \left( 2 + \frac{1}{\alpha} \right) \cdot (C(TP) + 1) \sqrt{2d|O| \cdot \frac{\alpha}{1+2\alpha} \cdot T \log(m)} \\ &\leq 50 \sqrt{\frac{1+2\alpha}{\alpha} \cdot (C(TP) + 1)^2 2d|O| T \log(m)} \\ &\leq 50 \sqrt{\left( 2 + \frac{1}{\alpha} \right) \cdot (C + 1)^2 2d|O| T \log(m)} \end{aligned}$$

where in the second equality we have denoted  $\log(m) = \log(\max\{LT, \frac{1}{\Delta^\alpha}\})$  and the last inequality comes from the fact that  $C$  is the upper bound on the number of switches that the transition path  $TP$  can have. Moving on to the computation of the rest of the components of the regret:

$$\begin{aligned} B &= \sum_{t=1}^T \frac{\eta_t}{8} \leq \frac{1}{8} \sqrt{\frac{T \log(1/\varepsilon)}{2|O|}} = O\left(\sqrt{\frac{T}{|O|}}\right) \\ D &= r_T (L_{C(TP),T} (C(TP) + 1) - 1) \\ &= \left( \frac{\alpha+1}{\alpha} + \varepsilon_2 \right) \log T + \log(1 + \varepsilon_2) - \left( \frac{\alpha+1}{\alpha} \right) \log \varepsilon_2 \end{aligned}$$

where  $\varepsilon_2 \in (0, 1)$  is a constant. Before we conclude, we observe that even though Corollary 1 of Gyorgy et al. (2012) is stated as a high-probability ex post result, the proof uses a result from Cesa-Bianchi and Lugosi (2006) (Lemma 4.1) which also holds for the expected regret. According to Gyorgy et al. (2012) the switching regret is the sum of the aforementioned  $A, B, D$ . Thus, we get the result. ■

This result has implications on the price of anarchy (PoA) of auctions. In the case of sponsored search where bidders' valuations are changing over time adversarially but non-adaptively, our result shows that if the valuation does not change more than  $C$  times, we can compete with any bid that is a function of the value of the bidder at each iteration, with regret rate given by the latter theorem. Therefore, by standard PoA arguments [Lykouris et al. \(2016\)](#), this would imply convergence to an approximately efficient outcome at a faster rate than bandit regret rates.

### 10.6.2 FEEDBACK GRAPHS OVER OUTCOMES

We now extend Section 10.5, by assuming that there is a directed feedback graph  $G = (O, E)$  over the outcomes. When outcome  $o_t$  is chosen, the player observes not only the outcome specific reward function  $r_t(\cdot, o_t)$ , for that outcome, but also for any outcome  $o$  in the out-neighborhood of  $o_t$  in the feedback graph, which we denote with  $N^{out}(o_t)$ . Correspondingly, we denote with  $N^{in}(o)$  the incoming neighborhood of  $o$  in  $G$ . Both neighborhoods include self-loops. Let  $G_\varepsilon = (O_\varepsilon, E_\varepsilon)$  be the sub-graph of  $G$  that contains only outcomes for which  $\Pr_t[o_t] \geq \varepsilon$  and subsequently, let  $N_\varepsilon^{in}$  and  $N_\varepsilon^{out}$  be the in and out neighborhoods of this sub-graph.

Based on  $G$ , we construct a WIN-EXP algorithm with step-size  $\eta = \sqrt{\frac{\log(|B|)}{8T\alpha \ln\left(\frac{16|O|^2T}{\alpha}\right)}}$ , utility estimate  $\tilde{u}_t(b) = \mathbb{1}\{o_t \in O_\varepsilon\} \sum_{o \in N_\varepsilon^{out}(o_t)} \frac{(r_t(b,o)-1)\Pr_t[o|b]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']}$  and feedback structure as described in the previous paragraph. With these changes we can show that the regret grows as a function of the *independence number of the feedback graph*, denoted with  $\alpha$ , rather than the *number of outcomes*.

#### Theorem 10.5: Regret of WIN-EXP-G

The regret of the WIN-EXP-G algorithm with step size  $\eta = \sqrt{\frac{\log(|B|)}{8T\alpha \ln\left(\frac{16|O|^2T}{\alpha}\right)}}$  is bounded by:  

$$R(T) \leq 2\sqrt{8\alpha T \log(|B|) \ln\left(\frac{16|O|^2T}{\alpha}\right)} + 1.$$

We first bound the moments of the estimates we build.

**Lemma 10.6.** *At each iteration  $t$ , for any action  $b \in B$ , the random variable  $\tilde{u}_t(b)$  has bias with respect to  $u_t(b) - 1$  bounded by:  $|\mathbb{E}[\tilde{u}_t(b)] - (u_t(b) - 1)| \leq 2\varepsilon|O|$  and has expected second moment bounded by:  

$$\forall b \in B : \mathbb{E}[\tilde{u}_t(b)^2] \leq 4 \sum_{o \in O_\varepsilon} \frac{\Pr_t[o|b]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']}$$*

---

**Algorithm 10.4:** WIN-EXP-G algorithm for learning with outcome-based feedback and a feedback graph over outcomes

---

1 Let  $\pi_1(b) = \frac{1}{|B|}$  for all  $b \in B$  (i.e. the uniform distribution over bids),  $\eta = \sqrt{\frac{\log(|B|)}{8T\alpha \ln\left(\frac{16|O|^2T}{\alpha}\right)}}$ .

2 **for each iteration  $t$  do**

3     Draw an action  $b_t \sim \pi_t(\cdot)$ , multinomial.

4     Observe  $x_t(\cdot)$ , chosen outcome  $o_t$  and associated reward function  $r_t(\cdot, o_t)$ .

5     Observe and associated reward function  $r_t(\cdot, \cdot)$  for all neighbor outcomes  $N_\varepsilon^{in}, N_\varepsilon^{out}$ .

6     Compute estimate of utility:

$$\tilde{u}_t(b) = \mathbb{1}\{o_t \in O_\varepsilon\} \sum_{o \in N_\varepsilon^{out}(o_t)} \frac{(r_t(b, o) - 1)\Pr_t[o|b]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} \quad (10.6.3)$$

Update  $\pi_t(\cdot)$  based on the Exponential Weights Update:

$$\forall b \in B : \pi_{t+1}(b) \propto \pi_t(b) \cdot \exp\{\eta \cdot \tilde{u}_t(b)\} \quad (10.6.4)$$


---

*Proof of Lemma 10.6.* For the expected utility we have:

$$\begin{aligned} \mathbb{E}[\tilde{u}_t(b)] &= \mathbb{E}_{o_t} \left[ \mathbb{1}\{o_t \in O_\varepsilon\} \sum_{o \in N_\varepsilon^{out}(o_t)} \frac{(r_t(b, o) - 1)\Pr_t[o|b]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} \right] \\ &= \sum_{o_t \in O_\varepsilon} \sum_{o \in N_\varepsilon^{out}(o_t)} \frac{(r_t(b, o) - 1)\Pr_t[o|b]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} \Pr_t[o_t] \\ &= \sum_{o \in O_\varepsilon} \sum_{o_t \in N_\varepsilon^{in}(o)} \frac{(r_t(b, o) - 1)\Pr_t[o|b]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} \Pr_t[o_t] \\ &= \sum_{o \in O_\varepsilon} \frac{(r_t(b, o) - 1)\Pr_t[o|b]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} \sum_{o_t \in N_\varepsilon^{in}(o)} \Pr_t[o_t] \\ &= \sum_{o \in O_\varepsilon} (r_t(b, o) - 1)\Pr_t[o|b] \\ &= \sum_{o \in O} (r_t(b, o) - 1)\Pr_t[o|b] - \sum_{o \notin O_\varepsilon} (r_t(b, o) - 1)\Pr_t[o|b] \\ &= u_t(b) - 1 - \sum_{o \notin O_\varepsilon} (r_t(b, o) - 1)\Pr_t[o|b] \end{aligned}$$

Thus, we get that the bias of  $\tilde{u}$  with respect to  $u_t - 1$  is bounded by:

$$|\mathbb{E}[\tilde{u}_t(b)] - (u_t(b) - 1)| \leq 2\varepsilon|O| \quad (10.6.5)$$

Similarly for the second moment:

$$\begin{aligned}\mathbb{E}[\tilde{u}_t(b)^2] &\leq \mathbb{E}_{o_t} \left[ \left( \mathbb{1}_{\{o_t \in O_\varepsilon\}} \sum_{o \in N_\varepsilon^{out}(o_t)} \frac{(r_t(b, o) - 1) \Pr_t[o|b]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} \right)^2 \right] \\ &= \sum_{o_t \in O_\varepsilon} \left( \sum_{o \in N_\varepsilon^{out}(o_t)} \frac{(r_t(b, o) - 1) \Pr_t[o|b]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} \right)^2 \Pr_t[o_t]\end{aligned}\quad (10.6.6)$$

Observe that the quantity inside the square:

$$\sum_{o \in N_\varepsilon^{out}(o_t)} \frac{(r_t(b, o) - 1)}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} \Pr_t[o|b]$$

can be thought of as an expected value of the quantity  $\frac{(r_t(b, o) - 1)}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']}$ , where  $o$  is the random variable and is drawn from the distribution of outcomes conditional on a bid  $b$ . Thus, by Jensen's inequality, the square of the latter expectation is at most the expectation of the square, i.e.:

$$\left( \sum_{o \in N_\varepsilon^{out}(o_t)} \frac{(r_t(b, o) - 1)}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} \Pr_t[o|b] \right)^2 \leq \sum_{o \in N_\varepsilon^{out}(o_t)} \frac{(r_t(b, o) - 1)^2}{\left( \sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o'] \right)^2} \Pr_t[o|b]$$

Combining with Equation (10.6.6), we get:

$$\begin{aligned}\mathbb{E}[\tilde{u}_t(b)^2] &\leq \sum_{o_t \in O_\varepsilon} \sum_{o \in N_\varepsilon^{out}(o_t)} \frac{(r_t(b, o) - 1)^2}{\left( \sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o'] \right)^2} \Pr_t[o|b] \Pr_t[o_t] \\ &= \sum_{o \in O_\varepsilon} \sum_{o_t \in N_\varepsilon^{in}(o)} \frac{(r_t(b, o) - 1)^2}{\left( \sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o'] \right)^2} \Pr_t[o|b] \Pr_t[o_t] \\ &= \sum_{o \in O_\varepsilon} \frac{(r_t(b, o) - 1)^2}{\left( \sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o'] \right)^2} \Pr_t[o|b] \sum_{o_t \in N_\varepsilon^{in}(o)} \Pr_t[o_t] \\ &= \sum_{o \in O_\varepsilon} \frac{(r_t(b, o) - 1)^2}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} \Pr_t[o|b] \\ &\leq 4 \sum_{o \in O_\varepsilon} \frac{\Pr_t[o|b]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']}\end{aligned}$$

where the last inequality holds since  $r_t(\cdot, \cdot) \in [-1, 1]$ . ■

*Proof of Theorem 10.5.* Observe that regret with respect to utilities  $u_t(\cdot)$  is equal to regret with re-

spect to the translated utilities  $u_t(\cdot) - 1$ . We use the fact that the exponential weight updates with an estimate  $\tilde{u}_t(\cdot) \leq 0$  which has bias with respect to the true utilities, bounded by  $\kappa$ , achieves expected regret of the form:

$$R(T) \leq \frac{\eta}{2} \sum_{t=1}^T \sum_{b \in B} \pi_t(b) \cdot \mathbb{E} [\tilde{u}_t(b)^2] + \frac{1}{\eta} \log(|B|) + 2\kappa T$$

For the detailed proof of the above claim, please see Appendix F.2. Invoking the bound on the bias and the second moment by Lemma 10.6, we get:

$$\begin{aligned} R(T) &\leq 2\eta \sum_{t=1}^T \sum_{b \in B} \pi_t(b) \cdot \sum_{o \in O_\varepsilon} \frac{\Pr_t[o|b]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} + \frac{1}{\eta} \log(|B|) + 4\varepsilon|O|T \\ &= 2\eta \sum_{t=1}^T \sum_{o \in O_\varepsilon} \sum_{b \in B} \pi_t(b) \cdot \frac{\Pr_t[o|b]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} + \frac{1}{\eta} \log(|B|) + 4\varepsilon|O|T \\ &= 2\eta \sum_{t=1}^T \sum_{o \in O_\varepsilon} \frac{\Pr_t[o]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} + \frac{1}{\eta} \log(|B|) + 4\varepsilon|O|T \end{aligned}$$

We can now invoke Lemma 5 of [Alon et al. \(2015\)](#), which states that:

**Lemma 10.7** ([Alon et al. \(2015\)](#)). *Let  $G = (V, E)$  be a directed graph with  $|V| = K$ , in which each node  $i \in V$  is assigned a positive weight  $w_i$ . Assume that  $\sum_{i \in V} w_i \leq 1$ , and that  $w_i \geq \varepsilon$  for all  $i \in V$  for some constant  $0 < \varepsilon < 1/2$ . Then*

$$\sum_{i \in V} \frac{w_i}{\sum_{j \in N^{in}(i)} w_j} \leq 4\alpha \ln \frac{4K}{\alpha\varepsilon} \quad (10.6.7)$$

where neighborhoods include self-loops and  $\alpha$  is the independence number of the graph.

Invoking the above lemma for the feedback graph  $G_\varepsilon$  (and noting that the independence number cannot increase by restricting to a sub-graph), we get:

$$\sum_{o \in O_\varepsilon} \frac{\Pr_t[o]}{\sum_{o' \in N_\varepsilon^{in}(o)} \Pr_t[o']} \leq 4\alpha \ln \frac{4|O|}{\alpha\varepsilon} \quad (10.6.8)$$

Thus, we get a bound on the regret of:

$$R(T) \leq 8\eta\alpha \ln \left( \frac{4|O|}{\alpha\varepsilon} \right) T + \frac{1}{\eta} \log(|B|) + 4\varepsilon|O|T$$

Picking  $\varepsilon = \frac{1}{4|O|T}$ , we get:

$$R(T) \leq 8\eta\alpha \ln \left( \frac{16|O|^2 T}{\alpha} \right) T + \frac{1}{\eta} \log(|B|) + 1$$

Picking  $\eta = \sqrt{\frac{\log(|B|)}{8T\alpha \ln \left( \frac{16|O|^2 T}{\alpha} \right)}}$ , we get the theorem. ■

In the case of learning in auctions, the feedback graph over outcomes can encode the possibility that winning an item can help you uncover your value for other items. For instance, in a combinatorial auction for  $m$  items, the reader should think of each node in the feedback graph as a bundle of items. Then the graph encodes the fact that winning bundle  $o$  can teach you the value for all bundles  $o' \in N^{out}(o)$ . If the feedback graph has small dependence number then a much better regret is achieved than the dependence on  $\sqrt{2^m}$ , that would have been derived by our outcome-based feedback results of prior sections, if we treated each bundle of items separately as an outcome.

## 10.7 EXPERIMENTAL RESULTS

In this section, we present our results from our comparative analysis between EXP3 and WIN-EXP on a simulated sponsored search system that we built and which is a close proxy of the actual sponsored search algorithms deployed in the industry. We implemented the weighted GSP auction as described in definition 10.2. The auctioneer draws i.i.d rank scores that are bidder and round specific; as is the case throughout our paper, here we have assumed a stochastic auctioneer with respect to the rank scores. After bidding, the bidder will always be able to observe the allocation function. Now, if the bidder gets allocated to a slot and she gets clicked, then, she is able observe the *value* and the payment curve. Values are assumed to lie in  $[0, 1]$  and they are obviously adversarial. Finally, the bidders choose bids from some  $\varepsilon$ -discretized grid of  $[0, 1]$  (in all experiments, apart from the ones comparing the regrets for different discretizations, we use  $\varepsilon = 0.01$ ) and update the probabilities of choosing each discrete bid according to EXP3 or WIN-EXP. Regret is measured with respect to the best fixed discretized bid in hindsight.

We distinguish three cases of the bidding behavior of the rest of the bidders (apart from our learner): i) all of them are *stochastic* adversaries drawing bids at random from some distribution,

ii) there is a subset of them that are bidding *adaptively*, by using an EXP3 online learning algorithm and iii) there is a subset of them that are bidding *adaptively* but using a WINEXP online learning algorithm (self play). Validating our theoretical claims, in all three cases, WIN-EXP outperforms EXP3 in terms of regret. We generate the event of whether a bidder gets clicked or not as follows: we draw a round specific threshold value in  $[0, 1]$  and the learner gets a click in case the CTR of the slot she got allocated (if any) is greater than this threshold value. Note here that the choice of a round specific threshold imposes *monotonicity*, i.e. if the learner did not get a click when allocated to a slot with CTR  $x_t(b)$ , she should not be able to get a click from slots with lower CTRs. We ran simulations with 3 different distributions of generating CTRs, so as to understand what is the effect of different levels of click-through-rates on the variance of our regret: i)  $x_t(b) \sim U[0.1, 1]$ , ii)  $x_t(b) \sim U[0.3, 1]$  and iii)  $x_t(b) \sim U[0.5, 1]$ . Finally, we address robustness of our results to errors in CTR estimation. For this, we add random noise to the CTRs of each slot and we report to the learners the allocation and payment functions that correspond to the erroneous CTRs. The noise was generated according to a normal distribution  $\mathcal{N}(0, \frac{1}{m})$ , where  $m$  could be viewed as the number of training samples on which a machine learning algorithm was ran in order to output the CTR estimate ( $m = 100, 1000, 10000$ ).

For each of the following simulations, there are  $N = 20$  bidders,  $k = 3$  slots and we ran the experiment for each round for a total of 30 times. For the simulations that correspond to adaptive adversaries we used  $a = 4$  adversaries. Our results for the *cumulative regret* are presented below. We measured ex-post regret with respect to the realized thresholds that determine whether a bidder gets clicked or not. Note that the solid plots correspond to the empirical mean of the regret, whereas the opaque bands correspond to the 10-th and 90-th percentile.

**DIFFERENT DISCRETIZATIONS.** In Figure 10.2 we present the comparative analysis of the estimated average regret of WIN-EXP vs EXP3 for different discretizations,  $\varepsilon$ , of the bidding space when the learner faces adversaries that are stochastic, adaptive using EXP3 and adaptive using WINEXP. As it was expected from the theoretical analysis, the regret of WIN-EXP, as the discretized space ( $|B|$ ) increases exponentially, remains almost unchanged compared to the regret of EXP3. In summary, *finer discretization of the bid space helps our WIN-EXP algorithm's performance, but hurts the performance*

of EXP3.

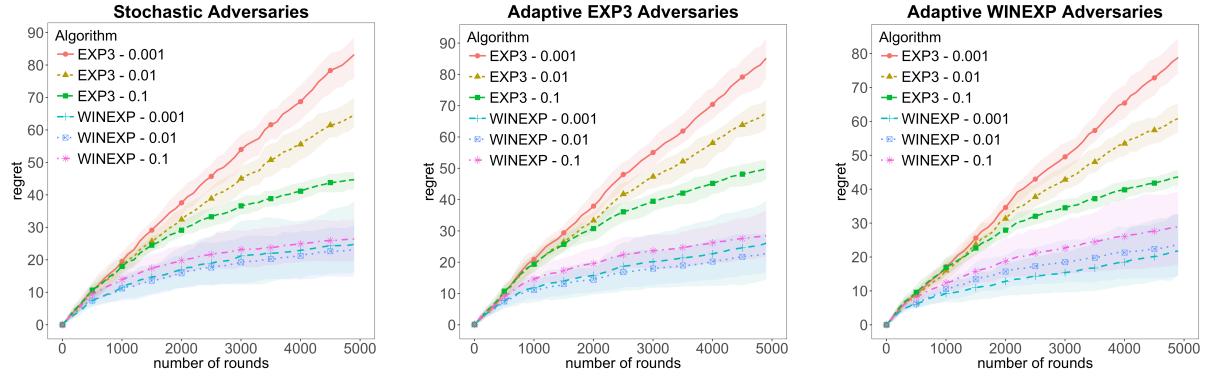


Figure 10.2: Regret of WIN-EXP vs EXP3 for different discretizations  $\varepsilon$  ( $CTR \sim U[0.5, 1]$ ).

**DIFFERENT CTR DISTRIBUTIONS.** In Figures 10.3, 10.4 and 10.5 we present the results of the regret performance of WIN-EXP compared to EXP3, when the learner discretizes the bidding space with  $\varepsilon = 0.01$  and when she faces stochastic, adaptive adversaries using EXP3 and adaptive adversaries using WINEXP, respectively. For all three cases, the estimated average regret of WIN-EXP is less than the estimated average regret that EXP3 yields.

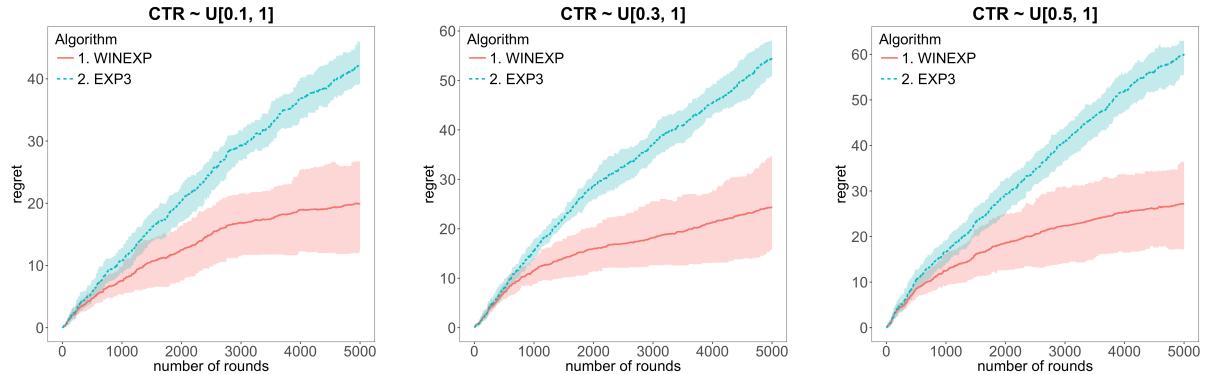


Figure 10.3: Regret of WIN-EXP vs EXP3 for different CTR distributions and stochastic adversaries,  $\varepsilon = 0.01$ .

**ROBUSTNESS TO NOISY CTR ESTIMATES.** In Figures 10.6, 10.7, 10.8 we empirically tested the robustness of our algorithm to random perturbations of the allocation function that the auctioneer presents to the learner, for perturbations of the form  $\mathcal{N}(0, \frac{1}{m})$ , where  $m$  could be viewed as the number of training examples used from the auctioneer in order to derive an approximation of the allocation

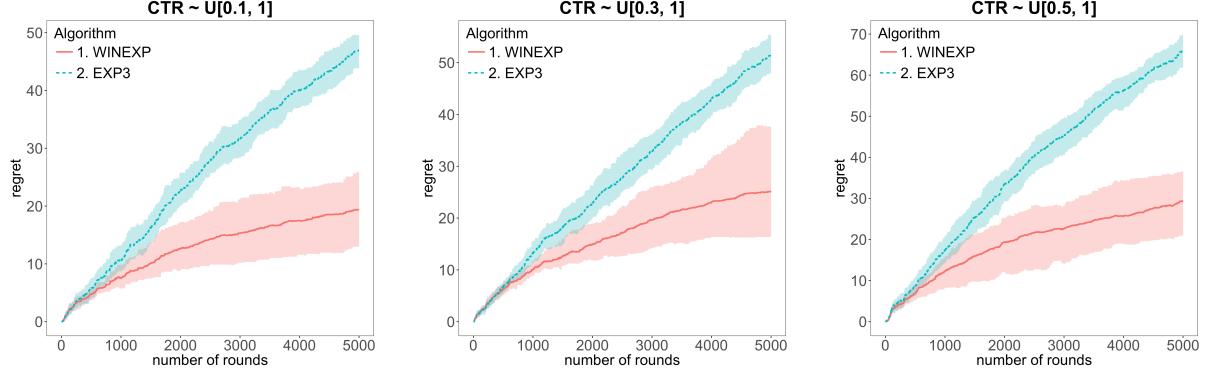


Figure 10.4: Regret of WIN-EXP vs EXP3 for different CTR distributions and adaptive EXP3 adversaries,  $\varepsilon = 0.01$ .

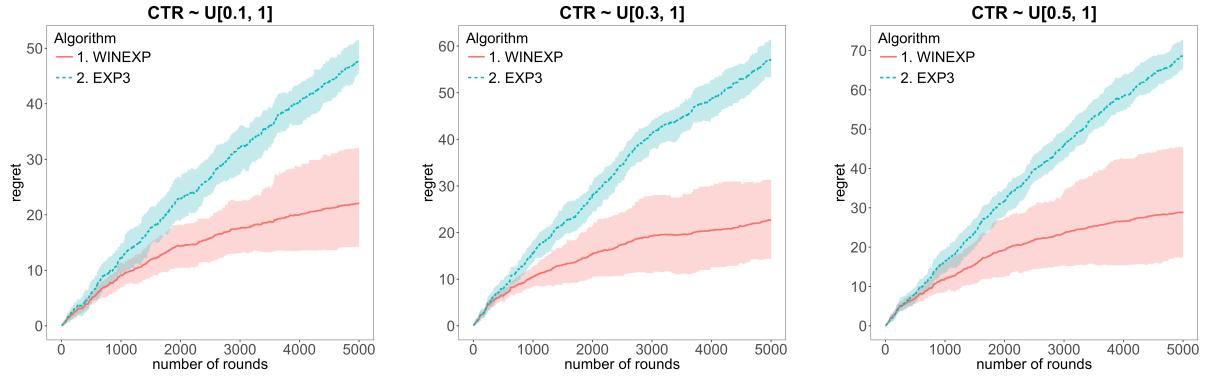


Figure 10.5: Regret of WIN-EXP vs EXP3 for different CTR distributions and adaptive WINEXP adversaries,  $\varepsilon = 0.01$ .

curve. When the number of training samples is relatively small ( $m = 100$ ) the empirical mean of WINEXP outperforms EXP3 in terms of regret, i.e., it is more robust to such perturbations. As the number of training samples increases, WINEXP clearly outperforms EXP3. The latter validates one of our claims throughout the paper; namely, that even though the learner might not see the exact allocation curve, but a randomly perturbed proxy, WIN-EXP still performs better than the EXP3.

## 10.8 DISCUSSION AND OPEN QUESTIONS

We addressed learning in repeated mechanism design scenarios where players do not know their valuation for the items at sale. We formulated an online learning framework with partial feedback which captures the information available to players in typical auction settings like sponsored search and provided an algorithm which achieves almost full information regret rates. Hence, we

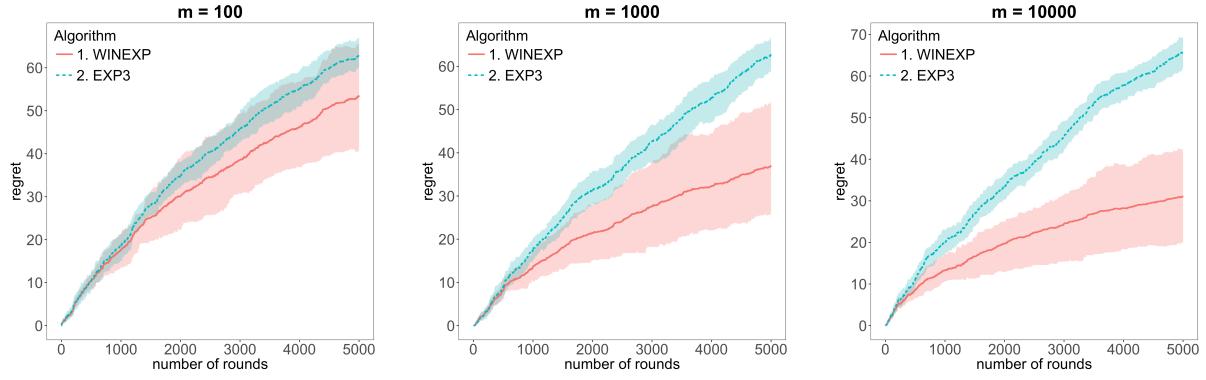


Figure 10.6: Regret of WIN-EXP vs EXP3 with noise  $\sim \mathcal{N}(0, \frac{1}{m})$  for stochastic adversaries,  $\varepsilon = 0.01$ .

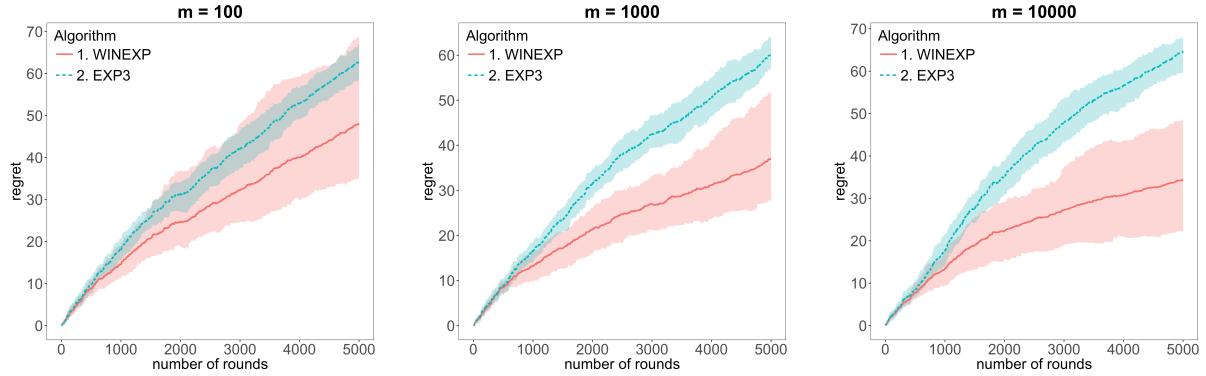


Figure 10.7: Regret of WIN-EXP vs EXP3 with noise  $\sim \mathcal{N}(0, \frac{1}{m})$  for adaptive EXP3 adversaries,  $\varepsilon = 0.01$ .

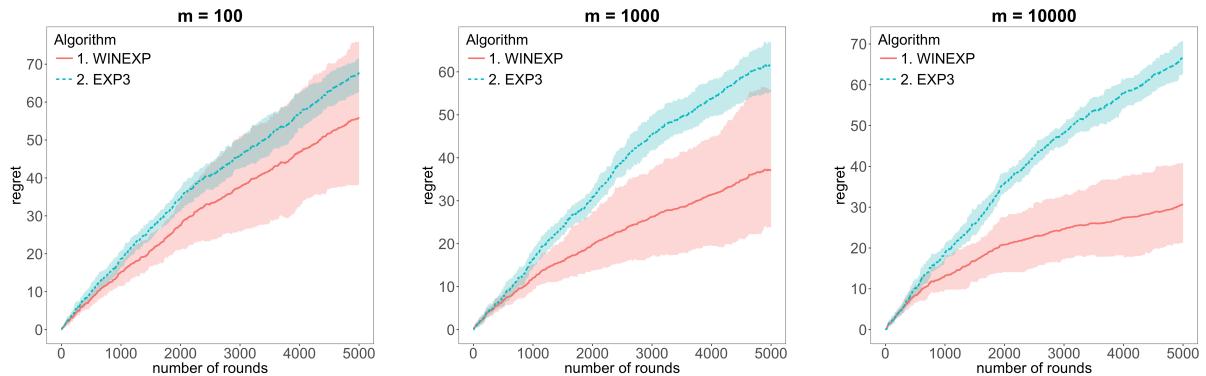


Figure 10.8: Regret of WIN-EXP vs EXP3 with noise  $\sim \mathcal{N}(0, \frac{1}{m})$  for adaptive WINEXP adversaries,  $\varepsilon = 0.01$ .

portrayed that not knowing your valuation is a benign form of incomplete information learning in auctions. Our experimental evaluation also showed that the improved learning rates are robust to violations of our assumptions and are valid even when the information assumed is corrupted. We believe that exploring further avenues of relaxing the informational assumptions or being more robust to erroneous information given by the auction system is an interesting future research direction. We believe that our outcome based learning framework can facilitate such future work.

# 11

## The Platform Perspective: Bandits with Long-Term Effects

### 11.1 CHAPTER OVERVIEW

In this chapter, we focus on the online advertising problem from the decision-maker/platform perspective. We do so by studying a bandit learning problem where the choices made in each round have long-term environmental impact. The reward collected by the algorithm in each round is a function of both the short term reward of the arm and of how healthy the environment is. As we mentioned already from Chapter 1, the success of online advertising depends on the increased engagement of the users with the ads, who do so according to their “value”/“utility” for the content

it represents. Understanding better what drives user engagement has been a major research question since the advent of online advertising not just because of its potential to drive revenue, but also, due to its potential to increase user satisfaction. Despite the proliferation of models put forth to explain user behavior, most of them have focused on users that are short-sighted/myopic; i.e., users who make engagement decisions not caring about their prior interactions with the system.

To showcase the generality of our results, in the proceeding sections we abstract away from the auction setting and prove our results for a general bandit optimization viewpoint.

Our first contribution is to propose a model for learning to choose a sequence of actions, which captures the long-term effects of prior decisions. To the best of our knowledge, we are the first to propose such a model capturing ad blindness/sightedness in the context of bandit learning. We state our model and results for a more general, abstract setting, as they are useful to capture other applications of learning with long-term effects too (see e.g., crop rotation).

Crucial to our model is the notion of a “state”, which captures the effects of the sequence of arms pulled so far to the reward that the learner obtains at each round. We define a *speed parameter*  $\lambda \in [0, 1]$ , which controls how fast the state evolves. Note that a speed parameter of  $\lambda = 0$  corresponds to the standard  $K$ -Multi-Armed-Bandit (MAB) problem. The main hardness that arises in our setting is that although the way that the state affects the expected rewards is known, the state itself and the underlying expected rewards are unknown.

Our results are threefold. First, we obtain a deep understanding of the optimal sequence of arms defined in the benchmark for the regret and how one could approximate it, had they been given access to an offline oracle using a dynamic programming procedure on approximations of the expected rewards (Chapter 11.3). Second, we provide an algorithm for general speed parameters  $\lambda \in (0, 1)$  achieving regret  $\tilde{\mathcal{O}}((K \log(\lambda) / \log(1 - \lambda))^{1/3} \cdot T^{2/3})$  (Chapter 11.4). Third, we study the special case of “sticky” arms where  $\lambda = 1$ . For this case, we show that the optimal sequence of arms includes only cycles of size at most 2, and we subsequently provide an algorithm with regret  $\tilde{\mathcal{O}}(K^2 \sqrt{T})$  (Chapter 11.5). We conclude with various open questions and research directions.

### 11.1.1 RELATED WORK

Closest to this chapter is the work of [Hohnhold et al. \(2015\)](#), who also studied models for ad blindness/ad sightedness. The present chapter has orthogonal strengths. [Hohnhold et al. \(2015\)](#) first estimate the ad blindness/sightedness parameters and then they use these to redesign online ad auctions. We, instead, study a more fundamental learning setting, our results are not calibrated to a single search engine, and our algorithms cover other settings with long-term effects as well.

From the online learning literature, the present chapter has connections with papers both on Multi-Armed Bandit (MAB) problems and more general RL settings. There has been a lot of recent interest in settings where the expected rewards of the arms evolve over time. [Levine et al. \(2017\)](#) and [Seznec et al. \(2019\)](#) study “rotting bandits”, where the long-term effect is that as you pull an arm the realized reward presented to the learner decreases. The main difference with our problem is that in “rotting bandits” there is no way to “replenish” what you lost from an arm as you kept pulling it. Additionally, the benchmark policy in rotting bandits is to greedily play the optimal arm at each round, had you known everything in advance, which is not at all the case in our setting.

[Kleinberg and Immorlica \(2018\)](#) study “recharging bandits”, where rewards accrue as time goes by since the last time the arm was played. In “blocking bandits” ([Basu et al., 2019, 2021](#), [Bishop et al., 2020](#)) playing an arm makes it unavailable for a fixed number of time slots thereafter. In [Heidari et al. \(2016\)](#), [Leqi et al. \(2021\)](#), the rewards of the arms increase/decrease as they get played. In “rested bandits” ([Gittins, 1979](#)) an arm’s expected rewards change only when it is played. In “restless bandits” ([Whittle, 1988](#)) rewards evolve independently from the play of each arm. In ([Cella and Cesa-Bianchi, 2020](#)) the rewards increase as a function of the time elapsed since the last pull. In “recovering bandits” ([Pike-Burke and Grunewalder, 2019](#)) the expected reward of an arm is expressed as a function of the time since the last pull drawn from a Gaussian Process with known kernel. In [Warlop et al. \(2018\)](#), the rewards are linear functions of the recent history of actions. In ([Mintz et al., 2020](#)), rewards are a function of a context that evolve according to known deterministic dynamics. In our case, the *inherent* rewards of the arms per se do not change; instead, they are filtered through the state which is affected by all previously played arms.

[Lykouris et al. \(2020\)](#) consider the case where the arms have a stochastic component and an

adversarial one, which is chosen at each round by adversary. The final mean reward is the product between the stochastic and adversarial components. The difference with our setting is that in our case, stochastic reward is multiplied by the *state*, which is defined deterministically based on the sequence of prior actions, and cannot be chosen arbitrarily by an adversary. In a similar vein, [Gupta et al. \(2021\)](#) consider the setting where the rewards of pulling different arms are correlated. In our case, rewards are also correlated but they are governed by the state. Correlations also really arise once you pull the arms sequentially, as opposed to their problem, where correlation *requires* arms to be pulled simultaneously.

This chapter is also related to RL with MDPs with deterministic transition functions (e.g., [\(Ortner, 2008, Dekel and Hazan, 2013\)](#) for stochastic and adversarial respectively) and with [\(Ortner and Ryabko, 2012\)](#) which studies a stochastic RL setting with a continuous state space. The core difference with this chapter, however, is that the aforementioned works assume that the learner can observe the state that they find themselves at each round.

## 11.2 MODEL

We model the problem as a  $K$ -Multi-Armed-Bandit (MAB) instance. Each arm  $i \in [K]$  is associated with tuple  $(r_i, b_i) \in [0, 1]^2$ .  $r_i$  denotes the *in-the-void reward* (IVR) of arm  $i$ , i.e., the reward sampled from this arm if it were to be played in isolation, and abstracting away from the long-term effects of previously pulled arms.  $b_i$  denotes the *baseline reward* (BR) of this arm if one were to play it for an infinite number of rounds as a result of the long-term effects. To capture said effects, we use the notion of a “state”; at each round  $t$ , the learner finds themselves at a different state of the environment depending on the sequence of arms pulled so far. Let  $I_t$  denote the arm chosen at round  $t$ , and  $H_{s:t}^{\text{ALG}}$  the history of arms played by algorithm ALG from round  $s$  until round  $t$ , i.e.,  $H_{s:t}^{\text{ALG}} = \{I_\tau\}_{\tau=s}^t$ . The *state* at round  $t$  when pulling arms according to algorithm ALG is defined as:

$$q_{t+1}(H_{1:t}^{\text{ALG}}) = q_t(H_{1:t-1}^{\text{ALG}}) + \lambda \cdot (b_{I_t} - q_t(H_{1:t-1}^{\text{ALG}})) = (1 - \lambda) \cdot q_t(H_{1:t-1}^{\text{ALG}}) + \lambda \cdot b_{I_t}, \quad (11.2.1)$$

where  $\lambda$  is a known speed parameter controlling how much the present state is affected by the most recently pulled arm versus the earlier arms. Equation (11.2.1) captures the fact that we take gradient steps on the state function parametrized by arm  $I_t$  (see Fig. 11.1). Note that  $\lambda = 0$  corresponds to the standard MAB for which the optimal regret is  $\tilde{\mathcal{O}}(\sqrt{TK})^*$  (Auer et al., 2002b). We use  $q_0$  to denote the initial state. When clear from context, we drop the explicit dependence of  $q_t(\cdot)$  on the history.

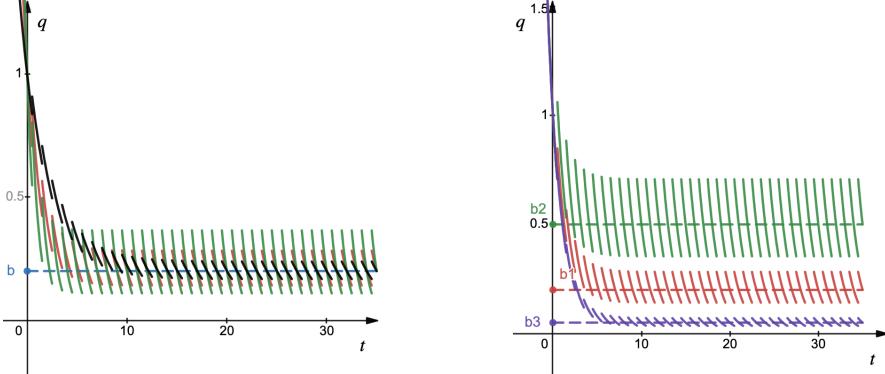


Figure 11.1: State evolution for different parameters. **Left:** fixed arm with  $b_i = 0.2$ , varying  $\lambda$ . The green plot is for  $\lambda = 0.7$ , the red for  $\lambda = 0.5$ , and the black for  $\lambda = 0.3$ . **Right:** varying arms, fixed  $\lambda = 0.5$ . The green plot is for  $b_i = 0.5$ , the red for  $b_i = 0.2$ , and the purple for  $b_i = 0.05$ .

At each round  $t$ , the learning protocol is: First, the learner pulls arm  $I_t \in [K]$ . Second, they observe reward that is sampled from  $\text{Bern}(r_{I_t} \cdot q_t(H_{1:t-1}^{\text{ALG}}))$ . Third, the state is updated as in Eq. (11.2.1).

Importantly, the learner never gets to observe the current state  $q_t(H_{1:t-1}^{\text{ALG}})$  and they also never observe the tuple  $(r_{I_t}, b_{I_t})$ . The learner's goal is to choose a sequence of arms  $\{I_t\}_{t \in [T]}$  that minimize a notion of regret. Let  $\text{OPT}$  denote the policy choosing the sequence of arms to maximize the expected reward when the tuples  $(r_i, b_i)_{i \in [K]}$  are known. Then, the regret is defined as:

$$\text{REGRET}(T) = \mathbb{E} \left[ \sum_{t \in [T]} r_{I_t^{\text{OPT}}} \cdot q_t(H_{1:t-1}^{\text{OPT}}) \right] - \mathbb{E} \left[ \sum_{t \in [T]} r_{I_t} \cdot q_t(H_{1:t-1}^{\text{ALG}}) \right] \quad (11.2.2)$$

**TRANSLATION TO THE ONLINE ADS EXAMPLE.** Before we move on to the technical sections of the chapter, we find it useful to translate the general model to our main motivating example. For online ads, the *arms* correspond to *ads* and their optimal sequence corresponds to the optimal ad schedule. The

\*As is customary in the literature,  $\tilde{\mathcal{O}}(\cdot)$  hides terms poly-logarithmic in  $K, T$ .

*state* of round  $t$  corresponds to the user's propensity to click after engaging with the system for  $t$  ads. The speed  $\lambda$  corresponds to the speed according to which *ad sightedness/blindness* affects the user's satisfaction from round to round. The *in-the-void reward* of an arm corresponds to the *inherent click-through-rate (CTR)* that the ad would have for a given user had there not been long-term effects. The *baseline reward* of an arm corresponds to the baseline sightedness/blindness of the respective arm, had it been presented constantly. The fact that at each round  $t$  the reward is sampled from  $\text{Bern}(r_{I_t} \cdot q_t(H_{1:t-1}^{\text{ALG}}))$  translates to observing a click with probability  $r_{I_t} \cdot q_t(H_{1:t-1}^{\text{ALG}})$ .

### 11.3 RELAXATION: DYNAMIC PROGRAMMING WITH APPROXIMATE REWARDS

In this section, we discuss the problem's relaxation, where for each arm the learner has estimates about  $(r_i, b_i)$ . Subsequently, we show the efficiency (compared to OPT) of a Dynamic Programming approach that takes as input these estimates and outputs a sequence of arms. We first prove the closed form solution for the state at each  $t$ .

**Lemma 11.1.** *Let ALG be an algorithm pulling arm  $I_t$  at round  $t$ . The closed form solution for computing the state at each round is:*

$$q_{t+1} \left( H_{1:t}^{\text{ALG}} \right) = (1 - \lambda)^t \cdot q_0 + \lambda \cdot \sum_{s=0}^{t-1} (1 - \lambda)^{t-1-s} \cdot b_{I_s} \quad (11.3.1)$$

*Proof.* We prove the lemma using induction. For the base case  $t = 1$ , from Equation (11.2.1) it holds:

$$q_1 \left( H_{1:1}^{\text{ALG}} \right) = (1 - \lambda) \cdot q_0 \left( H_0^{\text{ALG}} \right) + \lambda \cdot b_{I_1} = (1 - \lambda) + \lambda \cdot b_{I_1},$$

which is equal to  $q_{t+1}(H_{1:t}^{\text{ALG}}) = (1 - \lambda)^1 \cdot q_0 + \lambda \cdot (1 - \lambda)^0 \cdot b_{I_1}$  from Equation (11.3.1).

For the inductive step, assume that Equation (11.3.1) holds for some  $t = n$ . Then, for  $t = n + 1$

from Equation (11.2.1) we have that:

$$\begin{aligned}
q_{n+2}(H_{1:n+1}^{\text{ALG}}) &= (1 - \lambda) \cdot q_{n+1}(H_{1:n}^{\text{ALG}}) + \lambda \cdot b_{i_{n+1}} \\
&= (1 - \lambda) \left[ (1 - \lambda)^{n+1} \cdot q_0 + \lambda \cdot \sum_{s=0}^n (1 - \lambda)^{n-s} \cdot b_{I_s} \right] + \lambda \cdot b_{I_{n+1}} \quad (\text{inductive step}) \\
&= (1 - \lambda)^{n+2} \cdot q_0 + \lambda \cdot \sum_{s=0}^n (1 - \lambda)^{n+1-s} \cdot b_{I_s} + \lambda \cdot b_{I_{n+1}} \\
&= (1 - \lambda)^{n+2} \cdot q_0 + \lambda \cdot \sum_{s=0}^{n+1} (1 - \lambda)^{n+1-s} \cdot b_{I_s}
\end{aligned}$$

which is exactly the form that  $q_{n+2}(H_{1:n+1}^{\text{ALG}})$  takes from Equation (11.3.1).  $\blacksquare$

This closed-form solution for the state is very important as it allows us to directly decompose  $q_{t+1}(H_t^{\text{ALG}})$  to the baseline rewards of the arms pulled so far. This is helpful because, as we argue below, we do not need to have full knowledge of the exact baseline rewards; instead, good approximations are enough to give us a solution that is close to the OPT.

**Lemma 11.2.** *Let  $\widehat{\text{DP}}$  denote the expected reward of the solution returned by a dynamic programming algorithm with inputs  $(\widehat{r}_i, \widehat{b}_i)_{i \in [K]}$ , where  $|\widehat{r}_i - r_i| \leq \delta$  and  $|\widehat{b}_i - b_i| \leq \delta$ . This dynamic programming algorithm outputs a policy for choosing an arm at each round. Then,  $\widehat{\text{DP}} \geq \text{OPT} - \delta T$ .*

*Proof.* Let  $\pi_1, \dots, \pi_T$  be the sequence of actions chosen by OPT given tuples  $(r_i, b_i), \forall i$  as input.

$$\begin{aligned}
\widehat{\text{DP}} &= \max_{i_1, \dots, i_T} \sum_{t \in [T]} \left[ (1 - \lambda)^t + \lambda \sum_{s=0}^{t-1} (1 - \lambda)^{t-1-s} \widehat{b}_{i_s} \right] \cdot \widehat{r}_{i_t} \quad (\text{Equation (11.3.1)}) \\
&\geq \max_{i_1, \dots, i_T} \sum_{t \in [T]} \left[ (1 - \lambda)^t + \lambda \sum_{s=0}^{t-1} (1 - \lambda)^{t-1-s} (b_{i_s} - \delta) \right] \cdot (r_{i_t} - \delta) \quad (\widehat{r}_i \geq r_i - \delta, \widehat{b}_i \geq b_i - \delta) \\
&\geq \max_{i_1, \dots, i_T} \sum_{t \in [T]} \left[ (1 - \lambda)^t + \lambda \sum_{s=0}^{t-1} (1 - \lambda)^{t-1-s} b_{i_s} \right] \cdot r_{i_t} - \sum_{t \in [T]} \lambda \sum_{s=0}^{t-1} (1 - \lambda)^{t-1-s} \cdot \delta \\
&\geq \max_{i_1, \dots, i_T} \sum_{t \in [T]} \left[ (1 - \lambda)^t + \lambda \sum_{s=0}^{t-1} (1 - \lambda)^{t-1-s} b_{i_s} \right] \cdot r_{i_t} - \sum_{t \in [T]} \lambda \cdot \frac{1}{\lambda} \cdot \delta \\
&\geq \sum_{t \in [T]} \left[ (1 - \lambda)^t + \lambda \sum_{s=0}^{t-1} (1 - \lambda)^{t-1-s} b_{\pi_s} \right] \cdot r_{\pi_t} - \delta \cdot T \quad (\text{properties of max}) \\
&= \text{OPT} - \delta T
\end{aligned}$$

where the second inequality also uses the fact that  $r_i, b_i \in [0, 1]$ . ■

Below, we use a dynamic programming idea as follows. First, we show how to obtain the estimates  $(\hat{r}_i, \hat{b}_i)_{i \in [K]}$  if there exists a *known* “replenishing” arm  $i_R$ , for which it holds that  $b_{i_R} \in [1 - 2\varepsilon, 1 - \varepsilon]$ . Second, we show a method for relaxing this assumption while still obtaining meaningful estimates.

#### 11.4 ALGORITHM FOR GENERAL SPEED PARAMETERS

In this section, we focus on the case where  $\lambda \in (0, 1)$  (i.e.,  $\lambda$  is a general speed parameter), and we present an algorithm that learns to pull the arms in sequences that achieve regret  $\tilde{\mathcal{O}}(K^{1/3}T^{2/3})$ . For the ease of exposition, we describe the results of this section with a simplifying assumption; namely, that there exists a known “replenishing” arm  $i_R$  for which it holds that  $b_{i_R} \in [1 - 2\varepsilon, 1 - \varepsilon]$ . At the end of the section, we explain how the general case (without the assumption) can be analyzed.

We start with two thought experiments. For the first one, assume that the  $b_i$ ’s were known. Due to Lemma 11.1, this would then translate to us knowing the state at which we are at any round. In that case, we could simply build estimators  $\hat{r}_i$  for the  $r_i$ ’s such that  $|\hat{r}_i - r_i| \leq \delta$  with high probability. Given the  $\hat{r}_i$ ’s and the actual  $b_i$ ’s we could then feed  $(\hat{r}_i, b_i)_{i \in [K]}$  to the dynamic program and according to Lemma 11.2 obtain a solution that is  $\delta T$  close to the OPT. Tuning  $\delta$  appropriately would then give us a no-regret algorithm. The challenge is that  $b_i$ ’s are also unknown and this means that generally, we cannot understand the state where the system is at any point.

For the second thought experiment, assume that  $r_i$ ’s are now known, but the  $b_i$ ’s are not. Similarly to before, we could now build estimators  $\hat{b}_i$  that are  $\delta$ -close to  $b_i$ , and then use again the dynamic programming solution. Again, we cannot really use this solution as-is, since both the  $r_i$ ’s and the  $b_i$ ’s are not known.

The main idea of our algorithm is to first to build estimators for the  $r_i$ ’s and subsequently, use these when trying to infer the  $b_i$ ’s. The tricky part is to make sure that we can still estimate correctly the  $b_i$ ’s despite only having approximations of the  $r_i$ ’s and while the two quantities are connected multiplicatively. One key property of our setting is that irrespective of the history of plays and the starting state, then playing repeatedly the same arm  $i$  for a fixed number of  $N$  rounds makes the

state become approximately equal to  $b_i$ . Moreover,  $N$  is constant with respect to  $\varepsilon$  and  $T$ .

**Lemma 11.3.** *Fix an arm  $i \in [K]$  and a scalar  $\varepsilon > 0$ . Assume that at some round  $s$ , after a history of plays  $H'$ , we are at state  $q_s$ . Then, playing repeatedly arm  $i$  for  $N(\lambda) \geq c(\lambda) \cdot \log(1/(\lambda\varepsilon))$  (where  $c(\lambda) = \log^{-1}(1/(1-\lambda))$ ) rounds makes the state become  $\tilde{q}$ , such that:  $|q_{N(\lambda)} - b_i| \leq \varepsilon$ .*

*Proof.* Let  $\text{REP}_i$  be the algorithm that continuously plays arm  $i$ , and let  $\tilde{H}_{s:t}^{\text{REP}_i} = H_{s:t}^{\text{REP}_i} \cup H'$ . We first prove by induction that if  $i = i_\tau, \forall \tau \in \{1, \dots, N(\lambda)\}$ , then:

$$q_{s+\tau+1} \left( \tilde{H}_{s:s+\tau}^{\text{REP}_i} \right) - b_i = (1 - \lambda)^{\tau+1} (q_s - b_i). \quad (11.4.1)$$

For the base case  $\tau = 1$ , note that  $q_{s+1}(H') - b_i = (1 - \lambda)(q_s - b_i)$ , which is equal to the definition in Eq. (11.2.1), if the first round was  $s$  instead of 1. For the inductive step, assume that the claim holds for  $\tau = n$ , i.e.,

$$q_{s+n+1} \left( \tilde{H}_{s:s+n}^{\text{REP}_i} \right) - b_i = (1 - \lambda)^{n+1} (q_s - b_i) \quad (11.4.2)$$

Then, for  $\tau = n + 1$ , from Equation (11.2.1), we have:

$$\begin{aligned} q_{s+n+2} \left( \tilde{H}_{s:s+n+1}^{\text{REP}_i} \right) &= (1 - \lambda) q_{s+n+1} \left( \tilde{H}_{s:s+n}^{\text{REP}_i} \right) + \lambda b_i \Leftrightarrow \\ q_{s+n+2} \left( \tilde{H}_{s:s+n+1}^{\text{REP}_i} \right) - b_i &= (1 - \lambda) \left( q_{s+n+1} \left( \tilde{H}_{s:s+n}^{\text{REP}_i} \right) - b_i \right) \end{aligned}$$

Substituting Equation (11.4.2) in the latter completes the proof of the induction.

To simplify notation, let us use  $q_{\tau+1} = q_{s+\tau+1}(\tilde{H}_{s:s+\tau}^{\text{REP}_i})$ . Taking the absolute on both sides of Equation (11.4.1) we get:

$$|q_{N(\lambda)} - b_i| = \left| (1 - \lambda)^{N(\lambda)} (q_s - b_i) \right|$$

Substituting the expression for  $N(\lambda)$  from the lemma statement, we get:

$$\begin{aligned}
|q_{N(\lambda)} - b_i| &= \left| (1 - \lambda)^{\frac{\log(\lambda\varepsilon)}{\log(1-\lambda)}} (q_s - b_i) \right| \leq \left| (1 - \lambda)^{\frac{\log(\lambda\varepsilon)}{\log(1-\lambda)}} \right| \cdot |q_s - b_i| && (\text{Cauchy-Schwarz}) \\
&\leq \left| (1 - \lambda)^{\frac{\log(\lambda\varepsilon)}{\log(1-\lambda)}} \right| && (q_s, b_i \in [0, 1]) \\
&= 2^{\frac{\log(\lambda\varepsilon)}{\log(1-\lambda)} \cdot \log(1-\lambda)} \\
&= \lambda\varepsilon \leq \varepsilon && (\lambda \in (0, 1))
\end{aligned}$$

This concludes our proof. ■

An important corollary is that irrespective of the history of plays and the current state, if one were to play the replenishing arm for  $N_R := N(\lambda)$  rounds, then, the state returns (approximately) to  $q_0$ .

**Corollary 11.1.** *Let  $q_s$  be the state reached at some round  $s$  after history of plays  $H'$ . Then, playing repeatedly  $i_R$  for  $N_R \geq c(\lambda) \cdot \log(1/(\lambda\varepsilon))$  (where  $c(\lambda) = \log^{-1}(1/(1-\lambda))$ ) times makes the state become  $q_{N_R} = 1 - 2\varepsilon + \lambda\varepsilon^2 > 1 - 2\varepsilon$ .*

*Proof.* Similarly to the proof of Lemma 11.3, let  $\text{REP}_i$  be the algorithm that continuously plays arm  $i$ , and let  $\tilde{H}_{s:t}^{\text{REP}_i} = H_{s:t}^{\text{REP}_i} \cup H'$ . Then, from Equation (11.4.1) simplifying notation:  $q_{N_R} = q_{s+N_R}(\tilde{H}_{s:N_R-1}^{\text{REP}_i})$ , we get:

$$q_{N_R} - b_{i_R} = (1 - \lambda)^{N_R} \cdot (q_0 - b_{i_R})$$

Using the fact that  $1 - 2\varepsilon \leq b_{i_R} \leq 1 - \varepsilon$ , the latter becomes:

$$q_{N_R} - (1 - 2\varepsilon) \geq (1 - \lambda)^{N_R} \cdot (q_0 - (1 - \varepsilon))$$

Substituting for  $q_0 = 1$  and  $N_R$  as given in the lemma statement:

$$q_{N_R} - (1 - 2\varepsilon) \geq \lambda\varepsilon \cdot (\varepsilon)$$

Re-arranging, we obtain the result. ■

We are now ready to state our algorithm. Note that for Lemmas 11.4, 11.5 and 11.6 that follow, we use a fixed  $\varepsilon$ . We tune this  $\varepsilon$  optimally in the end to obtain the no-regret guarantee.

---

**Algorithm 11.1: MAB with Long-Term Effects Algorithm with Known  $i_R$** 


---

```

1 Set  $\varepsilon, \delta, M$  as stated in Theorem 11.1.
2 Initialize rounds  $t = 1$ .
   /* Explore in-the-void rewards and build their estimators:  $\{\hat{r}_i\}_{i \in [K]}$  */
3 for arm  $i \in [K]$  do
4   Initialize reward estimate  $\hat{r}_i = 0$ .
5   for blocks  $j \in [M]$  do
6     for pulls  $1, \dots, N_R$  do // Restore the state to at least  $1 - \varepsilon$ 
7       Play arm  $i_R$ .
8       Update  $t \leftarrow t + 1$ .
9     Play arm  $i$ , observe reward  $R_j^i$ , and update:  $\hat{r}_i \leftarrow \hat{r}_i + \frac{R_j^i}{M}$ . // Play  $i$  when  $q \approx 1 - \varepsilon$ .
10    Update  $t \leftarrow t + 1$ .
11   /* Explore baseline rewards and build estimators:  $\{\hat{b}_i\}_{i \in [K]}$  */
12  for arm  $i \in [K]$  do
13    Initialize state estimator  $\hat{v}_i = 0$ .
14    for blocks  $j \in [M]$  do
15      for pulls  $1, \dots, N(\lambda)$  do
16        Play arm  $i$ .
17        Update  $t \leftarrow t + 1$ .
18      Play arm  $i$ , observe reward  $S_j^i$ , and update:  $\hat{v}_i \leftarrow \hat{v}_i + \frac{S_j^i}{M}$ . // Play  $i$  when  $q \approx b_i$ 
19    Compute baseline reward estimator:  $\hat{b}_i = \hat{v}_i / \hat{r}_i$ .
20  Play  $i_R$  for  $N_R$  rounds & update  $t \leftarrow t + 1$  after each one. // Restore state to at least  $1 - \varepsilon$ 
21  Feed  $(\hat{r}_i, \hat{b}_i)$  in the Dynamic Programming algorithm and play the solution until end  $T$ .

```

---

**Notation.** To simplify the exposition and the notation with the explicit dependence on the history of plays, we denote with  $t_j^i$  the round  $t$  after the final play of arm  $i_R$  during block  $j$  for arm  $i$  (i.e., Line 9), and with  $\tilde{t}_j^i$  the round  $t$  after the final play of arm  $i$  during block  $j$  for arm  $i$  (i.e., Line 17).

**Theorem 11.1: MAB with Long-Term Effects Regret for Known  $i_R$**

Tuning  $\delta = \varepsilon/4$ ,  $M = \ln(T)/\varepsilon^2$  and

$$\varepsilon = \left( \frac{K \cdot \ln(T) \cdot \log(\lambda)}{T \cdot \log(1 - \lambda)} \right)^{1/3},$$

Algorithm 11.1 incurs regret  $\text{REGRET}(T) = \mathcal{O} \left( \left( \frac{K \ln(T) \log(\lambda)}{\log(1 - \lambda)} \right)^{1/3} \cdot T^{2/3} \right)$ .

We first prove that the reward estimators we build are good approximations for the true rewards.

**Lemma 11.4.** *For the in-the-void reward estimator of each arm  $i$  in Line 9 of Algorithm 11.1 and any scalar  $\delta > 0$ , it holds that:  $\Pr[|\hat{r}_i - r_i| \geq \delta] \leq 2 \exp(-2M \cdot (\delta - \varepsilon)^2)$ .*

*Proof.* From Hoeffding's inequality on  $\hat{r}_i$  and using the fact that the block size is  $M$  rounds, we get:

$$\Pr[|\hat{r}_i - \mathbb{E}[\hat{r}_i]| \geq \delta] \leq 2 \exp(-2M\delta^2) \quad (11.4.3)$$

From Corollary 11.1, regardless of the starting state and the prior history, if arm  $i_R$  is played repeatedly for  $N_R$  rounds, then at round  $t_j^i$  the system's state is at  $q_{t_j^i} \geq 1 - \varepsilon$ . So (by definition of our setting) the expected reward at the right next round (i.e., Line 9 of Algorithm 11.1) is

$$\mathbb{E}[R_j^i] = q_{t_j^i} \cdot r_i \in [(1 - \varepsilon) \cdot r_i, r_i],$$

with probability 1. As a result, by the linearity of expectation and using the definition of  $\hat{r}_i$ :

$$\mathbb{E}[\hat{r}_i] = \frac{\mathbb{E}[R_j^i]}{M} = \frac{\sum_{j \in [M]} q_{t_j^i} \cdot r_i}{M} = r_i \cdot \frac{\sum_{j \in [M]} q_{t_j^i}}{M} \Rightarrow \mathbb{E}[\hat{r}_i] \in [r_i \cdot (1 - \varepsilon), r_i]$$

From Equation (11.4.3), we have that:

$$\begin{aligned} 2 \exp(-2M\delta^2) &\geq \Pr[\hat{r}_i - \mathbb{E}[\hat{r}_i] \geq \delta \text{ or } \hat{r}_i - \mathbb{E}[\hat{r}_i] \leq -\delta] \\ &\geq \Pr[\hat{r}_i \geq r_i + \varepsilon + \delta \text{ or } \hat{r}_i - \mathbb{E}[\hat{r}_i] \leq -\delta] \quad (\mathbb{E}[\hat{r}_i] \leq r_i \leq r_i + \varepsilon) \\ &\geq \Pr[\hat{r}_i \geq r_i + \varepsilon + \delta \text{ or } \hat{r}_i \leq r_i - \varepsilon - \delta] \quad (\mathbb{E}[\hat{r}_i] \geq r_i \cdot (1 - \varepsilon) \geq r_i - \varepsilon) \\ &= \Pr[|\hat{r}_i - r_i| \geq \delta + \varepsilon] \end{aligned}$$

Using as  $\delta' = \delta + \varepsilon$  and substituting in the above gives us the result. ■

For arm  $i \in [K]$ , let  $v_i = r_i \cdot b_i$ . Then, we denote as  $\hat{v}_i$  the estimator of  $v_i$  for each arm  $i$  through Algorithm 11.1. We prove below that  $\hat{v}_i$  is a good estimator for  $v_i$  for all  $i \in [K]$ .

**Lemma 11.5.** *For estimator  $\hat{v}_i$  of arm  $i$  in Line 17 of Algorithm 11.1 and any  $\delta > 0$ , it holds:  $\Pr[|\hat{v}_i - v_i| \geq \delta] \leq 2 \exp(-2M \cdot (\delta - \varepsilon)^2)$ .*

*Proof.* The proof is similar to the proof of Lemma 11.4, but we include it for completeness. From Hoeffding's inequality on  $\hat{v}_i$  and using the fact that the block size is  $M$  rounds, we get:

$$\Pr [|\hat{v}_i - \mathbb{E}[\hat{v}_i]| \geq \delta] \leq 2 \exp(-2M\delta^2) \quad (11.4.4)$$

From Lemma 11.3, regardless of the history of plays, if you start from state  $q_0$  and play the same arm for  $N(\lambda)$  rounds, then the state becomes approximately equal to the baseline reward of that arm. In other words:  $|q_{t_j^i} - b_i| \leq \varepsilon$  and this means that:

$$\mathbb{E}[S_j^i] = q_{t_j^i} \cdot r_i \in [(b_i - \varepsilon) \cdot r_i, (b_i + \varepsilon) \cdot r_i] \Rightarrow \mathbb{E}[S_j^i] \in [v_i, (1 + \varepsilon) \cdot v_i]$$

with probability 1. Note that the last derivation is because  $v_i - \varepsilon r_i \leq v_i$  and  $v_i + \varepsilon r_i \geq v_i + \varepsilon v_i r_i$ . As a result, by the linearity of expectation and using the definition of  $\hat{v}_i$ :

$$\mathbb{E}[\hat{v}_i] = \frac{\mathbb{E}[S_j^i]}{M} = \frac{\sum_{j \in [M]} q_{t_j^i} \cdot r_i}{M} = r_i \cdot \frac{\sum_{j \in [M]} q_{t_j^i}}{M} \Rightarrow \mathbb{E}[\hat{v}_i] \in [v_i, (1 + \varepsilon) \cdot v_i]$$

From Equation (11.4.4), we have that:

$$\begin{aligned} 2 \exp(-2M\delta^2) &\geq \Pr[\hat{v}_i - \mathbb{E}[\hat{v}_i] \geq \delta \text{ or } \hat{v}_i - \mathbb{E}[\hat{v}_i] \leq -\delta] \\ &\geq \Pr[\hat{v}_i \geq v_i + \varepsilon + \delta \text{ or } \hat{v}_i - \mathbb{E}[\hat{v}_i] \leq -\delta] \quad (\mathbb{E}[\hat{v}_i] \leq v_i \leq v_i + \varepsilon) \\ &\geq \Pr[\hat{v}_i \geq v_i + \varepsilon + \delta \text{ or } \hat{v}_i \leq r_i - \varepsilon - \delta] \quad (\mathbb{E}[\hat{v}_i] \geq r_i \cdot (1 - \varepsilon) \geq v_i - \varepsilon) \\ &= \Pr[|\hat{v}_i - v_i| \geq \delta + \varepsilon] \end{aligned}$$

■

It remains to show that using the estimators  $\hat{r}_i, \hat{v}_i$ , one can obtain a good estimator for the baseline rewards  $\hat{b}_i$  for each arm  $i \in [K]$ . Note that this is trickier than before because  $\hat{r}_i, \hat{v}_i$  are almost unbiased estimators and we are dealing with their product.

**Lemma 11.6.** *For the baseline reward estimators of each arm  $i$  in Line 18 of Algorithm 11.1 and any scalar*

$\delta > 0$ , it holds that:

$$\Pr \left[ \left| b_i - \hat{b}_i \right| \geq \delta \right] \leq 4 \exp(-2M \cdot (\varepsilon^2 - \varepsilon\delta)) + 4 \exp(-2M \cdot (\varepsilon - \delta)^2)$$

*Proof.* Fix an arm  $i \in [K]$  and let us use  $e_v$  and  $e_r$  to denote the following quantities:  $e_v = \hat{v}_i - v_i$  and  $e_r = \hat{r}_i - r_i$  respectively. Then, we have that:

$$\begin{aligned} \Pr \left[ \left| \frac{\hat{v}_i}{\hat{r}_i} - \frac{v_i}{r_i} \right| \geq \delta \right] &= \Pr \left[ \left| \frac{v_i + e_v}{r_i + e_r} - \frac{v_i}{r_i} \right| \geq \delta \right] = \Pr \left[ \left| \frac{r_i e_v - v_i e_r}{r_i(r_i + e_r)} \right| \geq \delta \right] \\ &\leq \Pr \left[ \left| \frac{e_v}{r_i + e_r} \right| + \left| b_i \frac{e_r}{r_i + e_r} \right| \geq \delta \right] \quad (\text{triangle inequality}) \\ &\leq \underbrace{\Pr \left[ \left| \frac{e_v}{r_i + e_r} \right| \geq \delta/2 \right]}_{Q_1} + \underbrace{\Pr \left[ b_i \cdot \left| \frac{e_r}{r_i + e_r} \right| \geq \delta/2 \right]}_{Q_2} \end{aligned} \tag{11.4.5}$$

To upper bound  $Q_1$  and  $Q_2$ , we condition on the following event:  $\mathcal{E}'_i = \{|e_r| \leq \delta\}$ . Note that the probability with which the complement  $\mathcal{E}_i$  happens is given by Lemma 11.4 and is:

$$\Pr[\mathcal{E}_i] \geq 2 \exp(-2M \cdot (\delta - \varepsilon)^2) \tag{11.4.6}$$

Rewriting  $Q_1$ :

$$Q_1 = \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot |r_i + e_r| \right] \leq \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot \left| |r_i| - |e_r| \right| \right] \tag{11.4.7}$$

Conditioning on  $\mathcal{E}'_i$  we get:

$$\begin{aligned} \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot \left| |r_i| - |e_r| \right| \mid \mathcal{E}'_i \right] &\leq \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot |r_i - \delta| \right] \\ &\leq 2 \exp \left( -2M \cdot \left( \frac{\delta}{2} \cdot |r_i - \delta| - \varepsilon \right)^2 \right) \quad (\text{Lemma 11.5}) \\ &\leq 2 \exp(-2M \cdot (\varepsilon^2 - \varepsilon\delta)) \end{aligned} \tag{11.4.8}$$

where the last inequality is due to the fact that  $|r_i - \delta| \leq 1$ . From the law of total probability:

$$\begin{aligned} Q_1 &= \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot |r_i + e_r| \mid \mathcal{E}'_i \right] \cdot \Pr [\mathcal{E}'_i] + \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot |r_i + e_r| \mid \mathcal{E}_i \right] \cdot \Pr [\mathcal{E}_i] \\ &\leq \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot ||r_i| - |e_r|| \mid \mathcal{E}'_i \right] \cdot \Pr [\mathcal{E}'_i] + \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot ||r_i| - |e_r|| \mid \mathcal{E}_i \right] \cdot \Pr [\mathcal{E}_i] \\ &\leq 2 \exp(M \cdot (\varepsilon^2 - \delta)) \cdot 1 + 1 \cdot 2 \exp(-2M \cdot (\delta - \varepsilon)^2) \end{aligned} \quad (11.4.9)$$

where the first inequality is due to Eq. (11.4.7) and the last one is due to Eqs. (11.4.6), (11.4.8).

We now turn our attention to  $Q_2$ :

$$Q_2 = \Pr \left[ |e_r| \geq \frac{\delta}{2} \cdot \frac{|r_i + e_r|}{b_i} \right] \leq \Pr \left[ |e_r| \geq \frac{\delta}{2} \cdot |r_i + e_r| \right]$$

where the inequality is due to the fact that  $b_i \leq 1$ . Using exactly the same reasoning as above, but now coupled with Lemma 11.4 instead of Lemma 11.5 we have that:

$$Q_2 \leq 2 \exp(M \cdot (\varepsilon^2 - \varepsilon\delta)) + 2 \exp(-2M \cdot (\delta - \varepsilon)^2)$$

Adding the two upper bounds from  $Q_1$  and  $Q_2$  to Equation (11.4.5) we get the stated result. ■

We are now ready to prove Theorem 11.1.

*Proof of Theorem 11.1.* For the rounds that pass while we are on lines 3 – 19 of Algorithm 11.1, we pick up regret at most 1 at each of them. Hence, the regret picked up in total equals to the number of rounds between these lines which are  $2c(\lambda) \cdot \log(1/\lambda\varepsilon) \cdot K \cdot M$ .

We next define events  $\mathcal{E}_{r,i} = \{|\hat{r}_i - r_i| \leq \delta\}$  and  $\mathcal{E}_{b,i} = \{|\hat{b}_i - b_i| \leq \delta\}$  for all  $i \in [K]$ . Then, conditional on the event  $\mathcal{E} = \{\cap_{i \in [K]} (\mathcal{E}_{r,i} \text{ and } \mathcal{E}_{v,i})\}$  and due to Lemma 11.2, the regret picked up for all the remaining rounds after feeding estimates  $\{(\hat{r}_i, \hat{b}_i)\}_{i \in [K]}$  to the dynamic programming procedure is at most  $\delta T$ . As a result, from the law of total probability, the regret for all  $T$  rounds:

$$\text{REGRET} \leq 2KM \frac{\log(\lambda\varepsilon)}{\log(1 - \lambda)} + \delta \cdot T \cdot \Pr [\mathcal{E}] + T \cdot \Pr [\mathcal{E}'] \leq 2KM \frac{\log(\lambda\varepsilon)}{\log(1 - \lambda)} + \delta \cdot T + T \cdot \Pr [\mathcal{E}'] \quad (11.4.10)$$

We next compute  $\Pr[\mathcal{E}']$ .

$$\begin{aligned}
\Pr[\mathcal{E}'] &= \Pr\left[\bigcup_{i \in [K]} (\mathcal{E}'_{r,i} \text{ or } \mathcal{E}'_{v,i})\right] && (\Pr[(A \cap B)''] = \Pr[A' \cup B'']) \\
&\leq \sum_{i \in [K]} (\Pr[\mathcal{E}'_{r,i}] + \Pr[\mathcal{E}'_{v,i}]) && (\text{union bound}) \\
&\leq 6K \exp(-2M(\delta - \varepsilon)^2) + 4 \exp(-2M \cdot (\varepsilon^2 - \varepsilon\delta)) && (\text{Lemmas 11.4, 11.6})
\end{aligned}$$

Tuning  $\delta = \varepsilon/4$  the latter becomes:  $\Pr[\mathcal{E}'] \leq 8K \exp(-M\varepsilon^2)$ . Tuning  $M = \ln(T)/\varepsilon^2$ :  $\Pr[\mathcal{E}'] \leq 8K/T$ . As a result, the regret from Equation (11.4.10) becomes:

$$\text{REGRET} \leq 2K \cdot \frac{\ln(T)}{\varepsilon^2} \cdot \frac{\log(\lambda\varepsilon)}{\log(1-\lambda)} + \frac{\varepsilon}{4} \cdot T + 8K$$

Tuning  $\varepsilon$  as stated gives us the result. ■

**SKETCH FOR THE UNKNOWN  $i_R$  CASE.** Note that two instances of arms  $(r_i, b_i)_{i \in [K]}$  and  $(cr_i, b_i/c)_{i \in [K]}$  are equivalent. So, we can always scale the  $b_i$ 's to make sure that we have a “replenishing” arm. The next part is to show how Algorithm 11.1 changes if we do not know which among the  $K$  arms is the replenishing arm. The only thing that changes is the way that we estimate  $\hat{r}_i$ 's (i.e., Lines 3-10). Instead of using  $i_R$  as the benchmark arm, we instead sample randomly an arm  $z$ . Then, after enough rounds, we can guarantee that with high probability  $\hat{r}_i \rightarrow \bar{b}r_i$ , where  $\bar{b} = \sum_j b_j/K$ . The second part of Algorithm 11.1 (i.e., Lines 11-18) remains the same. We can then guarantee that we have obtained estimates  $\hat{b}_i \rightarrow b_i/\bar{b}$  with high probability. Tuning again  $\delta, \varepsilon, M$  we obtain the *same order* regret guarantee. The details and the new algorithm can be found in Appendix G.

## 11.5 “STICKY” ARMS

In this section, we study the special case where  $\lambda = 1$ . As we show, for this special case, we can provide an algorithm that scales as  $\tilde{\mathcal{O}}(K^2\sqrt{T})$ , hence has much better regret guarantees than the ones we offer for the case of general  $\lambda$  in the previous section. We call this special case of the problem the case of “sticky” arms. The reason for this is that when  $\lambda = 1$ , after playing arm  $I_t$  then the current state becomes equal to arm  $I_t$ 's baseline reward:  $q_{t+1} = b_{I_t}$  (see Equation 11.2.1).

This has an important consequence; namely, that the optimal sequence of actions always alternates between 2 arms.

**Lemma 11.7.** *When  $\lambda = 1$ , then the optimal sequence of actions is a cycle of size 2.*

*Proof of Lemma 11.7.* Let us specify an ordering for the arms in terms of their in-the-void and baseline rewards. Indeed, assume that without loss of generality:

$$r_1 \geq r_2 \geq \cdots \geq r_K \quad \& \quad b_1 \geq b_2 \geq \cdots \geq b_K$$

We prove the lemma through contradiction. Assume that the optimal sequence includes cycle of arms  $j - k - m$ , i.e., every 3 rounds, it collects expected reward:

$$r_j b_m + r_k b_j + r_m b_k \tag{11.5.1}$$

We distinguish the following cases:

1. Arms (1, 2) are included in the cycle  $j - k - m$ , and without loss of generality, assume  $j = 1, m = 2$ . Then, from Equation (11.5.1), we have:  $r_1 b_2 + r_k b_1 + r_2 b_k \leq r_1 b_2 + r_2 b_1 + r_2 b_2$ , where the last inequality is due to the fact that  $r_k \geq r_2$  and  $b_k \leq b_2$ . Note that the right hand side of the above inequality is the expected reward obtained by cycle 2 – 1 – 2. This contradicts the assumption that  $j - k - m$  is the optimal cycle.
2. None of (1, 2) are included in the cycle  $j - k - m$ . Then from Equation (11.5.1) we have that  $r_j b_m + r_k b_j + r_m b_k \leq r_1 b_1 + r_2 b_1 + r_1 b_2$ , which is the expected reward obtained by cycle 1 – 2 – 1. This contradicts the assumption that  $j - k - m$  is the optimal cycle.

This proof directly generalizes to cycles of size greater than or equal to 4. ■

**Remark 11.1.** *Based on the proof, it may seem as if just playing arm 1 constantly is the optimal sequence. Note that this is only the case when the in-the-void and the baseline reward of this arm is the best among all others. In general, that is not the case as it may very well be that:*

$$r_{\sigma_1} \geq r_{\sigma_2} \geq \cdots \geq r_{\sigma_K} \quad \& \quad b_{\sigma'_1} \geq b_{\sigma'_2} \geq \cdots \geq b_{\sigma'_K}$$

Using the structure of the optimal sequence, we design our algorithm for the “sticky” arms case. We first discuss an example so as to give intuition regarding our algorithm.

Consider a setting where we have  $K = 3$  arms  $\{A, B, C\}$  with reward tuples  $(r_A, b_A) = (1, 0)$ ,  $(r_B, b_B) = (0, 1)$ ,  $(r_C, b_C) = \frac{1}{\sqrt{2}}(1 + x, 1 + x)$  where  $x \in [-\Delta, \Delta]$  for some scalar  $\Delta > 0$ . As we proved (Lemma 11.7), the optimal policy consists of at most 2 arms. It is easy to see that for our stated example, depending on whether  $x > 1$ , the optimal policy is either cycle  $AB$  or just arm  $C$ . So now the main challenge in designing a policy which naively switches between  $AB$  and  $C$  such as:

$$\underbrace{ABA - CCC}_{c} - \underbrace{ABA - CCC}_{c} - \underbrace{ABA - CCC}_{c} - \dots$$

is that the expected reward of cycle  $\mathcal{C}$  is:  $(r_B b_A + r_A b_B) + (r_C b_A + r_A b_C) + 2r_C b_C$ . If one were to define the *meta-arms* “ $AB$ ” and “ $CC$ ”, note that these satisfy  $r_B b_A + r_A b_B = 1$  and  $2r_C b_C = (1+x)^2$ . However,  $r_A b_C + r_C b_A = (1+x)/\sqrt{2}$ . This basically means if we are using an arm-elimination idea to distinguish the two meta-arms  $AB$  and  $CC$ , each transition (i.e., switch between  $A$  to  $C$  or  $C$  to  $A$ ) is going to cost us a constant regret. This would lead to regret  $\Omega(T^{2/3})$ .

To drop the exponent from  $2/3$  to  $1/2$ , we still use meta-arms  $AB$  and  $CC$  but we come up with a strategy to minimize switches. To do so, we define *batches* of meta-arms’ being played. In our running example, the batches would be defined as follows:

$$\underbrace{ABABA \cdots ABA - CCC \cdots C}_{\text{Batch 1}} - \underbrace{ABABA \cdots ABA - CCC \cdots C}_{\text{Batch 2}} - \dots$$

This idea can be generalized using intuition from batched bandits (Esfandiari et al., 2021) to more complex settings that contain more arms with arbitrary reward tuples  $(r, b)$ . We first create  $K(K+1)/2$  meta-arms. Our meta-arms consist of pairs  $\{(i, j) \mid i \leq j \in [K]\}$ . In our above example with  $K = 3$ , our 6 meta-arms would be  $\{(A, A), (B, B), (C, C), (A, B), (A, C), (B, C)\}$ . Note that based on Lemma 11.7, the optimal policy is playing one of these meta-arms repeatedly.

### Theorem 11.2: MAB with “Sticky” Arms Regret

Tuning  $B = 2 \ln T$ , Algorithm 11.2 incurs regret  $\text{REGRET}(T) = \tilde{\mathcal{O}}(K^2 \sqrt{T})$ .

*Proof.* We first list a property that is very useful for our proof. Note that the average of rewards

---

**Algorithm 11.2:** Batched Bandit Algorithm on Meta-Arms for the “Sticky” Arms Case

---

1 **Input.** Number of batches  $B$ ,  $K$  arms, time horizon  $T$ .  
 2 Set  $w = T^{1/B}$  and generate  $M = K(K + 1)/2$  meta-arms  $(a_i, a_j)$ , with  $i \leq j, (i, j) \in [K]^2$ .  
 3 Set active meta-arms  $\mathcal{A} = \{(i, j) \mid i \leq j, i, j \in [K]\}$ . //  $|\mathcal{A}| = M$  initially  
 4 For  $i \leq j \in [K]$  initialize estimated means  $\hat{\mu}_{(i,j)} = 0$ .  
 5 **for** batch  $\beta = 1$  to  $B - 1$  **do**  
 6     **if**  $\lfloor w^\beta \rfloor \cdot |\mathcal{A}| > \text{remaining rounds}$  **then**  
 7         Break  
 8     **for** each active arm  $(i, j)$  in  $\mathcal{A}$  **do**  
 9         Play  $(i, j)$  for  $U_\beta = \lfloor w^\beta \rfloor$  times. // contains  $2U_\beta + 1$  actions  
 10         Drop the first reward observation that has expectation  $r_i q_0$ , where  $q_0 = 1$   
 11         Pair the other observations into  $U_\beta$  groups of size 2.  
 12         Update  $\hat{\mu}_{(i,j)}$  using these new  $U_\beta$  observations (sample mean).  
 13         Update the number of observations of all existing meta-arms according to  
 14              $c_\beta = \sum_{l=1}^{\beta} U_l$ .  
 15     **for** each active arm  $(i, j)$  in  $\mathcal{A}$  **do**  
 16         Eliminate this arm if it is sub-optimal, i.e., remove it from  $\mathcal{A}$  if it satisfies

$$\hat{\mu}_{(i,j)} < \max_{(u,v) \in \mathcal{A}} \hat{\mu}_{(u,v)} - \sqrt{\frac{2 \ln(2K^2TB)}{c_\beta}}$$

16 In the last batch, play the optimal remaining meta-arm, i.e., the one with the highest  $\hat{\mu}_{(i,j)}$  for  $(i, j) \in \mathcal{A}$ .

---

observed in each group of size 2 satisfies  $\mathbb{E}[(r_t + r_{t+1})/2] = (r_i b_j + r_j b_i)/2$ , since we have i.i.d. and  $\sigma/\sqrt{2}$ -subgaussian observations.

Let  $(i^*, j^*)$  be the optimal meta-arm. Let  $\Delta_{(i,j)}$  be the gap of meta-arm  $(i, j)$ , defined as:

$$\Delta_{(i,j)} = \frac{r_{i^*} b_{j^*} + r_{j^*} b_{i^*} - r_i b_j - r_j b_i}{2}.$$

Then, the regret incurred throughout  $T$  rounds can be written as:

$$\begin{aligned} \text{REGRET}(T) &= \sum_{t=1}^T \left( \frac{r_{i^*} b_{j^*} + r_{j^*} b_{i^*}}{2} - r_{I_t} b_{I_{t-1}} \right) \\ &\leq \sum_{1 \leq i \leq j \leq K} \Delta_{(i,j)} N_{(i,j)} + \sum_{t \in [T]} \mathbb{I}[\text{transition between two meta-arms happens at } t] \end{aligned}$$

where  $N_{(i,j)}$  is the number of pulls of meta-arm  $(i, j)$  during  $T$  rounds. Since  $r, b \in [0, 1]$ , then the second term in the above is upper bounded by  $BK(K + 1)/2$  as in each batch the transition happens only between active arms. As a result, the regret is upper bounded by:

$$\text{REGRET}(T) \leq 2 \sum_{1 \leq i \leq j \leq K} \Delta_{(i,j)} N_{(i,j)} + \frac{BK(K + 1)}{2}. \quad (11.5.2)$$

Next, we bound  $N_{(i,j)}$  using variations of standard arm-elimination techniques. We call the estimation for a meta-arm  $(i, j)$  at the end of batch  $\beta$ ,  $\delta$ -correct, if the true mean of that meta-arm is within  $\sqrt{2 \ln(1/\delta)/c_\beta}$  of estimated value, i.e.,

$$\left| \hat{\mu}_{(i,j)} - \frac{r_i b_j + r_j b_i}{2} \right| \leq \sqrt{\frac{2 \ln(1/\delta)}{c_\beta}}.$$

Now as  $\hat{\mu}_{(i,j)}$  contains of  $c_\beta$  i.i.d. samples with mean  $\mu_{(i,j)} = (r_i b_j + r_j b_i)/2$  (standard deviation at most 1), Hoeffding inequality implies that each active meta-arm is  $\delta$ -correct with probability at least  $1 - \delta$ . Since we have  $K(K + 1)/2$  meta-arms and  $B$  batches, then selecting  $\delta = 1/(2K^2 BT)$  and a union bound implies that with probability  $1 - 1/T$ , all active meta-arms are  $\delta$  valid in all batches.

Now if this happens, it basically means that all active arms  $(i, j)$  at the end of every batch satisfy

$$\left| \widehat{\mu}_{(i,j)} - \frac{r_i b_j + r_j b_i}{2} \right| \leq \sqrt{\frac{2 \ln(2K^2 BT)}{c_\beta}}.$$

This means that the best meta-arm  $(i^*, j^*)$  is never eliminated. We can now derive an upper bound on the number of pulls of each of these sub-optimal  $(i, j)$  meta-arms as follows. Let  $\beta + 1$  be the last batch in which arm  $(i, j)$  was active. Since this arm was not eliminated at batch  $\beta$ , we have

$$\Delta_{(i,j)} \leq 2 \sqrt{\frac{2 \ln(2K^2 BT)}{c_\beta}},$$

which after re-arrangement means that  $c_\beta \leq 8 \ln(2K^2 BT) \Delta_{(i,j)}^{-2}$ . Note that this also means that

$$N_{(i,j)} \leq \max\{T, c_{\beta+1}\} = \max\{T, w + wc_\beta\} = \max\{T, w + 8w \ln(2K^2 BT) \Delta_{(i,j)}^{-2}\}.$$

Putting everything together and replacing  $w = T^{1/B}$ , from Equation (11.5.2) we have

$$\begin{aligned} & \text{REGRET}(T) \\ & \leq 2 + 2 \sum_{1 \leq i \leq j \leq K} \max\{T \Delta_{(i,j)}, T^{1/B} \Delta_{(i,j)} + 8T^{1/B} \ln(2K^2 BT) \Delta_{(i,j)}^{-1}\} + \frac{K(K+1)B}{2}, \end{aligned}$$

Tuning  $B = 2 \ln(T)$  then we have

$$\begin{aligned} \text{REGRET}(T) & \leq 2 + 36 \ln(2K^2 BT) \sum_{1 \leq i \leq j \leq K} \max\{T \Delta_{(i,j)}, \Delta_{(i,j)}^{-1}\} + K(K+1) \ln(T) \\ & \leq 2 + 36K^2 \ln(2K^2 BT) \sqrt{T} + K(K+1) \ln(T) = \mathcal{O}\left(K^2 \sqrt{T} \ln(2K^2 T)\right) \end{aligned}$$

■

## 11.6 DISCUSSION AND OPEN QUESTIONS

In this chapter, we studied a bandit learning setting which accounts for long-term effects and whose main application is online advertising. At the heart of our construction is the notion of the state

and the parameter  $\lambda$ , which denotes how fast the system's long-term effects evolve.

We think there are two limitations regarding this chapter. First, the different algorithms that we have provided are for general  $\lambda$ 's and  $\lambda = 1$ . There is another special case which is of great interest; the case of  $\lambda$  being infinitesimally small. This case corresponds to settings where the state evolution happens very slowly and no single arm has a driving effect on it. Second, it is unclear whether our bounds are optimal. That said, we conjecture that they are, as in the worst-case (i.e., general  $\lambda$ ) we could not upper bound the size of the optimal cycle. Both of the aforementioned limitations are excellent avenues for future research. Finally, studying a "contextual" version of bandit learning with long-term effects is another very intriguing question.

## **Part VI**

# **Conclusion and Open Questions**

# 12

## Closing Thoughts

In this dissertation we have laid the foundations of incentive-aware ML for decision making in a variety of contexts. Below we outline the biggest and boldest open questions of the field, not tied to any particular result among the ones we presented.

### INCENTIVE-COMPATIBLE ML WITHOUT ACCURACY DEGRADATION

In Part I, we began by analyzing settings where the strongest incentive guarantees were obtainable in offline and online learning settings. Although we saw that any incentive-compatible ML algorithm *has* to suffer a degradation of accuracy in the case of linear regression with the dependent variable being the manipulable quantity, we presented an algorithm for online learning in the full information case where incentive-compatibility and optimal accuracy guarantees are possible! *Is*

*there an inherent problem property that allows incentive compatibility to not cause accuracy degradation?*

Definitely (and contrary to intuition), the property is not related to the dynamic nature of the setting. One conjecture is that it maybe is related to *when* the strategizing happens. Note that in our model for linear regression, the agent strategizing was happening at *train time*. However, for the online learning setting, the training and test time are interleaved (since it is an online setting).

## TOWARDS BETTER MODELS FOR AGENT BEHAVIOR

In incentive-aware ML settings, the agents' behavior is of utmost importance; indeed, any guarantees that we are able to prove are based on appropriately modeling human behavior. It seems that when it comes to assumptions regarding the agent behavior there are two opposing ends of a line. On the one end, we have the standard viewpoint of adversarial ML where we assume that agents are worst-case adversaries that take actions so as to sabotage the ML algorithms. On the other opposing end, we have the Game Theory viewpoint where agents are perfect optimizers of some very well defined underlying utility functions.

I believe that the truth lies somewhere in the middle. In fact, I believe that agents are a mix between being fully adversarial and fully strategic. *What properties or settings make an agent fully adversarial or fully strategic? And do agents even strategize consciously or is it almost like a reflex in some settings?* For example, think about the way that all of us use social media apps. The content that we consume today in an app is inherently linked with the content that the app will present to us in the future. Anecdotal evidence suggests that people do engage in some sort of strategizing in order to curate the content that they will be exposed to in the future. The conjecture that people strategize subconsciously gives further credence to the idea that they are not perfect utility maximizers. Recommendation systems are an excellent application domain for building experiments to test these hypotheses, due to their widespread adoption.

## GAMING VS IMPROVEMENT

The majority of this dissertation focused on agent actions that constituted “gaming”, while Chapter 9 focused on actions that help achieve honest outcome improvement. So *how do we distinguish between the two?* The literature so far either assumes that there is a set of actions that strictly do not

help outcome improvement, and another set of actions that strictly do. But even this causal view of the world misses two important points.

The first point can be summarized in the question: *are there actions that up to an extent lead to improvement and after some threshold they lead to gaming?* Think for example about a student who repeatedly takes the GRE test in order to improve their scores. One could argue that the first two times that one takes the test can really help the student with understanding the material better, but after the second try, any further attempt is just memorizing test tricks. Maybe there is some form of diminishing returns for taking a test like GRE or generally optimizing each independent feature.

As for the second point, *how do we even define “gaming” or “improvement”?* In the paragraph above, I have used the simplification that any time that an action helps change the outcome to something higher, then it is improvement. But this is a very simplistic, mathematical, technical view of a deeply societal issue. For example, it is unclear if there indeed exists a hidden, global function that can explain the ground truth outcome for a student in the college admissions case. How can we claim that applications correlate directly with the actual ability of students to be academically successful without consulting with sociologists?

### “INTERPRETABILITY” VS INCENTIVES

Based on the models we have analyzed in this dissertation, one could argue that all of our troubles with strategizing would be solved if we made all of ML models *obscure*. Indeed, in most of our models in this dissertation (apart from Chapter 9) we have imposed the fundamental assumption that the ML model gets revealed (or somehow leaked) to the agents, and they afterwards know exactly how to manipulate their inputs. So keeping everything secret would seemingly solve all the incentives issues.

That said, when ML models are deployed for consequential decision-making (as has been predominantly the case in this dissertation) people should be able to probe the models, trust them, and understand them. This is because oftentimes people want to honestly improve their standing with respect to the algorithms (i.e., recourse) rather than only gaming them. I believe that different notions of “interpretability” can help alleviate the aforementioned tension. In a most basic form, we should strive for “interpretable” algorithms that incentivize recourse but do not reveal enough

information to the agents so that they can game their inputs or strategize. *What is the set of properties for an “interpretable” algorithm that incentivizes certain actions from individuals but does not leak the full details of the model, thus making it prone to strategizing?* This is a fundamental question both from the side of institutions who need to have their models trusted and robust to strategizing but also individuals and society who strive for social welfare.

One potential way to address this issue is using cryptographic primitives (e.g., zero-knowledge proofs). Cryptographic primitives could allow the organization to reveal some information regarding their model (or to certify that some computations of their model is as they are claiming them to be) without leaking the whole model to the individual. So the big question here becomes *what are the properties of well-known cryptographic primitives that would offer enough interpretability for recourse to the individuals?* Apart from the theoretical underpinnings of this big question, even if we identify these properties it still remains to see whether individuals would really understand this use of cryptography. Another closely connected question here is whether there could be a “knob” for society to tune whether we (and by “we” I mean not only individuals but the Law as well) would like more recourse from a model or more cryptographic obscurity for the sake of the institution’s protection.

As is the case with most questions related to socially aware ML, Computer Scientists and folks from technical/mathematical backgrounds should be wary of not reinventing the wheel; chances are “a Law or a Policy scholar had thought about versions of our problems back in the eighties”\*. Not surprisingly, what I am advocating for above (i.e., separating what the individuals know about the model and what the actual model is although still incentivizing individuals to take a specific set of actions) is known in the Law as *acoustic separation* ([Dan-Cohen, 1984](#)).<sup>†</sup> [Dan-Cohen \(1984\)](#) built an experiment where there would be a separation between people to two subpopulations; the first one would be the population of legal experts and the second one would be the population of “ordinary” people. “Acoustic separation” corresponds to the fact that ordinary people would only see the conduct rules (which hopefully incentivize some sort of behavior), while legal experts would get to see the full decision rules. [Dan-Cohen \(1984\)](#)’s experiment introduced the question of how

---

\*An actual quote from Yonadav Shavit during our FAccT21 tutorial.

<sup>†</sup>I learned about this through personal communication with Karen Levy.

should decision and conduct rules be built if we want them to be addressed to different audiences, but still incentivize a certain behavior from both. This is fundamentally the same question that we need to address next in the area of incentive-aware ML for decision making.

# A

## Appendix for Chapter 4

This section includes the supplementary material for Chapter 4.

### A.1 GRADIENT DESCENT VIOLATES INCENTIVE COMPATIBILITY

In gradient descent the loss function that we are trying to optimize is  $(r_t - \sum_{i \in [K]} \pi_{i,t} p_{i,t})^2$ . Assume that for all the experts  $j \neq i$ ,  $b_{j,t} = p_{j,t} = 0$ . Then, from the perspective of expert  $i$  and according to their belief  $b_{i,t}$  their expected weight at the next round is

$$\mathbb{E}_{r_t \sim \text{Bern}(b_{i,t})} [\pi_{i,t+1}] = b_{i,t} \cdot \underbrace{\frac{\pi_{i,t} + 2\eta p_{i,t}(1 - \pi_{i,t} p_{i,t})}{1 + 2\eta p_{i,t}(1 - \pi_{i,t} p_{i,t})}}_{Q_1} + (1 - b_{i,t}) \cdot \underbrace{\frac{\pi_{i,t} - 2\eta p_{i,t}^2 \pi_{i,t}}{1 - 2\eta p_{i,t}^2 \pi_{i,t}}}_{Q_2}.$$

We begin with a specific case:  $K = 10$ ,  $\pi_{i,t} = 0.1$ ,  $b_{i,t} = 0.6$ . Then, for any  $\eta \geq 2.85 \cdot 10^{-15}$  reporting  $p_{i,t} = 0.61$  is a beneficial manipulation for the expert. To construct similar counterexamples for any  $\eta$ , one needs to focus on cases where  $\pi_{i,t} \rightarrow 0$  (which can be achieved by, for instance, allowing the number of experts to grow large), hence  $Q_2$  is almost 0 and  $Q_1$  ends up thus being maximized when  $p_{i,t}$  is maximum (i.e., for  $p_{i,t} \rightarrow 1$ ).

## A.2 SUPPLEMENTARY MATERIAL FOR SECTIONS 4.3–4.4

### A.2.1 TECHNICAL LEMMA

**Lemma A.1.** *For all  $x \leq 1/2$ , it holds that:  $\ln(1 - x) \geq -x - x^2$ .*

*Proof.* Let function  $f(x)$ ,  $x \leq 1/2$  be defined as  $f(x) = \ln(1 - x) + x + x^2$ . It suffices to show that  $f(x) \geq 0$  for the domain of interest. Taking the first derivative we get

$$f'(x) = \frac{-x(2x - 1)}{1 - x}.$$

For  $x \leq 1/2$ ,  $f'(x) = 0$  for  $x = 0$  and  $x = 1/2$ . Now, since  $f'(x) \leq 0$ ,  $x \leq 0$  and  $f'(x) \geq 0$ ,  $0 \leq x \leq 1/2$  we get that  $f(x)$  is decreasing for  $x \in (-\infty, 0]$  and increasing for  $x \in [0, 1/2]$ . As such, it presents a minimum at  $x = 0$ , and for  $x \leq 1/2$ ,  $f(x) \geq f(0) = \ln(1) + 0 + 0 = 0$ . Hence,  $\ln(1 - x) \geq -x - x^2$ . ■

### A.2.2 REGRET OF WSU FOR UNKNOWN TIME HORIZON $T$

To provide an anytime variant of WSU, we use a standard doubling trick (Auer et al., 2002b). We maintain an estimated upper bound on the time horizon  $T$ , denoted  $n$ , starting with  $n = 1$ . For all  $t \in (n/2, n]$ , we run WSU using  $\eta = \eta_n = \sqrt{\ln(K)/n}$ . If at any round  $t'$  we have that  $t' > n$ , then we double our estimated horizon upper bound to  $2n$  (changing  $\eta$  accordingly) and restart WSU by initializing all weights to  $1/K$ . As we prove, this process increases the regret only by constants.

**Lemma A.2.** *For an a-priori unknown time horizon  $T$ , WSU with a doubling trick is incentive-compatible and incurs regret  $R \leq \frac{2\sqrt{2}}{\sqrt{2}-1} \sqrt{T \ln K}$ .*

*Proof.* Using the doubling trick, the time horizon  $T$  can be divided into phases during which  $n$ , and hence also  $\eta$ , remain constant. Because of this, from the perspective of an expert  $i$ , it does not matter in which phase the algorithm is currently at: their probability at the next round is computed as  $\pi_{i,t+1} = \eta_n \Gamma_i^{\text{WSWM}}(\mathbf{p}_t, \boldsymbol{\pi}_t, r_t) + (1 - \eta_n) \pi_{i,t}$ , hence it still is a convex combination of a WSM payment and  $\pi_{i,t}$ , which cannot be influenced by  $i$ 's report at round  $t$ . Since the algorithm every time restarts (i.e., experts' weights are re-initialized to  $1/K$ ) using the new  $\eta_n$  for all the rounds, this ends up being equivalent to having a constant  $\eta$  throughout  $T$  rounds in terms of incentives.

Since the length of each phase,  $n$ , is doubled at the end of each phase, the number of these phases is at most  $\lceil \log T \rceil$ . Also, the actual regret throughout the  $T$  rounds is upper-bounded by the sum of the regret of each phase. Hence, using Theorem 4.1 we have that:

$$\begin{aligned} R &\leq \sum_{n=0}^{\lfloor \log T \rfloor} 2\sqrt{2^n \ln K} \leq \left(2\sqrt{\ln K}\right) \sum_{n=0}^{\lfloor \log T \rfloor} (\sqrt{2})^n \\ &= \left(2\sqrt{\ln K}\right) \frac{1 - \sqrt{2}^{\lfloor \log T \rfloor + 1}}{1 - \sqrt{2}} \\ &= \left(2\sqrt{\ln K}\right) \frac{2^{\frac{1}{2}\lfloor \log T \rfloor} \cdot \sqrt{2} - 1}{\sqrt{2} - 1} \\ &\leq \left(2\sqrt{\ln K}\right) \frac{2^{\lfloor \log T^{1/2} \rfloor} \cdot \sqrt{2}}{\sqrt{2} - 1} \\ &= \left(2\sqrt{2}\sqrt{\ln K}\right) \frac{T^{1/2}}{\sqrt{2} - 1} = \frac{\left(2\sqrt{2}\sqrt{T \ln K}\right)}{\sqrt{2} - 1} \end{aligned}$$

where the first equality comes from the definition of a geometric series with rate  $\sqrt{2}$ . ■

### A.2.3 REGRET OF WSU-UX FOR UNKNOWN TIME HORIZON $T$

Similarly to Appendix A.2.2, here we use the doubling trick (Auer et al., 2002b) to achieve regret guarantees for WSU-UX for the case of an unknown horizon  $T$ . Formally, we prove the following.

**Lemma A.3.** *For an a-priori unknown time horizon  $T$ , WSU-UX with a doubling trick is incentive-compatible and incurs regret  $R \leq \frac{8}{2^{2/3}-1} T^{2/3} (K \ln K)^{1/3}$ .*

*Proof.* Algorithm WSU-UX is divided into phases during which  $n$  and  $\eta$  remain constant. This coupled with the fact that at every phase the algorithm is restarted and the experts' weights are re-

initialized to  $1/K$  (i.e., hence all previous weights have been updated with the same  $\eta$ ) means that from the perspective of an expert, the incentives structure remains the same. As a result, WSU-UX with a doubling trick is incentive-compatible.

The number of the algorithm's phases is at most  $\lfloor \log T \rfloor$ . The actual regret through the  $T$  rounds is upper bounded by the sum of the regret of each phase. So, from Theorem 4.2 we obtain that:

$$\begin{aligned} R &\leq \sum_{n=0}^{\lfloor \log T \rfloor} 2 \cdot 4^{2/3} \cdot (K \ln K)^{1/3} \cdot (2^n)^{2/3} = 2 \cdot 4^{2/3} \cdot (K \ln K)^{1/3} \sum_{n=0}^{\lfloor \log T \rfloor} \left(2^{2/3}\right)^n \\ &= 2 \cdot 4^{2/3} \cdot (K \ln K)^{1/3} \frac{1 - (2^{2/3})^{\lfloor \log T \rfloor + 1}}{1 - 2^{2/3}} \leq 2 \cdot 2^{2/3} \cdot 4^{2/3} \cdot (K \ln K)^{1/3} \frac{(2^{2/3})^{\lfloor \log T \rfloor}}{2^{2/3} - 1} \\ &= 2 \cdot 2^{2/3} \cdot 4^{2/3} \cdot (K \ln K)^{1/3} \frac{(2)^{\frac{2}{3} \lfloor \log T \rfloor}}{2^{2/3} - 1} = \frac{8}{2^{2/3} - 1} (K \ln K)^{1/3} T^{2/3} \end{aligned}$$

■

### A.3 SUPPLEMENTARY MATERIAL FORWARD-LOOKING EXPERTS (SECTION 4.5)

We begin with a definition of incentive compatibility when experts may look more than one round into the future. This stronger version of incentive compatibility requires that for any round  $t$  and future round  $t^f > t$ , experts maximize their expected weight at round  $t^f$  by truthfully reporting their beliefs at all rounds between  $t$  and  $t^f$ .

**Definition A.1** (Incentive Compatibility for Forward-Looking Experts). *An online learning algorithm is incentive-compatible for forward-looking experts if for every round  $t \in [T]$  and every future round  $t^f > t$ , every expert  $i$  with beliefs  $(b_{i,t'})_{t \leq t' < t^f}$ , and every set of reports of expert  $i$ ,  $(p_{i,t'})_{t \leq t' < t^f}$ , reports of the other experts  $(\mathbf{p}_{-i,t'})_{t \leq t' < t^f}$ , and every history of reports  $(\mathbf{p}_{t''})_{t'' < t}$  and outcomes  $(r_{t''})_{t'' < t}$ ,*

$$\begin{aligned} &\mathbb{E}_{(r_{t'} \sim \text{Bern}(b_{i,t'}))_{t \leq t' < t^f}} [\pi_{i,t^f} | (b_{i,t'})_{t \leq t' < t^f}, (\mathbf{p}_{-i,t'})_{t \leq t' < t^f}, (\mathbf{p}_{t''})_{t'' < t}, (r_{t''})_{t'' < t}] \\ &\geq \mathbb{E}_{(r_{t'} \sim \text{Bern}(b_{i,t'}))_{t \leq t' < t^f}} [\pi_{i,t^f} | (p_{i,t'})_{t \leq t' < t^f}, (\mathbf{p}_{-i,t'})_{t \leq t' < t^f}, (\mathbf{p}_{t''})_{t'' < t}, (r_{t''})_{t'' < t}]. \end{aligned}$$

WSU and WSU-UX do not satisfy incentive compatibility for forward-looking experts. We present an example for WSU, but note that adding a small amount of uniform exploration will still yield a violation. Observe also that the incentives to deviate in the following example are very small.

It is an open problem whether WSU can sometimes produce larger incentives to misreport, or, conversely, whether it satisfies some notion of  $\epsilon$ -incentive compatibility.

**Theorem A.1**

WSU is not incentive-compatible for forward-looking experts.

*Proof.* Let  $K = 2$ ,  $T = 3$ , and  $b_{1,1} = 0.7$ ,  $b_{1,2} = 0.6$ ,  $b_{2,1} = 0.4$ , and  $b_{2,2} = 0$ . If both experts report truthfully at both rounds, it can be checked that the expected weight of expert 1 at round 3 is  $\mathbb{E}_{r_1 \sim \text{Bern}(b_{1,1}), r_2 \sim \text{Bern}(b_{1,2})}[\pi_{1,3}] = 0.5 + 0.1125\eta - 0.00188325\eta^2$ . But if expert one instead reports  $p_{1,1} = 0.699$ , then his expected weight at round 3 is  $\mathbb{E}_{r_1 \sim \text{Bern}(b_{1,1}), r_2 \sim \text{Bern}(b_{1,2})}[\pi_{1,3}] = 0.5 + 0.112499944\eta - 0.0018719238\eta^3$ . It is easy to check that the latter is larger than the former for all  $\eta > 0.0703$ .

For ease of presentation we do not present a possible manipulation for smaller values of  $\eta$ , but note that such manipulations can be obtained by considering  $0.699 < p_{1,1} < 0.7$ . ■

For completeness, we include here some discussion as to the distinction between our ELF-X algorithm and the ELF algorithm of [Witkowski et al. \(2018\)](#), who designed ELF for selecting the winner of a forecasting competition. ELF works similarly to ELF-X as defined in Section 4.5, except that the “winner”  $x_\tau$  of each round  $\tau \in [t]$  is chosen with probability  $\frac{1}{K} \left(1 - \ell_{i,t'} + \frac{1}{K-1} \sum_{j \in [K] \setminus \{i\}} \ell_{j,t'}\right)$ . Unfortunately, direct application of ELF in the online learning settings we are considering in this paper yields an algorithm with linear regret in the worst case. In particular, when there are two experts and the reports of each expert are always either 0 or 1, ELF reduces to the Follow-the-Leader algorithm that, at every round, selects the expert with the lowest cumulative loss. It is well known that Follow-the-Leader has linear regret even under this restriction. ELF-X avoids this problem by adding additional randomness into the selection of each round’s winner.

We now provide a sketch proof of Theorem 4.3, that ELF-X is incentive-compatible for forward-looking experts. For details, we refer the reader to [Witkowski et al. \(2018\)](#).

*Proof Sketch of Theorem 4.3.* Incentive compatibility rests on the fact that each expert maximizes his (subjective) probability of being selected as the event winner of any round  $\tau$  by reporting  $p_{i,\tau} = b_{i,\tau}$ . This is because an expert’s probability of being selected as the winner of event  $\tau$  is exactly their payment from participating in a Weighted Score Wagering Mechanism where every expert has

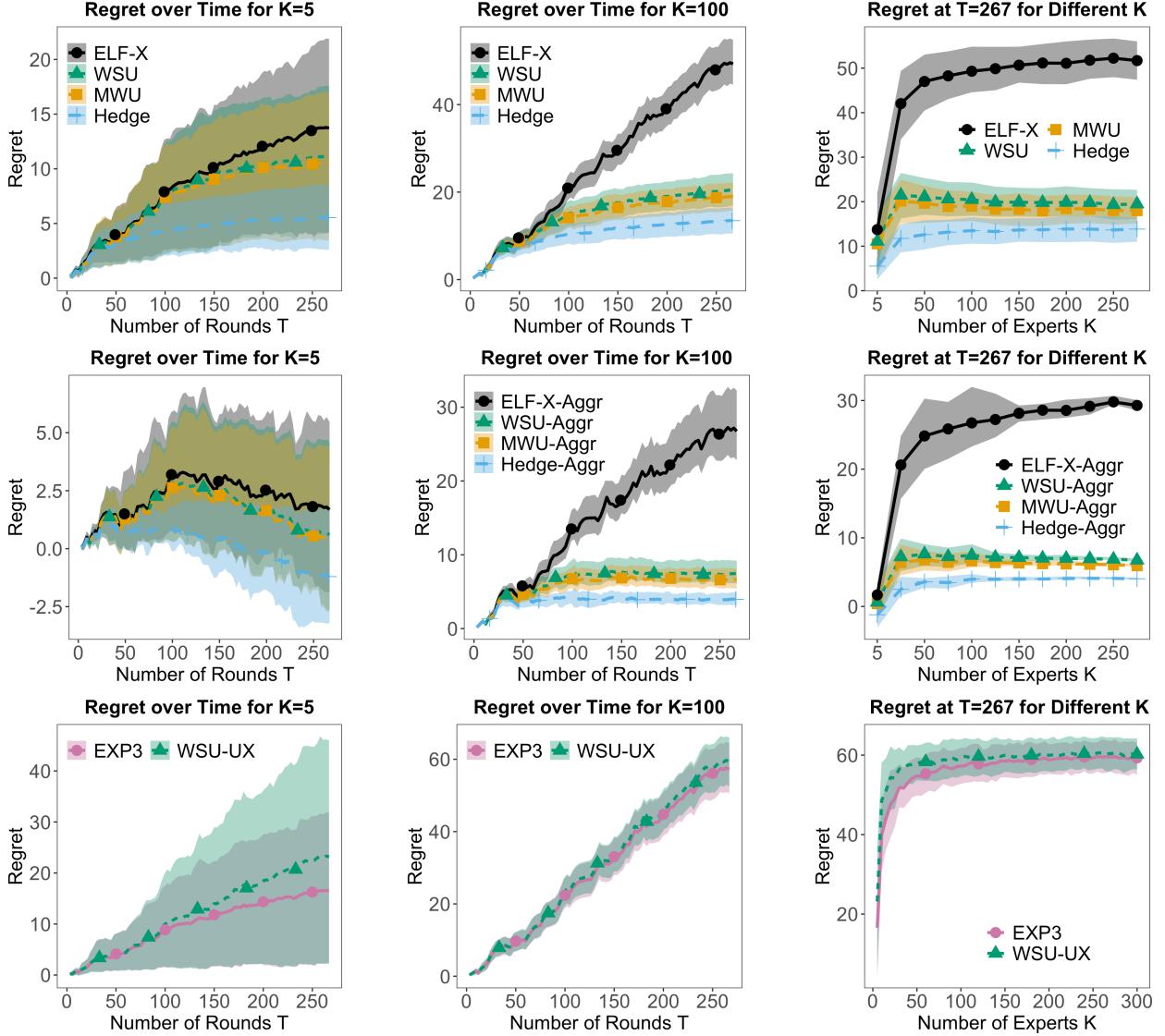


Figure A.1: Experiments on the 2019–2020 FiveThirtyEight NFL dataset. Top: Full-information setting with  $\bar{p}_t$  the prediction of a single expert chosen according to  $\pi_t$ . Middle: Full-information setting with  $\bar{p}_t = \sum_{i \in [K]} \pi_{i,t} p_{i,t}$ . Bottom: Partial information setting.

wager  $1/K$ . Further, it is easy to check that an expert  $i$  minimizes the probability of any other expert  $j$  being selected as winner of round  $\tau$  (according to  $i$ 's belief  $b_{i,\tau}$ ).

Fix the winners on all rounds other than  $\tau$ . Because the winner at each round is chosen independently of all other rounds, it is a dominant strategy for each expert to report his belief  $b_{i,\tau}$ . Incentive compatibility follows by applying this argument to all rounds  $\tau$ . ■

## A.4 ADDITIONAL EXPERIMENTS (SUPPLEMENTARY FOR SECTION 4.6).

### A.4.1 FIVE THIRTY EIGHT NFL 2019–2020 DATASET

In this subsection we present in Figure A.1 the results of our experiments for the 2019–2020 FiveThirtyEight NFL dataset. The findings and conclusions are almost identical to those drawn using the 2018–2019 FiveThirtyEight NFL dataset found in Section 4.6.

### A.4.2 MONTE CARLO SIMULATIONS WITH LARGE HORIZON $T$

In this subsection, we present our results for Monte Carlo simulations for larger horizons in Figure A.2. We simulated the following setup:  $K = 50$ ,  $T = 2500$  and we repeated the simulations for 50 repetitions. The lines correspond to average regret (across all repetitions), and the error bands in Figure A.2 correspond to the 20th and the 80th percentiles.

The realized outcomes are sampled as follows: for rounds  $0 \leq t \leq T/2$ ,  $r_t \sim \text{Bern}(0.4)$ , and for rounds  $T/2 + 1 \leq t \leq T$ ,  $r_t \sim \text{Bern}(0.6)$ . The  $K$  experts are randomly partitioned into three equal-sized groups sampling their beliefs from three different distributions: for experts in the first group we draw  $b_{i,t} \sim \text{Unif}[0, 0.7]$  for all rounds  $t$ , for the second group  $b_{i,t} \sim \text{Unif}[0.3, 1]$  for all  $t$ , and for the third group  $b_{i,t} \sim \text{Unif}[0, 1]$  for all  $t$ . As a result, in expectation, experts from the first group perform best for the first  $T/2$  rounds, the second group performs best for the next  $T/2$  rounds, while the third group performs best when all  $T$  rounds are considered.

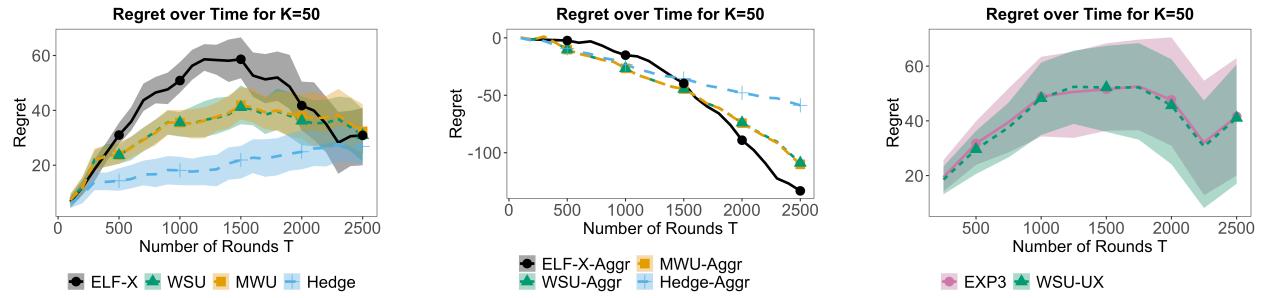


Figure A.2: Simulation results for  $K = 50$  experts. Left: Full-information setting with  $\bar{p}_t$  the prediction of a single expert chosen according to  $\pi_t$ . Middle: Full-information setting with  $\bar{p}_t = \sum_{i \in [K]} \pi_{i,t} p_{i,t}$ . Right: Partial information setting.

Due to the way we constructed the simulation parameters, examining the performance of the al-

gorithms for rounds between  $[0, T/2]$  provides intuition about their performance for settings where the experts' performance is relatively stable over time. However, their performance for rounds between  $[T/2, T]$  provides intuition for settings where the best expert is shifting over time. As a result, for rounds between  $[0, T/2]$  our findings are similar to the findings of our experiments on the FiveThirtyEight NFL datasets: ELF performs worse than WSU (which performs identically to MWU) and worse than Hedge, and WSU-UX performs almost identically to EXP3 despite our weaker theoretical bound.

Interestingly, for rounds between  $[T/2, T]$  we find that ELF briefly performs better than WSU, MWU and Hedge. We conjecture this is because ELF in the first  $T/2$  rounds takes longer than MWU, WSU and Hedge to converge to experts in the first group. Because these experts are no longer optimal through the  $T$  rounds, ELF has an advantage over other algorithms.

Lastly, we note that the regret performance of the aggregating variants of all algorithms is always negative due to the fact that the expectation over all experts is very close to issuing the optimal prediction for all rounds. As a result, a prediction that takes into account all of their predictions in a weighted fashion performs much better than the prediction of any fixed expert in hindsight. We also note that the fact that Hedge is performing worse than the other algorithms is not contradicting the theoretical results, which are only stated in terms of *worst case* upper bounds. Finally, we see that ELF-X-Aggr performs better than *all* algorithms in this setting. Explaining this phenomenon theoretically even for particular settings is a question of great interest.

# B

## Appendix for Chapter 5

### B.1 APPENDIX FOR SECTION 5.3

#### B.1.1 PURELY ADVERSARIAL AND COOPERATIVE STACKELBERG GAMES

Despite the worst-case incompatibility results that we have shown for the notions of external and Stackelberg regret, there are families of repeated games for which there is a clear *hierarchy* between the two. In this subsection, we study two of the most important ones; the family of *Purely Adversarial*, and the family of *Purely Cooperative Stackelberg Games*.

**Definition B.1** (Purely Adversarial Stackelberg Game (PASGs)). *We call a Stackelberg Game Purely Adversarial, if for all actions  $\alpha' \in \mathcal{A}$  for the loss of the learner it holds that:  $\ell(\alpha, \mathbf{r}(\alpha), y_t) \geq \ell(\alpha, \mathbf{r}(\alpha'), y_t)$ , i.e., the agent inflicts the highest loss to the learner, when best-responding to the action to which she com-*

mitted.

**Definition B.2** (Purely Cooperative Stackelberg Game (PCSGs)). *We call a Stackelberg Game Purely Cooperative if for all actions  $\alpha' \in \mathcal{A}$  for the loss of the learner it holds that:  $\ell(\alpha, \mathbf{r}(\alpha), y_t) \leq \ell(\alpha, \mathbf{r}(\alpha'), y_t)$ , i.e., the agent inflicts the lowest loss to the learner, when best-responding to the action to which she committed.*

We remark here that despite their similarities, PASGs and PC-SGs are *not* equivalent to zero-sum games; in fact, it is easy to see that every zero-sum game is either a PASG or a PCSG, but the converse is *not* true (see e.g., the example loss matrix given in Table B.1 where the first coordinate of tuple  $(i, j)$  corresponds to the loss of the learner, and the second to the loss of the agent). Next, we outline the hierarchy between external and Stackelberg regret in repeated PASGs and PCSGs.

**Lemma B.1.** *In repeated PASGs, Stackelberg regret is upper bounded by external regret, i.e.,  $\mathcal{R}(T) \leq R(T)$ . In other words, any no-Stackelberg regret sequence of actions is also a no-external regret one.*

*Proof.* Let  $\tilde{\alpha} = \arg \min_{\alpha \in \mathcal{A}} \sum_{t=1}^T \ell(\alpha, \mathbf{r}_t(\alpha_t), y_t)$  and  $\alpha^* = \arg \min_{\alpha \in \mathcal{A}} \sum_{t=1}^T \ell(\alpha, \mathbf{r}_t(\alpha), y_t)$ . Then:

$$\begin{aligned} R(T) &= \sum_{t=1}^T \ell(\alpha_t, \mathbf{r}_t(\alpha_t), y_t) - \sum_{t=1}^T \ell(\tilde{\alpha}, \mathbf{r}_t(\alpha_t), y_t) && \text{(definition of external regret)} \\ &\geq \sum_{t=1}^T \ell(\alpha_t, \mathbf{r}_t(\alpha_t), y_t) - \sum_{t=1}^T \ell(\alpha^*, \mathbf{r}_t(\alpha_t), y_t) && \text{(definition of } \tilde{\alpha}) \\ &\geq \sum_{t=1}^T \ell(\alpha_t, \mathbf{r}_t(\alpha_t), y_t) - \sum_{t=1}^T \ell(\alpha^*, \mathbf{r}_t(\alpha^*), y_t) && (\ell(\alpha^*, \mathbf{r}_t(\alpha_t), y_t) \leq \ell(\alpha^*, \mathbf{r}_t(\alpha^*), y_t)) \\ &= \mathcal{R}(T) \end{aligned}$$

|           | $\mathbf{r}_t(\alpha)$ | $\mathbf{r}_t(\alpha')$ |
|-----------|------------------------|-------------------------|
| $\alpha$  | (7, -1)                | (6, -3)                 |
| $\alpha'$ | (6, -3)                | (7, -1)                 |

Table B.1: Example of a PASG that is not zero-sum.

On the other hand, for PCSGs it holds that:

**Lemma B.2.** *In repeated PCSGs, Stackelberg regret is lower bounded by external regret, i.e.,  $\mathcal{R}(T) \geq R(T)$ . In other words, any no-external regret sequence of actions is also a no-Stackelberg regret one.*

*Proof.* Let  $\tilde{\alpha} = \arg \min_{\alpha \in \mathcal{A}} \sum_{t=1}^T \ell(\alpha, \mathbf{r}_t(\alpha_t), y_t)$  and  $\alpha^* = \arg \min_{\alpha \in \mathcal{A}} \sum_{t=1}^T \ell(\alpha, \mathbf{r}_t(\alpha), y_t)$ . Then:

$$\begin{aligned}
R(T) &= \sum_{t=1}^T \ell(\alpha_t, \mathbf{r}_t(\alpha_t), y_t) - \sum_{t=1}^T \ell(\tilde{\alpha}, \mathbf{r}_t(\alpha_t), y_t) && \text{(definition of external regret)} \\
&\leq \sum_{t=1}^T \ell(\alpha_t, \mathbf{r}_t(\alpha_t), y_t) - \sum_{t=1}^T \ell(\tilde{\alpha}, \mathbf{r}_t(\tilde{\alpha}), y_t) && \text{(definition of PCSGs)} \\
&\leq \sum_{t=1}^T \ell(\alpha_t, \mathbf{r}_t(\alpha_t), y_t) - \sum_{t=1}^T \ell(\alpha^*, \mathbf{r}_t(\alpha^*), y_t) && \text{(definition of } \alpha^*) \\
&= \mathcal{R}(T)
\end{aligned}$$

■

### B.1.2 THE FUNCTION $\ell(\alpha, \mathbf{r}_t(\alpha), y_t)$

As we mentioned in the main body, the learner's loss function  $\ell(\alpha, \mathbf{r}_t(\alpha), y_t)$  is generally not Lipschitz in her chosen action  $\alpha$ . For that, we study below the quantity  $|\ell(\alpha, \mathbf{r}_t(\alpha), y_t) - \ell(\alpha', \mathbf{r}_t(\alpha'), y_t)|$ .

**Lemma B.3.** *Let  $\ell(x, y, z)$  denote the learner's loss function in a Stackelberg game, such that  $\ell$  is  $L_1$ -Lipschitz with respect to the first argument, and  $L_2$ -Lipschitz with respect to the second. Then, for the learner's loss between any two actions  $\alpha, \alpha' \in \mathcal{A}$  it holds that:*

$$|\ell(\alpha, \mathbf{r}_t(\alpha), y_t) - \ell(\alpha', \mathbf{r}_t(\alpha'), y_t)| \leq \max \{L_1 \cdot \|\alpha' - \alpha\|, L_2 \cdot \|\mathbf{r}_t(\alpha) - \mathbf{r}_t(\alpha')\|\}$$

*Proof.* We split the set of actions  $\mathcal{A}$  into pairs  $(\alpha, \alpha')$  satisfying the following properties:

1. For pair  $(\alpha, \alpha')$ , we have:  $\ell(\alpha, \mathbf{r}_t(\alpha), y_t) \geq \ell(\alpha, \mathbf{r}_t(\alpha'), y_t)$  and  $\ell(\alpha', \mathbf{r}_t(\alpha'), y_t) \geq \ell(\alpha', \mathbf{r}_t(\alpha), y_t)$ .

In other words, by best-responding the agent causes the biggest loss to the learner. Observe that, given that  $\ell$  is  $L_1$ -Lipschitz in its first argument, we have that:

$$\begin{aligned}
\ell(\alpha', \mathbf{r}_t(\alpha'), y_t) - \ell(\alpha, \mathbf{r}_t(\alpha), y_t) &\geq \ell(\alpha', \mathbf{r}_t(\alpha), y_t) - \ell(\alpha, \mathbf{r}_t(\alpha), y_t) \geq -L_1 \|\alpha' - \alpha\| \\
\ell(\alpha', \mathbf{r}_t(\alpha'), y_t) - \ell(\alpha, \mathbf{r}_t(\alpha), y_t) &\leq \ell(\alpha', \mathbf{r}_t(\alpha'), y_t) - \ell(\alpha, \mathbf{r}_t(\alpha'), y_t) \leq L_1 \|\alpha' - \alpha\|
\end{aligned}$$

Therefore, for such pairs of actions function  $\ell(\alpha, \mathbf{r}_t(\alpha), y_t)$  is  $L_1$ -Lipschitz with respect to  $\alpha$ .

2. For pair  $(\alpha, \alpha')$ , we have:  $\ell(\alpha, \mathbf{r}_t(\alpha), y_t) \leq \ell(\alpha, \mathbf{r}_t(\alpha'), y_t)$  and  $\ell(\alpha', \mathbf{r}_t(\alpha'), y_t) \leq \ell(\alpha', \mathbf{r}_t(\alpha), y_t)$ .

In other words, by best-responding the agent causes the smallest loss to the learner. Similarly to Case 1, it is easy to see that on these pairs of actions, function  $\ell(\alpha, \mathbf{r}_t(\alpha), y_t)$  is again  $L_1$ -Lipschitz with respect to  $\alpha$ .

3. For pair  $(\alpha, \alpha')$ , we have that:

$$\ell(\alpha, \mathbf{r}_t(\alpha), y_t) \geq \ell(\alpha, \mathbf{r}_t(\alpha'), y_t) \quad (\text{B.1.1})$$

$$\ell(\alpha', \mathbf{r}_t(\alpha'), y_t) \leq \ell(\alpha', \mathbf{r}_t(\alpha), y_t) \quad (\text{B.1.2})$$

From Equations (B.1.1) and (B.1.2) we have that

$$\ell(\alpha', \mathbf{r}_t(\alpha'), y_t) - \ell(\alpha, \mathbf{r}_t(\alpha), y_t) \leq L_1 \|\alpha' - \alpha\| \quad (\text{B.1.3})$$

We further distinguish the following cases:

(a)  $\ell(\alpha, \mathbf{r}_t(\alpha), y_t) = \ell(\alpha', \mathbf{r}_t(\alpha'), y_t)$ . Clearly,  $|\ell(\alpha, \mathbf{r}_t(\alpha), y_t) - \ell(\alpha', \mathbf{r}_t(\alpha'), y_t)| \leq L_1 \cdot \|\alpha' - \alpha\|$ .

(b)  $\ell(\alpha, \mathbf{r}_t(\alpha), y_t) \leq \ell(\alpha', \mathbf{r}_t(\alpha'), y_t)$ . From Equation (B.1.3), we get:

$$|\ell(\alpha, \mathbf{r}_t(\alpha), y_t) - \ell(\alpha', \mathbf{r}_t(\alpha'), y_t)| \leq L_1 \cdot \|\alpha' - \alpha\|$$

(c)  $\ell(\alpha, \mathbf{r}_t(\alpha), y_t) \geq \ell(\alpha', \mathbf{r}_t(\alpha'), y_t)$  Observe now that if  $\ell(\alpha, \mathbf{r}_t(\alpha), y_t) \geq \ell(\alpha', \mathbf{r}_t(\alpha'), y_t)$ , then from Equation (B.1.2) the latter is lower bounded by  $\ell(\alpha', \mathbf{r}_t(\alpha'), y_t)$ , which leads to a contradiction. Hence, it has to be the case that  $\ell(\alpha, \mathbf{r}_t(\alpha), y_t) \leq \ell(\alpha', \mathbf{r}_t(\alpha'), y_t)$ . The latter, combined with the assumption that  $\ell$  is  $L_2$  - Lipschitz with respect to its second argument, implies that  $\ell(\alpha', \mathbf{r}_t(\alpha'), y_t) - \ell(\alpha, \mathbf{r}_t(\alpha), y_t) \geq -L_2 \cdot \|\mathbf{r}_t(\alpha') - \mathbf{r}_t(\alpha)\|$ .

4. For pair  $(\alpha, \alpha')$ , we have:  $\ell(\alpha, \mathbf{r}_t(\alpha), y_t) \leq \ell(\alpha, \mathbf{r}_t(\alpha'), y_t)$  &  $\ell(\alpha', \mathbf{r}_t(\alpha'), y_t) \geq \ell(\alpha', \mathbf{r}_t(\alpha), y_t)$ .

The case is analogous to Case 3.

■

To summarize, in PASGs (Case 1 from aforementioned proof) and PCSGs (Case 2 of afore-

mentioned proof) the loss function written in terms of the action of the agent is *Lipschitz*, i.e.,  $|\ell(\alpha, \mathbf{r}_t(\alpha), y_t) - \ell(\alpha', \mathbf{r}_t(\alpha'), y_t)| \leq L_1 \cdot \|\alpha' - \alpha\|$ . However, in General Stackelberg Games one can only guarantee that

$$|\ell(\alpha, \mathbf{r}_t(\alpha), y_t) - \ell(\alpha', \mathbf{r}_t(\alpha'), y_t)| \leq \max \{L_1 \cdot \|\alpha' - \alpha\|, L_2 \cdot \|\mathbf{r}_t(\alpha') - \mathbf{r}_t(\alpha)\|\} \quad (\text{B.1.4})$$

Using Equation (B.1.4), we show that there are some meaningful Stackelberg settings where  $\|\mathbf{r}_t(\alpha') - \mathbf{r}_t(\alpha)\|$  can be upper bounded by  $* \|\alpha' - \alpha\|$  multiplied by a constant. For example, from well known results in convex optimization (for completeness see Lemma B.4), we can see that this is exactly the case in settings where the agent's utility function,  $u_t(\alpha, r)$  is *strongly concave* in  $r$ , and *quasilinear*<sup>\*</sup> in  $\alpha$ .

**Lemma B.4** (Closeness of Maxima of Strongly Concave Functions (folklore)). *Let functions  $f : \mathcal{X} \mapsto \mathbb{R}, g : \mathcal{X} \mapsto \mathbb{R}$  be two multidimensional,  $1/\eta_c$ -strongly concave functions with respect to some norm  $\|\cdot\|$ . Let  $h(\mathbf{x}) = f(\mathbf{x}) - g(\mathbf{x}), \mathbf{x} \in \mathcal{X}$  be  $L_{f,g}$ -Lipschitz<sup>†</sup> with respect to the same norm  $\|\cdot\|$ . Then, for the maxima of the two functions:  $\mu_f = \arg \max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$  and  $\mu_g = \arg \max_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x})$  it holds that:*

$$\|\mu_f - \mu_g\| \leq L_{f,g} \cdot \eta_c \quad (\text{B.1.5})$$

*Proof.* First, we take the Taylor expansion of  $f$  around its maximum,  $\mu_f$  and use the strong concavity condition:

$$\begin{aligned} f(\mathbf{x}) &\leq f(\mu_f) + \langle \nabla f(\mu_f), \mathbf{x} - \mu_f \rangle - \frac{1}{2\eta} \|\mu_f - \mathbf{x}\|^2 && \text{(strong concavity)} \\ &= f(\mu_f) - \frac{1}{2\eta} \|\mu_f - \mathbf{x}\|^2 && (\nabla f(\mu_f) = 0, \text{ since } \mu_f \text{ is the maximum}) \end{aligned}$$

Similarly, by taking the Taylor expansion of  $g$  around its maximum and using strong concavity:

$$g(\mathbf{x}) \leq g(\mu_g) - \frac{1}{2\eta} \|\mu_g - \mathbf{x}\|^2 \quad (\text{B.1.6})$$

---

<sup>\*</sup>Quasilinearity in  $\alpha$  establishes that  $L_{f,g}$  which is used by Lemma B.4 will be linear in  $\|\alpha' - \alpha\|$ .

<sup>†</sup>We use the subscript  $f, g$  in the Lipschitzness constant to denote the fact that it depends on the two functions  $f$  and  $g$ .

Using the  $L_{f,g}$ -Lipschitzness of  $h(\mathbf{x})$  we get:

$$\begin{aligned}
L_{f,g} \cdot \|\mu_g - \mu_f\| &\geq |h(\mu_g) - h(\mu_f)| \geq h(\mu_g) - h(\mu_f) \\
&\geq f(\mu_g) - f(\mu_f) + g(\mu_f) - g(\mu_g) \\
&\geq \frac{1}{2\eta} \|\mu_f - \mu_g\|^2 + \frac{1}{2\eta} \|\mu_f - \mu_g\|^2 \quad (\text{from Taylor expansion}) \\
&\geq \frac{1}{\eta} \|\mu_f - \mu_g\|^2
\end{aligned}$$

Dividing both sides with  $\|\mu_g - \mu_f\|$  concludes the proof. ■

An example of such a utility function in the context of strategic classification (similar to the family of utility functions used in (Dong et al., 2018)) is presented below.

**EXAMPLE.** Let  $u_t(\alpha, \mathbf{r}(\alpha), \sigma_t) = \langle \alpha, \mathbf{r}(\alpha) \rangle - (\mathbf{x} - \mathbf{r}(\alpha))^2$ . Then, we would like to compute an upper bound on the difference between  $\|\mathbf{r}(\alpha) - \mathbf{r}(\alpha')\|$ , where  $\mathbf{r}(\alpha) = \arg \max_{\mathbf{z} \in \mathcal{X}; \mathbf{x}} u_t(\alpha, \mathbf{z}, \sigma_t)$  and  $\mathbf{r}(\alpha') = \arg \max_{\mathbf{z} \in \mathcal{X}; \mathbf{x}} u_t(\alpha', \mathbf{z}, \sigma_t)$ . Following Lemma B.4 we can define functions  $f(\mathbf{z}) = u_t(\alpha, \mathbf{z}, \sigma_t)$  and  $g(\mathbf{z}) = u_t(\alpha', \mathbf{z}, \sigma_t)$ . Now, observe that function  $h(\mathbf{z}) = f(\mathbf{z}) - g(\mathbf{z})$  is indeed  $\|\alpha - \alpha'\|$ -Lipschitz (i.e., the Lipschitzness constant depends on the specific actions):

$$|f(\mathbf{y}) - g(\mathbf{y}) - f(\mathbf{z}) + g(\mathbf{z})| = |\langle \alpha - \alpha', \mathbf{y} - \mathbf{z} \rangle| \leq \|\alpha - \alpha'\| \cdot \|\mathbf{y} - \mathbf{z}\|$$

where the last inequality comes from the Cauchy-Schwartz inequality. Furthermore, observe that both  $f(\cdot)$  and  $g(\cdot)$  are  $\frac{1}{2}$ -strongly concave. Therefore, from Lemma B.4 we get that:

$$\|\mathbf{r}(\alpha) - \mathbf{r}(\alpha')\| \leq \frac{\|\alpha - \alpha'\|}{2}$$

## B.2 APPENDIX FOR SECTION 5.4

### B.2.1 NOTATION REFERENCE TABLES.

Our model and proof use a lot of notation. For easier reference, we summarize the notation used in our analysis in Tables B.2 and B.3.

| Variable  | Description   |
|---|---|
| $d \in \mathbb{N}$  | dimension of the problem  |
| $\mathcal{A} \subseteq [-1, 1]^{d+1}$   | learner's action space  |
| $\alpha_t \in \mathcal{A}$  | learner's committed action for round $t$  |
| $\mathcal{X} \subseteq ([0, 1]^d, 1)$   | agent's feature vector space  |
| $\mathcal{Y} = \{-1, +1\}$  | labels' space   |
| $\mathbf{x}_t \in \mathcal{X}$  | agent's feature vector, <i>as chosen by nature</i>  |
| $\sigma_t = (\mathbf{x}_t, y_t), y_t \in \mathcal{Y}$                           | agent's labeled datapoint, <i>as chosen by nature</i>                                       |
| $\mathbf{r}_t(\alpha_t, \sigma_t) \in \mathcal{X}$ ( $\mathbf{r}_t(\alpha_t)$ ) | agent's <i>reported</i> feature vector  |
| $\hat{y}_t \in \mathcal{Y}$   | $\mathbf{r}_t(\alpha_t)$ 's label   |
| $\ell(\alpha_t, \mathbf{r}_t(\alpha_t), y_t)$                                   | learner's loss for action $\alpha_t$ against agent's report $\mathbf{r}_t(\alpha_t)$        |
| $u_t(\alpha_t, \mathbf{r}_t(\alpha_t), \sigma_t)$                               | agent's utility for reporting $\mathbf{r}_t(\alpha_t)$ , when learner commits to $\alpha_t$ |
| $R(T)$  | learner's <i>external</i> regret after $T$ rounds   |
| $\mathcal{R}(T)$  | learner's <i>Stackelberg</i> regret after $T$ rounds  |
| $\lambda(A)$  | Lebesgue measure of measurable space $A$  |

Table B.2: Model Notation Summary

## B.2.2 REMAINING PROOFS

**Lemma B.5.** *For  $\varepsilon \leq 1/2$ , we call  $\widetilde{\Pr}_t[\alpha_t]$  an  $\varepsilon$ -approximation oracle to  $\mathbb{P}_t^{\text{in}}[\alpha_t]$ , if  $|\widetilde{\Pr}_t[\alpha_t] - \mathbb{P}_t^{\text{in}}[\alpha_t]| \leq \varepsilon \mathbb{P}_t^{\text{in}}[\alpha_t]$ . Then, GRINDER run with oracle  $\widetilde{\Pr}_t[\cdot]$  instead of  $\mathbb{P}_t^{\text{in}}[\alpha_t]$  achieves Stackelberg regret  $\mathcal{R}(T) \leq \mathcal{O}(\sqrt{T \log(T \lambda(\mathcal{A})/\lambda(\underline{p})) \cdot \log(\lambda(\mathcal{A})/\lambda(\underline{p}))}) + 2\varepsilon T$ .*

*Proof of Lemma B.5.* We start by computing how the first moment of estimator  $\widehat{\ell}$  changes once you reweigh with  $\widetilde{\Pr}_t[\cdot]$  rather than  $\mathbb{P}_t^{\text{in}}[\cdot]$ :

$$\begin{aligned} \mathbb{E}_{\alpha_t \sim \mathcal{D}_t} [\widehat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)] &= \int_{\mathcal{A}} f_{\mathcal{A}_t}(\alpha') \frac{\ell(\alpha, \mathbf{r}_t(\alpha), y_t) \mathbb{1}\{\alpha \in N^{\text{out}}(\alpha')\}}{\widetilde{\Pr}_t[\alpha]} d\alpha' \\ &= \ell(\alpha, \mathbf{r}_t(\alpha), y_t) \cdot \frac{\mathbb{P}_t^{\text{in}}[\alpha]}{\widetilde{\Pr}_t[\alpha]} \end{aligned} \tag{B.2.1}$$

Since  $\widetilde{\Pr}_t[\alpha] \geq (1 - \varepsilon) \mathbb{P}_t^{\text{in}}[\alpha]$ , then from Equation (B.2.1) we have that:

$$\mathbb{E}_{\alpha_t \sim \mathcal{D}_t} [\widehat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)] \leq \frac{\ell(\alpha, \mathbf{r}_t(\alpha), y_t)}{1 - \varepsilon} \tag{B.2.2}$$

| Variable  | Description  |
|---|--|
| $\mathcal{P}_t$   | set of active polytopes at round $t$   |
| $\bar{\mathcal{P}}_t$   | set of active point-polytopes at round $t$   |
| $\mathcal{D}_t$   | induced distribution from 2-step sampling process  |
| $\Pr_t, f_{\mathcal{A}_t}$  | cdf and pdf of $\mathcal{D}_t$   |
| $\beta_t^u(\alpha_t) : \langle \mathbf{r}_t(\alpha_t), \mathbf{w} \rangle = 4\sqrt{d}\delta$  | upper boundary hyperplane  |
| $\beta_t^l(\alpha_t) : \langle \mathbf{r}_t(\alpha_t), \mathbf{w} \rangle = -4\sqrt{d}\delta$ | lower boundary hyperplane  |
| $H^+(\beta_t^u(\alpha))$  | $\alpha' \in H^+(\beta_t^u(\alpha)), \text{ if } \langle \mathbf{r}_t(\alpha), \alpha' \rangle \geq 4\sqrt{d}\delta$     |
| $H^-(\beta_t^l(\alpha))$  | $\alpha' \in H^-(\beta_t^l(\alpha)), \text{ if } \langle \mathbf{r}_t(\alpha), \alpha' \rangle \leq -4\sqrt{d}\delta$    |
| $\mathcal{P}_t^u(\alpha)$   | upper polytopes set ( $p \in \mathcal{P}_t : p \in H^+(\beta_t^u(\alpha))$ )   |
| $\mathcal{P}_t^l(\alpha)$   | lower polytopes set ( $p \in \mathcal{P}_t : p \in H^-(\beta_t^l(\alpha))$ )   |
| $\mathcal{P}_t^m(\alpha)$   | middle polytopes set ( $p \in \mathcal{P}_t : p \notin \mathcal{P}_t \setminus (\mathcal{P}_t^l \cup \mathcal{P}_t^u)$ ) |
| $\mathbb{P}_t^{\text{in}}[\alpha], \mathbb{P}_t^{\text{in}}[p]$                               | in-probability for $\alpha$ and $p$ (see Definition 5.3)   |
| $\underline{p}$   | polytope ( $\notin \bar{\mathcal{P}}_t$ ) with smallest Lebesgue measure at $T$  |

Table B.3: Notation Summary for Regret Analysis of GRINDER.

Additionally, since  $\widetilde{\Pr}_t[\alpha] \leq (1 + \varepsilon)\mathbb{P}_t^{\text{in}}[\alpha]$ , then from Equation (B.2.1) we have that:

$$\mathbb{E}_{\alpha_t \sim \mathcal{D}_t} [\widehat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)] \geq \frac{\ell(\alpha, \mathbf{r}_t(\alpha), y_t)}{1 + \varepsilon} \quad (\text{B.2.3})$$

We turn our attention to the second moment now, for which we will only need an upper bound.

$$\begin{aligned} \mathbb{E}_{\alpha_t \sim \mathcal{D}_t} [\widehat{\ell}(\alpha, \mathbf{r}_t(\alpha), y_t)^2] &= \int_{\mathcal{A}} f_{\mathcal{A}_t}(\alpha') \frac{\ell(\alpha, \mathbf{r}_t(\alpha), y_t)^2 \mathbb{1}\{\alpha \in N^{\text{out}}(\alpha')\}}{\widetilde{\Pr}_t[\alpha]^2} d\alpha' = \frac{\ell(\alpha, \mathbf{r}_t(\alpha), y_t)^2 \mathbb{P}_t^{\text{in}}[\alpha]}{\widetilde{\Pr}_t[\alpha]^2} \\ &\leq \frac{1}{(1 - \varepsilon)^2 \mathbb{P}_t^{\text{in}}[\alpha]} \end{aligned} \quad (\text{B.2.4})$$

Lemma 5.3 still holds without any change, as it is not affected by the exact definition of  $\widehat{\ell}(\cdot)$ , and so does Lemma 5.5. Taking expectations in Lemma 5.5 we obtain the following:

$$\begin{aligned} \sum_{t=1}^T \sum_{p \in \mathcal{P}_{t+1}} q_t(p) \mathbb{E} [\widehat{\ell}(p, \mathbf{r}_t(p), y_t)] - \sum_{t=1}^T \mathbb{E} [\widehat{\ell}(\alpha^\star, \mathbf{r}_t(\alpha^\star), y_t)] \\ \leq \frac{\eta}{2} \sum_{t=1}^T \sum_{p \in \mathcal{P}_{t+1}} q_t(p) \mathbb{E} [\widehat{\ell}(p, \mathbf{r}_t(p), y_t)^2] + \frac{1}{\eta} \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) \end{aligned}$$

Applying Equations (B.2.2), (B.2.3) and (B.2.4) on the latter we obtain:

$$\begin{aligned} & \frac{1}{1+\varepsilon} \sum_{t=1}^T \int_{\mathcal{A}} q_t(\alpha) \ell(\alpha, \mathbf{r}_t(\alpha), y_t) d\alpha - \frac{1}{1-\varepsilon} \sum_{t=1}^T \ell(\alpha^\star, \mathbf{r}_t(\alpha^\star), y_t) \\ & \leq \frac{\eta}{2} \frac{1}{(1-\varepsilon)^2} \sum_{t=1}^T \int_{\mathcal{A}} \frac{q_t(\alpha) d\alpha}{\mathbb{P}_t^{\text{in}}[\alpha]} + \frac{1}{\eta} \log \left( \frac{\lambda(\mathcal{A})}{\lambda(\underline{p})} \right) \end{aligned}$$

In the latter, applying Lemma 5.3, multiplying both sides by  $1 - \varepsilon$  and using the fact that  $\varepsilon \leq 1/2$  we obtain the result.  $\blacksquare$

**Lemma B.6.** *Provided access to algorithms for computing the volume of a polytope and to an in-probability oracle, GRINDER has runtime complexity  $\mathcal{O}(T^d)$ .*

*Proof of Lemma B.6.* With access to algorithms that compute the volume of a polytope and to an in-probability oracle, the complexity of GRINDER is dependent solely on the number of polytopes that get activated in the worst case. The latter depends on the number of new boundary hyperplanes that we introduce in the action space  $\mathcal{A}$  at each round.

If the real feature vectors  $\{\mathbf{x}_t\}_{t=1}^T$  is chosen adversarially, the number of new hyperplanes added in each round in  $\mathcal{A}$  is 2. So, *in the worst case*, after  $T$  rounds we have  $2T$  hyperplanes in general position in a  $d$ -dimensional space, which from Zaslavsky (1975), Stanley et al. (2004) are:

$$|\mathcal{P}_t| = O \left( \sum_{i=0}^d \binom{2T}{i} \right) = O \left( \frac{T^d}{d!} \right)$$

$\blacksquare$

### B.3 APPENDIX FOR SECTION 5.5

#### B.3.1 IMPLEMENTING GRINDER FOR CONTINUOUS ACTION SPACES

To implement GRINDER, we used the polytope library<sup>‡</sup>, which is part of the TuLiP python package. Other than some rounding-error fixes, we did not intervene with the core methods of the package.

In order to implement the 2-stage action draw method, we first chose a polytope (according to the probability function prescribed by GRINDER) and then, by using *rejection sampling* from the

---

<sup>‡</sup><https://github.com/tulip-control/polytope/tree/master/requirements>

bounding box around the polytope, we chose the action associated with it. Note that this is equivalent to the theoretical 2-stage draw.

In order to speed up our algorithm's performance, we also used the heuristic of bounding the allowable volume of any polytope to be greater than or equal to 0.01, but in all the simulations that we tried, we saw comparable regret results even without the heuristic.

### B.3.2 LOGISTIC REGRESSION ORACLE

In this subsection, we outline our implementation of the logistic regression algorithm on the agents' past data, which serves as an estimate of the in-probability for each action. For ease of exposition, we provide the description of the oracle for the case of a predefined action set, and subsequently, we outline the way it generalizes to the continuous implementation.

Before we embark on this, allow us first to observe that we already have a very crude (but potentially useful) lower bound for *every* action  $j \in \mathcal{A}$ . Indeed, each action *always* updates itself, and actions that belong in the upper and lower polytope sets are always updated by all actions within these sets. The latter is due to the fact that for any hyperplane chosen within these sets, there is no possible manipulation from the perspective of the agent. We denote this crude lower bound for each action  $j \in \mathcal{A}$  by  $c^j$ .

Labels are defined as  $l_i^j = 0$  if action  $j$  was *not* updated at round  $i$ <sup>§</sup>, and 1 otherwise. As a first step, this oracle computes for each action  $j \in \mathcal{A}$  the probability that each action from  $\mathcal{A}$  updates  $j$ , by using a logistic regression<sup>¶</sup> with feature vectors the set  $H_{1:t}$ , and  $L_{1:t}^j$  as the labels. Let  $p_i^j, i \in \mathcal{A}$  correspond to the output probabilities, i.e.,  $p_i^j$  encodes the probability that action  $j$  will be updated by action  $i$ . The in-probability of action  $j$  is ultimately defined as:

$$\Pr^{in}[j] = \max \left\{ \sum_{i \in \mathcal{A}} p_i^j \pi_t[i], c^j \right\}$$

At a high-level, it is not hard to see how this can generalize to the continuous grinding case; instead of actions, one now uses whole *polytopes*. The implementation, however, becomes significantly messier, as we need to propagate the history of past data for each polytope to its grinded

---

<sup>§</sup>In other words, action was at a distance less than  $2\delta$  from the best-response of the round.

<sup>¶</sup>Technically, we run a different logistic regression for every action in  $\mathcal{A}$ .

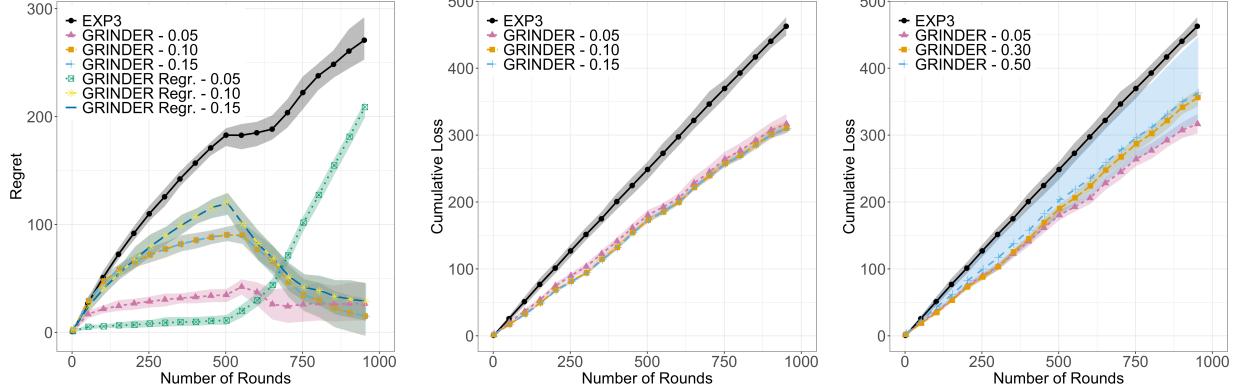


Figure B.1: GRINDER vs. EXP3 for utility function from Eq. (B.3.1). From left to right: discrete  $\mathcal{A}$  (accurate and regression oracle), continuous  $\mathcal{A}$  with  $\delta = 0.05, 0.10, 0.15$  and continuous  $\mathcal{A}$  with  $\delta = 0.05, 0.3, 0.5$ . Solid lines correspond to average regret/loss, and opaque bands correspond to 10th and 90th percentile.

sub-polytopes.

### B.3.3 DIFFERENT UTILITY FUNCTION AND DISTRIBUTION OF DATAPOINTS

The utility function that we assume for the agents at this subsection, is similar to the one studied by Dong et al. (2018), specifically:

$$u_t(\alpha_t, \mathbf{r}_t(\alpha_t), y_t) = \delta \cdot \langle \alpha_t, \mathbf{r}_t(\alpha_t) \rangle - \|\mathbf{x}_t - \mathbf{r}_t(\alpha_t)\|_2 \quad (\text{B.3.1})$$

for values of  $\delta = 0.05, 0.10, 0.15, 0.3, 0.5$ . Similarly to the paper's main body, we run GRINDER against EXP3 for a horizon  $T = 1000$ , where each round was repeated for 30 repetitions.

Figure B.1 presents the results for the case where the  $+1$  labeled points are drawn as  $\mathbf{x}_t \sim (\mathcal{N}(0.7, 0.3), \mathcal{N}(0.7, 0.3))$  and the  $-1$  labeled points are drawn from  $\mathbf{x}_t \sim (\mathcal{N}(0.4, 0.3), \mathcal{N}(0.4, 0.3))$ . The performance of GRINDER compared to EXP3 is similar to the one that we saw in Section 5.5 for the case of the different utility function. GRINDER outperforms EXP3, and its performance degrades as the power of the agent (i.e.,  $\delta$ ) increases. We also see that in this case, the regression oracles are performing slightly worse than the regression oracles for the case of the utility function analyzed in Section 5.5.

Finally, in Figure B.2, we present the results of our simulations of running GRINDER against EXP3, when the agents' utility function is defined by Equation (B.3.1), and the distribution of labeled

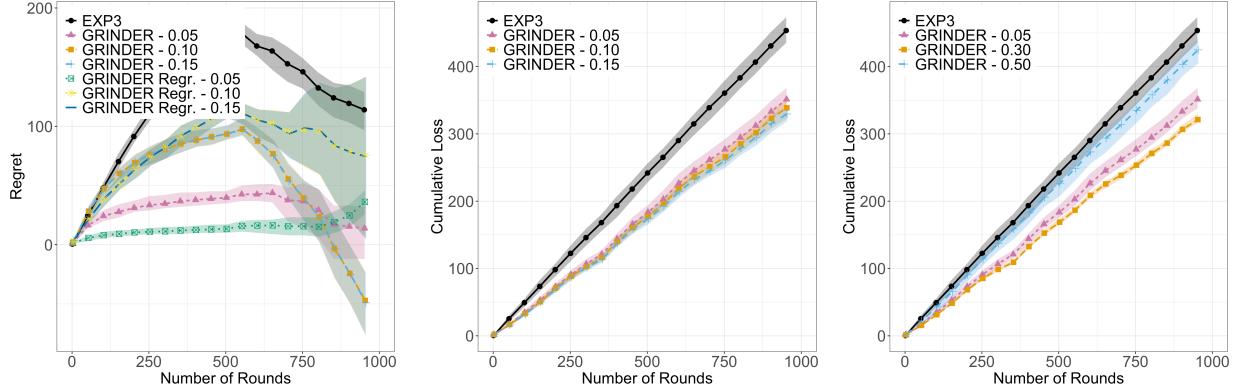


Figure B.2: GRINDER vs. EXP3 for “harder” distribution of labels. From left to right: discrete  $\mathcal{A}$  (accurate and regression oracle), continuous  $\mathcal{A}$  with  $\delta = 0.05, 0.10, 0.15$  and continuous  $\mathcal{A}$  with  $\delta = 0.05, 0.3, 0.5$ . Solid lines correspond to average regret/loss, and opaque bands correspond to 10th and 90th percentile.

points is the following: the  $+1$  labeled points are drawn as  $x_t \sim (\mathcal{N}(0.6, 0.4), \mathcal{N}(0.4, 0.6))$  and the  $-1$  labeled points are drawn from  $x_t \sim (\mathcal{N}(0.4, 0.6), \mathcal{N}(0.6, 0.4))$ . We note that while GRINDER still outperforms EXP3 its performance has become worse than what we saw in Figure (B.1). This is because in this new distribution of points creates much higher overlap of labels and there are fewer points for which a perfect linear classifier exists. This is also exhibited by the fact that EXP3’s performance is getting better in the horizon of  $T$  rounds compared to any single fixed action.

# C

## Appendix for Chapter 6

### C.1 PROBABILITY TOOLS

We state the Markov and the Azuma-Hoeffding inequalities, standard tools that we use.

**Lemma C.1** (Markov Inequality). *If  $\varphi$  is a monotonically increasing non-negative function for the non-negative reals,  $X$  is a random variable,  $a \geq 0$  and  $\varphi(a) > 0$ , then:*

$$\Pr [|X| \geq a] \leq \frac{\mathbb{E} [\varphi(X)]}{\varphi(a)}$$

**Lemma C.2** (Azuma-Hoeffding Inequality). *Suppose  $\{X_k : k = 0, 1, 2, \dots\}$  is a martingale and  $|X_k -$*

$X_{k-1} \leq c_k$ . Then, for all positive integers  $N$  and all  $\varepsilon > 0$  we have that:

$$\Pr [|X_N - X_0| \geq \varepsilon] \leq 2 \exp \left( -\frac{\varepsilon^2}{2 \sum_{k=1}^N c_k^2} \right).$$

We use the in-expectation Lipschitz condition (Equation (6.1.1)) to derive a high-probability, per-realization version, which is the version directly used by our analysis. It is a simple corollary of the Azuma-Hoeffding inequality, independent of the rest of the analysis.

**Lemma C.3** (Per-realization Lipschitz property). *Fix round  $t$ , two sequences of arms  $(y_1, \dots, y_t)$  and  $(y'_1, \dots, y'_t)$ , and failure probability  $\delta > 0$ . Then with probability at least  $1 - \delta$  we have:*

$$\sum_{\tau \in [t]} g_\tau(y_\tau) - \sum_{\tau \in [t]} g_\tau(y'_\tau) \leq 2\sqrt{2t \ln(2/\delta)} + \sum_{\tau \in [t]} \mathcal{D}(y_\tau, y'_\tau). \quad (\text{C.1.1})$$

*Proof.* We apply the Azuma-Hoeffding inequality to martingale  $Y_t = \sum_{\tau \in [t]} g_\tau(y_\tau) - \mathbb{E}[g_\tau(y_\tau)]$ . We conclude that  $\Pr [|Y_t| \geq \sqrt{2t \ln(2/\delta)}] \leq \delta$ . In other words, with probability at least  $1 - \delta$ ,

$$\left| \sum_{\tau=1}^t g_\tau(y_\tau) - \sum_{\tau=1}^t \mathbb{E}[g_\tau(y_\tau)] \right| \leq \sqrt{2t \ln(2/\delta)}.$$

A similar inequality holds for sequence  $(y'_1, \dots, y'_t)$ . Combining both inequalities with Equation (6.1.1) gives the stated result. ■

## C.2 EXTENSION TO ARBITRARY METRIC SPACES

In this appendix, we sketch out an extension to arbitrary metric spaces. The main change is that the zooming tree is replaced with a more detailed decomposition of the action space. Similar decompositions have been implicit in all prior work on adaptive discretization, starting from ([Kleinberg et al., 2019](#), [Bubeck et al., 2011a](#)). No substantial changes in the algorithm or analysis are needed.

**Preliminaries.** Fix subset  $S \subset \mathcal{A}$  and  $\varepsilon > 0$ . The diameter of  $S$  is  $\sup_{x,y \in S'} \mathcal{D}(x, y)$ . An  $\varepsilon$ -covering of  $S$  is a collection of subsets  $S' \subset \mathcal{A}$  of diameter at most  $\varepsilon$  whose union covers  $S$ . The  $\varepsilon$ -covering number of  $S$ , denoted  $\mathcal{N}_\varepsilon(S)$ , is the smallest cardinality of an  $\varepsilon$ -covering. Note that the covering property in (6.1.4) can be restated as  $\inf \{d \geq 0 : \mathcal{N}_\varepsilon(\mathcal{A}_\varepsilon) \leq \gamma \cdot \varepsilon^{-d}, \forall \varepsilon > 0\}$ .

A *greedy*  $\varepsilon$ -covering of  $S$  is an  $\varepsilon$ -covering constructed by the following “greedy” algorithm: while there is a point  $x \in S$  which is not yet covered, add the closed ball  $B(x, \varepsilon/2)$  to the covering. Thus, this  $\varepsilon$ -covering consists of closed balls of radius  $\varepsilon/2$  whose centers are at distance more than  $\varepsilon/2$ .

A *rooted* directed acyclic graph (DAG) is a DAG with a single source node, called the *root*. For each node  $u$ , the distance from the root is called the *height* of  $u$  and denoted  $h(u)$ . The subset of nodes reachable from  $u$  (including  $u$  itself) is called the *sub-DAG* of  $u$ . For an edge  $(u, v)$ , we say that  $u$  is a *parent* and  $v$  is a *child* relative to one another. The set of all children of  $u$  is denoted  $\mathcal{C}(u)$ .

**Metric Space Decomposition.** Our decomposition is a rooted DAG, called *Zooming DAG*, whose nodes correspond to balls in the metric space.

**Definition C.1** (zooming DAG). A zooming DAG is a rooted DAG of infinite height. Each node  $u$  corresponds to a closed ball  $B(u)$  in the action space, with radius  $r(u) = 2^{-h(u)}$  and center  $x(u) \in \mathcal{A}$ . These objects are called, respectively, the *action-ball*, the *action-radius*, and the *action-center* of  $u$ . The following properties are enforced:

- (a) each node  $u$  is covered by the children:  $B(u) \subset \cup_{v \in \mathcal{C}(u)} B(v)$ .
- (b) each node  $u$  overlaps with each child  $v$ :  $B(u) \cap B(v) \neq \emptyset$ .
- (c) for any two nodes of the same action-radius  $r$ , their action-centers are at distance  $> r$ .

The *action-span* of  $u$  is the union of all action-balls in the sub-DAG of  $u$ .

Several implications are worth spelling out:

- the nodes with a given action-radius  $r$  cover the action space (by property (a)), and there are at most  $\mathcal{N}_r(\mathcal{A})$  of them (by property (c)). Recall that  $\mathcal{N}_r(\mathcal{A}) \leq \gamma \cdot r^{-d}$ , where  $d$  is the covering dimension with multiplier  $\gamma$ .
- each node  $u$  has at most  $\mathcal{N}_{r(u)/2}(B(u)) \leq C_{\text{dbl}}$  children (by properties (b,c)), and its action-span lies within distance  $3r(u)$  from its action-center (by property (b)).

A zooming DAG exists, and can be constructed as follows. The nodes with a given action-radius  $r$  are constructed as a greedy  $(2r)$ -cover of the action space. The children of each node  $u$  are all nodes of action-radius  $r(u)/2$  whose action-balls overlap with  $B(u)$ .

Our algorithm only needs nodes of height up to  $O(\log T)$ . We assume that some “zooming DAG”, denoted  $\text{ZoomDAG}$ , is fixed and known to the algorithm.

Note that a given node in  $\text{ZoomDAG}$  may have multiple parents. Our algorithm adaptively constructs subsets of  $\text{ZoomDAG}$  that are directed trees. Hence a definition:

**Definition C.2** (zooming tree). *A subgraph of  $\text{ZoomDAG}$  is called a zooming tree if it is a finite directed tree rooted at the root of  $\text{ZoomDAG}$ . The ancestor path of node  $u$  is the path from the root to  $u$ .*

For a  $d$ -dimensional unit cube,  $\text{ZoomDAG}$  can be defined as a zooming tree, as per Section 6.2.

**Changes in the Algorithm.** When zooming in on a given node  $u$ , it activates all children of  $u$  in  $\text{ZoomDAG}$  that are not already active (whereas the version in Section 6.2 activates all children of  $u$ ). The representative arms  $\text{repr}_t(u)$  are chosen from the action-ball of  $u$ .

**Changes in the Analysis.** We account for the fact that the action-span of each node  $u$  lies within  $3r(u)$  of its action-center (previously it was just  $r(u)$ ). This constant 3 is propagated throughout.

# D

## Appendix for Chapter 7

### D.1 UNCORRUPTED CONTEXTUAL SEARCH FOR $\varepsilon$ -BALL LOSS

#### D.1.1 PROJECTEDVOLUME ALGORITHM AND INTUITION

In this subsection, we describe the PROJECTEDVOLUME algorithm of [Lobel et al. \(2018\)](#), which is the algorithm that CORPV.KNOWN builds on. PROJECTEDVOLUME minimizes the  $\varepsilon$ -ball loss for fully rational agents by approximately estimating  $\theta^*$ . At all rounds  $t \in [T]$  PROJECTEDVOLUME maintains a convex body, called the *knowledge set* and denoted by  $\mathcal{K}_t \in \mathbb{R}^d$ , which corresponds to all parameters  $\theta$  that are not ruled out based on the information until round  $t$ . It also maintains a set of orthonormal vectors  $S_t = \{\mathbf{s}_1, \dots, \mathbf{s}_{|S_t|}\}$  such that  $\mathcal{K}_t$  has small width along these directions, i.e.,  $\forall \mathbf{s} \in S_t : w(\mathcal{K}_t, \mathbf{s}) \leq \delta'$ . The algorithm “ignores” a dimension of  $\mathcal{K}_t$ , once it becomes small, and

focuses on the projection of  $\mathcal{K}_t$  onto a set  $L_t$  of dimensions that are orthogonal to  $S_t$  and have larger width, i.e.,  $\forall \mathbf{l} \in L_t : w(\mathcal{K}_t, \mathbf{l}) \geq \delta'$ .

---

**ALGORITHM D.1: PROJECTEDVOLUME** ([Lobel et al., 2018](#))

---

- 1 Initialize  $S_0 \leftarrow \emptyset, \mathcal{K}_0 \leftarrow \mathcal{K}$ .
  - 2 **for**  $t \in [T]$  **do**
  - 3     context  $\mathbf{x}_t$ , chosen by nature.
  - 4     Query point  $\omega_t = \langle \mathbf{x}_t, \kappa_t \rangle$ , where  $\kappa_t \leftarrow \text{approx-centroid}(\text{Cyl}(\mathcal{K}_t, S_t))$ .
  - 5     Observe feedback  $y_t$  and set  $\mathcal{K}_{t+1} \leftarrow \mathcal{K}_t \cap \mathbf{H}^+(\mathbf{x}_t, \omega_t)$  if  $y_t = +1$  or  $\mathcal{K}_{t+1} \leftarrow \mathcal{K}_t \cap \mathbf{H}^-(\mathbf{x}_t, \omega_t)$   
if  $y_t = -1$ .
  - 6     Add all directions  $\mathbf{u}$  orthogonal to  $S_t$  with  $w(\mathcal{K}_{t+1}, \mathbf{u}) \leq \delta' = \frac{\varepsilon^2}{16d(d+1)^2}$  to  $S_t$ .
  - 7     Set  $S_{t+1} = S_t$ .
- 

At round  $t$ , after observing  $\mathbf{x}_t$ , the algorithm queries point  $\omega_t = \langle \mathbf{x}_t, \kappa_t \rangle$  where  $\kappa_t$  is the *approximate* centroid of knowledge set  $\mathcal{K}_t$ . Based on the feedback,  $y_t$ , the algorithm eliminates one of  $\mathbf{H}^+(\mathbf{x}_t, \omega_t)$  or  $\mathbf{H}^-(\mathbf{x}_t, \omega_t)$ . The analysis uses the volume of  $\Pi_{L_t} \mathcal{K}_t$ , denoted by  $\text{vol}(\Pi_{L_t} \mathcal{K}_t)$ , as a potential function. After each query either the set of small dimensions  $S_t$  increases, thus making  $\text{vol}(\Pi_{L_t} \mathcal{K}_t)$  increase by a bounded amount (which can happen at most  $d$  times), or  $\text{vol}(\Pi_{L_t} \mathcal{K}_t)$  decreases by a factor of  $(1 - 1/e^2)$ . This potential function argument leads to a regret of at most  $\mathcal{O}(d \log(d/\varepsilon))$ .

### D.1.2 FAILURE OF PROJECTEDVOLUME AGAINST CORRUPTIONS IN ONE DIMENSION

When  $d = 1$  and  $\bar{c} = 0$ , there exists a  $\theta^* \in \mathbb{R}$  and nature replies whether  $\omega_t$  is *greater* or *smaller* than  $\theta^*$ . By appropriate queries, the learner can decrease the size of the knowledge set that is consistent with all past queries so that, after  $\log(1/\varepsilon)$  rounds, she identifies an  $\varepsilon$ -ball containing  $\theta^*$ . However, even when  $\bar{c} = 1$ , the above algorithm can be easily misled. Think about an example as in Figure D.1a where  $\theta^* = 3/4$ , the learner queries point  $1/2$ , and nature corrupts the feedback making her retain the interval  $[0, 1/2]$ , instead of  $[1/2, 1]$ , as her current knowledge set.

That said, if the learner knows  $\bar{c}$  then, by repeatedly querying the same point, she can guarantee that if she observes  $y_t = +1$  (resp.  $y_t = -1$ ) for *at least*  $\bar{c} + 1$  times, then feedback  $y_t = +1$  (resp.  $y_t = -1$ ) is surely consistent with  $\theta^*$ . Hence, by repeating each query  $2\bar{c} + 1$  times the learner incurs regret at most  $(2\bar{c} + 1) \log(1/\varepsilon)$ . Unfortunately, in higher dimensions it is impossible to repeat the

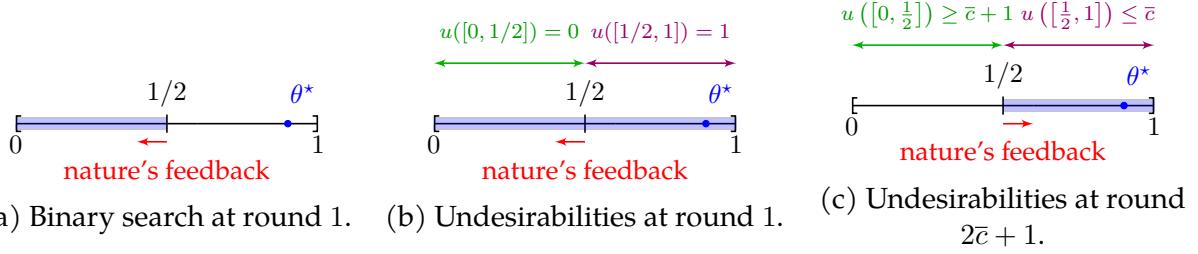


Figure D.1: Single dimensional binary search. The opaque band is the knowledge set after each query.

exact same query, as nature chooses different contexts.

## D.2 EXTENSION TO UNKNOWN CORRUPTION (PROOF OF THEOREM 7.1)

Before providing the proof of Theorem 7.1, we give some auxiliary probabilistic lemmas.

**Lemma D.1** ((Lykouris et al., 2018, Lemma 3.3)). *For corruption level  $C$ , each layer  $j \geq \log C$  observes at most  $\ln(1/\beta) + 3$  corruptions with probability at least  $1 - \beta$ .*

**Lemma D.2.** *Let  $X_1, \dots, X_n$  denote  $n$  random binary variables that take value of 0 with probability at most  $p_1, \dots, p_n$  respectively. Then, the following is true:*

$$\Pr \left[ \bigcap_{i \in [n]} X_i \right] \geq 1 - \sum_{i \in [n]} p_i$$

*Proof.* This inequality is proven using the union bound as follows:

$$\Pr \left[ \bigcap_{i \in [n]} X_i \right] = 1 - \Pr[\exists j : X_j = 0] \geq 1 - \sum_{j \in [n]} p_j$$

■

**Lemma D.3.** *Let  $X$  be a random variable following the binomial distribution with parameters  $n$  and  $p$ , such that  $p = 1/a$  for some  $a > 0$ . Then,  $\Pr[X < 1] \leq \delta$  for  $n = a \cdot \log(1/\delta)$ .*

*Proof.* Using the definition of the binomial distribution we have that:  $\Pr[X < 1] = \Pr[X = 0] =$

$(1 - p)^n$ . For any  $\beta$  in order for the result to hold one needs

$$n \geq \frac{\log(1/\beta)}{\log\left(\frac{1}{1-p}\right)} \quad (\text{D.2.1})$$

Since  $\log\left(\frac{1}{1-p}\right) \geq \frac{p}{1-p}$ , Equation (D.2.1) is satisfied. Choosing  $p = 1/a$  we get the result.  $\blacksquare$

*Proof of Theorem 7.1.* We present the proof for the  $\varepsilon$ -ball loss. Tuning  $\varepsilon = 1/T$  afterwards gives the stated result for the absolute and pricing loss.

We separate the layers into two categories: layers  $j \geq \log C$  are *corruption-tolerant*, and layers  $j < \log C$  are *corruption-intolerant*. Every layer  $j$ , if it were to run in isolation, would spend  $\Phi_j$  epochs until converging to a knowledge set with width at most  $\varepsilon$  in all the directions. However, in CorPV.AC layer  $j$ 's epoch potentially gets increased every time that a layer  $j' \geq j$  changes epoch. Since there are at most  $\log T$  layers, this results in an added  $\log T$  multiplicative overhead for the epochs of each layer. This overhead is suffered by the corruption-tolerant layers.

We first study the performance of the corruption-tolerant layers. Let  $\beta_j > 0$  denote the failure probability for layer  $j$  such that  $\beta_j \leq \frac{\beta}{\log T + 1}$ . From Lemma D.1, with probability at least  $1 - \beta_j$ , the actual corruption experienced by the tolerant layers is at most

$$\mathcal{C} = \ln\left(\frac{1}{\beta_j}\right) + 3 \leq \log\left(\frac{T}{\beta}\right). \quad (\text{D.2.2})$$

From Proposition 7.1 for all rounds that this corruption-tolerant layer was sampled, the regret incurred by each tolerant layer  $j$ , denoted by  $R_{\text{tol},j}$ , is upper bounded by:

$$R_{\text{tol},j} \leq \mathcal{O}\left((d^2\mathcal{C} + 1) d \log\left(\frac{d}{\varepsilon}\right)\right) \quad (\text{D.2.3})$$

There are at most  $\log T$  tolerant layers. So, with probability at least  $1 - \bar{\beta}$  (Lemma D.2), where  $\bar{\beta} = \sum_{j \in [\log T]} \beta_j = \frac{\log T}{\log T + 1} \beta$  for all the corruption-tolerant layers:

$$R_{\text{tolerant}} \leq \sum_{j=1}^{\log T} R_{\text{tol},j} \quad (\text{D.2.4})$$

We now move to the analysis of the corruption-intolerant layers. Let  $j^*$  denote the smallest corruption-tolerant layer, i.e.,  $j^* = \min_j \{j \geq \log C\}$ . Observe that each layer  $j \leq j^*$  is played until layer  $j^*$  identifies the target knowledge set having width at most  $\varepsilon$  in every direction. If  $j^*$  was run in isolation, from Equation (D.2.3) it would incur regret  $R_{\text{tol},j^*}$ . When a context is not costly for  $j^*$ , it is also not costly for layers  $j < j^*$ . This follows because we have consistent knowledge sets and sets of small dimensions across the layers. As a result, whenever a context causes regret for a corruption-intolerant layer, with probability  $1/C$ ,  $j^*$  is selected and it makes progress towards identifying the target. Using standard arguments for the binomial distribution (see Lemma D.3) we can show that for any scalar  $\tilde{\beta} > 0$  with probability at least  $1 - \tilde{\beta}$ , layer  $j^*$  is played *at least once* every  $N = C \log(1/\tilde{\beta})$  rounds. Set  $\tilde{\beta}$  to be  $\tilde{\beta} \leq \beta/(\log T + 1)$ . Hence, the total regret from corruption-intolerant layers can be bounded by the total regret incurred by the first corruption-tolerant layer times  $N$ . Mathematically:

$$\begin{aligned} R_{\text{intolerant}} &\leq N \cdot R_{j^*} = \mathcal{O}\left(N \cdot (2d(d+1) \log C + 1) d \log\left(\frac{d}{\varepsilon}\right)\right) \\ &= \mathcal{O}\left(C \cdot (2d(d+1) \log C + 1) d \log\left(\frac{d}{\varepsilon}\right) \log\left(\frac{1}{\tilde{\beta}}\right)\right) \end{aligned} \quad (\text{D.2.5})$$

until the appropriately small knowledge set is constructed for  $j^*$ ; subsequently this knowledge set dictates the behavior of the intolerant layers.

Putting everything together, and using the union bound again, we have that with probability at least  $1 - \sum_{j \in [n]} \beta_j - \tilde{\beta} = 1 - \beta$  the regret of CorPV.AC is:

$$\begin{aligned} R &= R_{\text{tolerant}} + R_{\text{intolerant}} && \text{(Equations (D.2.3) and (D.2.5))} \\ &\leq \mathcal{O}\left((\log T + C) \cdot (d^2 \mathcal{C} + 1) d \log\left(\frac{d}{\varepsilon}\right) \cdot \log\left(\frac{1}{\beta}\right)\right) \\ &\leq \mathcal{O}\left(d^3 \cdot \log\left(\min\left\{T, \frac{d}{\varepsilon}\right\} \cdot \frac{1}{\beta}\right) \cdot \log\left(\frac{d}{\varepsilon}\right) \cdot \log\left(\frac{1}{\beta}\right) \cdot (\log(T) + C)\right) \end{aligned}$$

We finally discuss the computational complexity of CorPV.AC. Note that the complexity is dictated by the choice of  $\mathcal{C} = \text{poly log}(T)$ . As a result, from Lemma 7.5, substituting  $C$  with  $\log T$ , we get the stated expected runtime for CorPV.AC: ■

### D.3 AUXILIARY LEMMAS

#### D.3.1 AUXILIARY LEMMAS FOR LEMMA 7.1

**Lemma D.4.** *If  $\nu > \underline{\nu}$  (where  $\underline{\nu} = \sqrt{d}\delta$ ) then, for any point  $\mathbf{p}$ , we have that:*

$$u_\phi(\mathbf{p}, \nu) = \sum_{t \in [\tau]} \mathbb{1} \left\{ (\langle \mathbf{p} - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) \cdot (\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) < 0 \right\}$$

*Proof.* In order to prove the lemma, we argue that  $\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu > 0$ . Recall that the feedback in Step 3 of the protocol is defined as  $\text{sgn}(\tilde{v}_t - \omega_t)$ , and that we set  $y_t = +1$  and  $\tilde{v}_t = \langle \mathbf{x}_t, \boldsymbol{\theta}_t \rangle$ . As a result,  $\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \mathbf{x}_t \rangle \geq 0$  and expanding:

$$\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{S_\phi} \mathbf{x}_t \rangle \geq 0 \quad (\text{D.3.1})$$

We proceed by upper bounding the quantity  $\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{S_\phi} \mathbf{x}_t \rangle$ . Let  $S$  be a matrix with columns corresponding to the basis of vectors in  $S_\phi$ , so that  $\Pi_{S_\phi} = SS^\top$ . Then, we obtain:

$$\begin{aligned} \langle \Pi_{S_\phi} \mathbf{x}_t, \mathbf{p} - \boldsymbol{\kappa}_\phi \rangle &= \langle \Pi_{S_\phi} \mathbf{x}_t, \Pi_{S_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi) \rangle \leq |\langle \Pi_{S_\phi} \mathbf{x}_t, \Pi_{S_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi) \rangle| \\ &\leq \|\Pi_{S_\phi} \mathbf{x}_t\|_2 \cdot \|\Pi_{S_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi)\|_2 && \text{(Cauchy-Schwarz inequality)} \\ &= \|\Pi_{S_\phi} \mathbf{x}_t\|_2 \cdot \|S^\top(\mathbf{p} - \boldsymbol{\kappa}_\phi)\|_2 \\ &\leq \|\mathbf{x}_t\|_2 \cdot \sqrt{d} \cdot \|S^\top(\mathbf{p} - \boldsymbol{\kappa}_\phi)\|_\infty && (\|\mathbf{z}\|_2 = \sqrt{\sum_{i \in [d]} \mathbf{z}_i^2} \leq \sqrt{d \cdot \|\mathbf{z}\|_\infty^2}) \\ &\leq 1 \cdot \delta \sqrt{d}. && (\|\mathbf{x}_t\|_2 = 1 \text{ and } w(\mathcal{K}_\phi, \mathbf{s}) \leq \delta, \forall \mathbf{s} \in S_\phi) \end{aligned}$$

Using this to relax Equation (D.3.1) along with  $\underline{\nu} = \sqrt{d}\delta$  we get that:  $\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle \geq -\underline{\nu}$ . Since  $\nu > \underline{\nu}$ , it follows that  $\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu \geq -\underline{\nu} + \nu > 0$ . Combined with Definition 7.2, this concludes the lemma.  $\blacksquare$

*Proof of Lemma 7.2.* By Lemma D.4,

$$u_\phi(\mathbf{p}, \nu) = \sum_{t \in [\tau]} \mathbb{1} \left\{ (\langle \mathbf{p} - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) \cdot (\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) < 0 \right\}.$$

For the uncorrupted rounds  $\theta^* = \mathbf{p} = \theta_t$ ; as a result, the corresponding summands are non-negative:  $(\langle \mathbf{p} - \kappa_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) \cdot (\langle \theta_t - \kappa_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) \geq 0$ . Hence, the only rounds for which  $\theta^*$  can incur undesirability are the corrupted rounds, of which there are at most  $\bar{c}$ . As a result,  $u_\phi(\theta^*, \nu) \leq \bar{c}$  and  $\theta^* \in \mathcal{P}(\bar{c}, \nu)$  by the definition of region  $\mathcal{P}(\bar{c}, \nu)$ . ■

Before proving Lemma D.6 we need the following technical lemma, whose proof follows ideas from [Lobel et al. \(2018\)](#) and at the end of the section for completeness.

**Lemma D.5.** *Let basis  $E_\phi = \{\mathbf{e}_1, \dots, \mathbf{e}_{d-|S_\phi|}\}$  be orthogonal to  $S_\phi$ . For all  $\{(\mathbf{x}_t, \omega_t)\}_{t \in [\tau]}$  such that  $w(\text{Cyl}(\mathcal{K}_\phi, S_\phi), \mathbf{x}_t) \geq \varepsilon$ , there exists  $i$  such that:  $|\langle \mathbf{e}_i, \mathbf{x}_t \rangle| \geq \bar{\nu}$ , where  $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\delta}{4\sqrt{d}}$ .*

The tuning of  $\bar{\nu}$  explains the constraint imposed on  $\delta$ , i.e.,  $\delta < \frac{\varepsilon}{2\sqrt{d+4\delta}}$ . This constraint is due to the fact that since  $\nu > \underline{\nu}$  and  $\nu < \bar{\nu}$ , then it must be the case that  $\underline{\nu} < \bar{\nu}$ , where  $\underline{\nu} = \sqrt{d}\delta$  and  $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}}{4\sqrt{d}}$ .

**Lemma D.6.** *For every round  $t \in [\tau]$ , any scalar  $\delta \in (0, \frac{\varepsilon}{2\sqrt{d+4\delta}})$ , any scalar  $\nu < \bar{\nu}$ , at least one of the landmarks in  $\Lambda_\phi$  gets one  $\nu$ -margin projected undesirability point, i.e.,*

$$\exists \mathbf{p} \in \Lambda_\phi : (\langle \mathbf{p} - \kappa_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) < 0.$$

*Proof.* By Lemma D.5, there exists a direction  $\mathbf{e}_i \in E_\phi$  such that  $|\langle \mathbf{e}_i, \mathbf{x}_t \rangle| \geq \bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\delta}{4\sqrt{d}}$ . The proof then follows by showing that for  $\nu < \bar{\nu}$  landmark points  $\mathbf{q}_+ = \kappa_\phi + \nu \cdot \mathbf{e}_i$  and  $\mathbf{q}_- = \kappa_\phi - \nu \cdot \mathbf{e}_i$  get different signs in the undesirability point definition. This is shown by the following derivation:

$$\begin{aligned} & (\langle \mathbf{q}_+ - \kappa_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) \cdot (\langle \mathbf{q}_- - \kappa_\phi, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) \\ &= (\langle \bar{\nu} \cdot \mathbf{e}_i, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) \cdot (\langle -\bar{\nu} \cdot \mathbf{e}_i, \Pi_{L_\phi} \mathbf{x}_t \rangle + \nu) \\ &= \nu^2 - (\bar{\nu} \cdot |\langle \mathbf{e}_i, \mathbf{x}_t \rangle|)^2 \leq \nu^2 - \bar{\nu}^2 < 0 \end{aligned}$$

where the last inequality comes from the fact that  $\nu \in (\underline{\nu}, \bar{\nu})$ . As a result there exists  $\mathbf{p} \in \{\mathbf{q}_+, \mathbf{q}_-\} \subseteq \mathcal{L}_\phi$  satisfying the condition in the lemma statement. ■

*Proof of Lemma 7.4.* At each of the  $\tau$  explore rounds, at least one of the landmarks gets a  $\nu$ -margin projected undesirability point (Lemma D.6). Since there are *at most*  $2d$  landmarks, by the pigeon-hole principle after  $\tau$  rounds, there exists at least one of them with  $\nu$ -margin projected undesirabil-

ity  $u_\phi(\mathbf{p}^\star, \nu) \geq \bar{c} \cdot (d+1) + 1$ . Since all points  $\mathbf{q}$  inside  $\text{conv}(\mathcal{P}(\bar{c}, \nu))$  have  $u_\phi(\mathbf{q}, \nu) \leq \bar{c} \cdot (d+1)$ , then  $\mathbf{p} \notin \text{conv}(\mathcal{P}(\bar{c}, \nu))$ . ■

*Proof of Lemma D.5.* We first show that  $\|\Pi_{L_\phi} \mathbf{x}_t\| \geq \frac{\varepsilon - 2\sqrt{d}\delta}{4}$ . Since for the contexts  $\{\mathbf{x}_t\}_{t \in [\tau]}$  that we consider in epoch  $\phi$  it holds that:  $w(\text{Cyl}(\mathcal{K}_\phi, S_\phi), \mathbf{x}_t) \geq \varepsilon$ , then there exists a point  $\mathbf{p} \in \text{Cyl}(\mathcal{K}_\phi, S_\phi)$  such that  $|\langle \mathbf{x}_t, \mathbf{p} - \boldsymbol{\kappa}_\phi \rangle| \geq \frac{\varepsilon}{2}$ . Applying the triangle inequality:

$$|\langle \Pi_{L_\phi} \mathbf{x}_t, \Pi_{L_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi) \rangle| + |\langle \Pi_{S_\phi} \mathbf{x}_t, \Pi_{S_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi) \rangle| \geq |\langle \mathbf{x}_t, \mathbf{p} - \boldsymbol{\kappa}_\phi \rangle| \geq \frac{\varepsilon}{2} \quad (\text{D.3.2})$$

Along the directions in  $S_\phi$  the following is true:

$$|\langle \Pi_{S_\phi} \mathbf{x}_t, \Pi_{S_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi) \rangle| \leq \|\Pi_{S_\phi} \mathbf{x}_t\|_2 \cdot \|\Pi_{S_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi)\|_2 \leq \|\mathbf{x}_t\|_2 \cdot \sqrt{d} \cdot \|\Pi_{S_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi)\|_\infty \leq 1 \cdot \delta \sqrt{d}$$

Using the latter, Equation (D.3.2) now becomes:

$$|\langle \Pi_{L_\phi} \mathbf{x}_t, \Pi_{L_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi) \rangle| \geq \frac{\varepsilon}{2} - \sqrt{d} \cdot \delta \quad (\text{D.3.3})$$

We next focus on upper bounding term  $|\langle \Pi_{L_\phi} \mathbf{x}_t, \Pi_{L_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi) \rangle|$ . By applying the Cauchy-Schwarz inequality, Equation (D.3.3) becomes:

$$\|\Pi_{L_\phi} \mathbf{x}_t\|_2 \|\Pi_{L_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi)\|_2 \geq |\langle \Pi_{L_\phi} \mathbf{x}_t, \Pi_{L_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi) \rangle| \geq \frac{\varepsilon}{2} - \sqrt{d} \cdot \delta \quad (\text{D.3.4})$$

For  $\|\Pi_{L_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi)\|_2$ , observe that  $\mathbf{p}$  and  $\boldsymbol{\kappa}_\phi$  are inside  $\text{Cyl}(\mathcal{K}_\phi, S_\phi)$ , and  $\mathcal{K}_\phi$  has radius at most 1. By the fact that  $\mathbf{p} \in \text{Cyl}(\mathcal{K}_\phi, S_\phi)$  and Definition 7.1, we can write it as  $\mathbf{p} = \mathbf{z} + \sum_{i=1}^{|S_\phi|} y_i \mathbf{s}_i$  where  $\mathbf{s}_i$  form a basis for  $S_\phi$  (which, recall, is orthogonal to  $L_\phi$ ) and  $\mathbf{z} \in \Pi_{L_\phi} \mathcal{K}_\phi$ . Since  $\mathcal{K}_\phi$  is contained in the unit  $\ell_2$  ball, we also have that  $\Pi_{L_\phi} \mathcal{K}_\phi$  is contained in the unit  $\ell_2$  ball. Hence  $\|\Pi_{L_\phi} \mathbf{p}\|_2 = \|\mathbf{z}\|_2 \leq 1$ . The same holds for  $\boldsymbol{\kappa}_\phi$ , and so by the triangle inequality, we have  $\|\Pi_{L_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi)\|_2 \leq 2$ . Hence, from Equation (D.3.4) we get that:  $\|\Pi_{L_\phi} \mathbf{x}_t\| \geq \frac{\varepsilon - 2\sqrt{d}\delta}{4}$ .

Assume for contradiction that there does not exist  $i$  such that  $|\langle \mathbf{e}_i, \mathbf{x}_t \rangle| \geq \frac{\varepsilon - 2\sqrt{d}\delta}{4\sqrt{d}}$ . This means that for all  $j \in [d - |S_\phi|]$  and all  $\{\mathbf{x}_t\}_{t \in [\tau]}$ :  $\langle \mathbf{e}_i, \mathbf{x}_t \rangle < \frac{\varepsilon - 2\sqrt{d}\delta}{4\sqrt{d}}$ . Denoting by  $(E\mathbf{x}_t)_j$  the  $j$ -th coordinate

of  $E\mathbf{x}_t$  we have that  $(E\mathbf{x}_t)_j = \langle \mathbf{e}_j, \mathbf{x}_t \rangle$ . Hence, if  $|\langle \mathbf{x}_t, \mathbf{e}_j \rangle| < \frac{\varepsilon - 2\sqrt{d} \cdot \delta}{4\sqrt{d}}$  then:

$$\|E\mathbf{x}_t\|_2 = \|\Pi_{L_\phi}\mathbf{x}_t\|_2 \leq \sqrt{\sum_{i=1}^d (\langle \mathbf{x}_t, \mathbf{e}_i \rangle)^2} < \sqrt{d \left( \frac{\varepsilon - 2\sqrt{d} \cdot \delta}{4\sqrt{d}} \right)^2} < \frac{\varepsilon - 2\sqrt{d} \cdot \delta}{4}$$

which contradicts the fact that  $\|\Pi_{L_\phi}\mathbf{x}_t\| \geq \frac{\varepsilon - 2\sqrt{d} \cdot \delta}{4}$  established above.  $\blacksquare$

### D.3.2 AUXILIARY LEMMAS FOR PROPOSITION 7.1

**Lemma D.7** (Cap Volume). *With probability at least  $\frac{1}{20\sqrt{d-1}}$ , a point randomly sampled from a ball of radius  $\zeta$  around  $\mathbf{p}^*$ ,  $\mathcal{B}_{L_\phi}(\mathbf{p}^*, \zeta)$ , lies on halfspace:  $\mathbf{H}^+ \left( \mathbf{h}_\phi^*, \langle \mathbf{h}_\phi^*, \mathbf{p}^* \rangle + \frac{\zeta \cdot \ln(3/2)}{\sqrt{d-1}} \right)$ .*

*Proof.* We want to compute the probability that a point randomly sampled from  $\mathcal{B}_{L_\phi}(\mathbf{p}^*, \zeta)$  falls in the following halfspace:

$$\mathbf{H}^+ \equiv \left\{ \mathbf{x} : \langle \mathbf{h}^*, \mathbf{x} - \mathbf{p}^* \rangle \geq \frac{\zeta \cdot \ln(3/2)}{\sqrt{d-1}} \right\}$$

Hence, we want to bound the following probability:  $\Pr[\mathbf{x} \in \mathbf{H}^+ | \mathbf{x} \in \mathcal{B}(\mathbf{p}^*, \zeta)]$ . If we normalize  $\mathcal{B}_{L_\phi}(\mathbf{p}^*, \zeta)$  to be the unit ball  $B$ , then this probability is equal to:

$$\Pr[\mathbf{x} \in \mathbf{H}^+ | \mathbf{x} \in \mathcal{B}_{L_\phi}(\mathbf{p}^*, \zeta)] = \Pr[\mathbf{x} \in \mathbf{H}^1 | \mathbf{x} \in B] = \frac{\text{vol}(B \cap \mathbf{H}^1)}{\text{vol}(B)} \quad (\text{D.3.5})$$

where  $\mathbf{H}^1$  is the halfspace such that  $\mathbf{H}^1 \equiv \left\{ \mathbf{x} : \langle \mathbf{h}^*, \mathbf{x} \rangle \geq \frac{\ln(3/2)}{\sqrt{d-1}} = r \right\}$ , and the last equality is due to the fact that we are sampling uniformly at random.

Similar to the steps in (Blum et al., 2016, Section 2.4.2), in order to compute  $\text{vol}(B \cap \mathbf{H}^1)$  we integrate the incremental volume of a disk with width  $dx_1$ , with its face being a  $(d-1)$ -dimensional ball of radius  $\sqrt{1 - x_1^2}$ . Let  $V(d-1)$  denote the volume of the  $(d-1)$ -dimensional unit ball. Then,

the surface area of the disk is:  $(1 - x_1^2)^{\frac{d-1}{2}} \cdot V(d-1)$ .

$$\begin{aligned}
\text{vol} \left( B \cap \mathbf{H}^1 \right) &= \int_r^1 (1 - x_1^2)^{\frac{d-1}{2}} \cdot V(d-1) dx_1 = V(d-1) \cdot \int_r^1 (1 - x_1^2)^{\frac{d-1}{2}} dx_1 \\
&\quad (V(d-1) \text{ is a constant}) \\
&\geq V(d-1) \cdot \int_r^{\sqrt{\frac{\ln 2}{d-1}}} (1 - x_1^2)^{\frac{d-1}{2}} dx_1 \quad (\sqrt{\frac{\ln 2}{d-1}} < 1, \forall d \geq 2) \\
&\geq V(d-1) \cdot \int_r^{\sqrt{\frac{\ln 2}{d-1}}} \left( e^{-2x_1^2} \right)^{\frac{d-1}{2}} dx_1 \\
&\quad (1 - x^2 \geq e^{-2x^2}, x \in [0, 0.8], \frac{\ln 2}{d-1} \leq 0.8, \forall d \geq 2) \\
&= V(d-1) \cdot \int_r^{\sqrt{\frac{\ln 2}{d-1}}} e^{-x_1^2(d-1)} dx_1 \\
&\geq V(d-1) \cdot \int_r^{\sqrt{\frac{\ln 2}{d-1}}} \sqrt{\frac{d-1}{\ln 2}} \cdot x_1 \cdot e^{-x_1^2(d-1)} dx_1 \quad (x_1 \leq \sqrt{\frac{\ln 2}{d-1}}) \\
&\geq -\frac{V(d-1)}{2\sqrt{(d-1) \cdot \ln 2}} \left[ e^{-(d-1)x^2} \right]_r^{\sqrt{\frac{\ln 2}{d-1}}} \\
&= \frac{V(d-1)}{2\sqrt{(d-1) \cdot \ln 2}} \left( e^{-\ln(3/2)} - e^{-\ln 2} \right) = \frac{V(d-1)}{2\sqrt{(d-1) \cdot \ln 2}} \left( \frac{2}{3} - \frac{1}{2} \right) \\
&= \frac{V(d-1)}{12\sqrt{(d-1) \cdot \ln 2}}
\end{aligned} \tag{D.3.6}$$

Next we show how to upper bound the volume of the unit ball  $B$ . First we compute the volume of one of the ball's hemispheres, denoted by  $\text{vol}(H)$ . Then, the volume of the ball is  $\text{vol}(B) = 2\text{vol}(H)$ . The volume of a hemisphere is *at most* the volume of a cylinder of height 1 and radius 1, i.e.,  $V(d-1) \cdot 1$ . Hence,  $\text{vol}(B) \leq 2V(d-1)$ . Combining this with Equation (D.3.6), Equation (D.3.5) gives the following ratio:

$$\frac{\text{vol} \left( B \cap \mathbf{H}^1 \right)}{\text{vol}(B)} \geq \frac{1}{24\sqrt{(d-1) \cdot \ln 2}} \geq \frac{1}{20\sqrt{d-1}}.$$

This concludes our proof. ■

This lower bound on the probability that a randomly sampled point has the large enough margin that Perceptron requires for efficient convergence, suffices for us to guarantee that after a polynomial number of rounds, such a  $\tilde{\mathbf{q}}$  has been identified in expectation.

**Lemma D.8.** *In expectation, after  $N = 20\sqrt{d-1}$  samples from  $\mathcal{B}_{L_\phi}(\mathbf{p}^*, \zeta)$ , at least one of the samples lies in halfspace  $\mathbf{H}^+ \left( \mathbf{h}_\phi^*, \langle \mathbf{h}_\phi^*, \mathbf{p}^* \rangle + \frac{\zeta \cdot \ln(3/2)}{\sqrt{d-1}} \right)$ .*

*Proof.* From Lemma D.7, the probability that a point randomly sampled from  $\mathcal{B}_{L_\phi}(\mathbf{p}^*, \zeta)$  lies on halfspace  $\mathbf{H}^+ \left( \mathbf{h}_\phi^*, \langle \mathbf{h}_\phi^*, \mathbf{p}^* \rangle + \frac{\zeta \cdot \ln(3/2)}{\sqrt{d-1}} \right)$  is at least  $\frac{1}{20\sqrt{d-1}}$ . Hence, in expectation after  $20\sqrt{d-1}$  samples we have identified one such point by union bound. ■

**Auxiliary Lemmas on Volumetric Progress.** The next lemma states that a convex body  $\mathcal{K}$  with width at least  $\delta$  in every direction must fit a ball of diameter  $\delta/d$  inside it.

**Lemma D.9** (([Lobel et al., 2018](#), Lemma 6.3)). *If  $\mathcal{K} \subset \mathbb{R}^d$  is a convex body such that  $w(\mathcal{K}, \mathbf{u}) \geq \delta$  for every unit vector  $\mathbf{u}$ , then  $\mathcal{K}$  contains a ball of diameter  $\delta/d$ .*

**Lemma D.10** (Directional Grünbaum ([Lobel et al., 2018](#), Theorem 5.3)). *If  $\mathcal{K}$  is a convex body with centroid  $\boldsymbol{\kappa}$ , then, for every unit vector  $\mathbf{u} \neq 0$ , the set  $\mathcal{K}_+ = \mathcal{K} \cap \{x | \langle \mathbf{u}, x - \boldsymbol{\kappa} \rangle \geq 0\}$  satisfies:*

$$\frac{1}{d+1}w(\mathcal{K}, \mathbf{v}) \leq w(\mathcal{K}_+, \mathbf{v}) \leq w(\mathcal{K}, \mathbf{v}), \quad \text{for all unit vectors } \mathbf{v}.$$

The Approximate Grünbaum lemma, which is stated next, relates the volume of a set  $\mathcal{K}_+^\mu = \{x \in \mathcal{K} : \langle \mathbf{u}, x - \boldsymbol{\kappa} \rangle \geq \mu\}$  with the volume of set  $\mathcal{K}$ , when  $\mu \leq 1/d$  for any unit vector  $\mathbf{u}$ . Its proof (provided below) is similar to the proof of ([Lobel et al., 2018](#), Lemma 5.5) with the important difference that  $\mu$  is no longer  $w(\mathcal{K}, \mathbf{u})/(d+1)^2$ , but rather,  $\mu < 1/d$ .

**Lemma D.11** (Approximate Grünbaum). *Let  $\mathcal{K}$  be a convex body and  $\boldsymbol{\kappa}$  be its centroid. For an arbitrary unit vector  $\mathbf{u}$  and a scalar  $\mu$  such that  $0 < \mu < \frac{1}{d}$ , let  $\mathcal{K}_+^\mu = \{x \in \mathcal{K} : \langle \mathbf{u}, x - \boldsymbol{\kappa} \rangle \geq \mu\}$ . Then:  $\text{vol}(\mathcal{K}_+^\mu) \geq \frac{1}{2e^2} \text{vol}(\mathcal{K})$ .*

In order to prove the Appoximate Grünbaum lemma we make use of Brunn's theorem and the Grünbaum Theorem, both stated below.

**Lemma D.12** (Brunn's Theorem). *For convex set  $\mathcal{K}$  if  $g(x)$  is the  $(d-1)$ -dimensional volume of the section  $\mathcal{K} \cap \{\mathbf{y} | \langle \mathbf{y}, \mathbf{e}_i \rangle = x\}$ , then the function  $r(x) = g(x)^{\frac{1}{d-1}}$  is concave in  $x$  over its support.*

**Lemma D.13** (Grünbaum Theorem). *Let  $\mathcal{K}$  denote a convex body and  $\kappa$  its centroid. Given an arbitrary non-zero vector  $\mathbf{u}$ , let  $\mathcal{K}_+ = \{\mathbf{x} | \langle \mathbf{u}, \mathbf{x} - \kappa \rangle \geq 0\}$ . Then:*

$$\frac{1}{e} \text{vol}(\mathcal{K}) \leq \text{vol}(\mathcal{K}_+) \leq \left(1 - \frac{1}{e}\right) \text{vol}(\mathcal{K})$$

*Proof of Lemma D.11.* For this proof we assume without loss of generality that  $\mathbf{u} = e_1$ , and that the projection of  $\mathcal{K}$  onto  $e_1$  is interval  $[a, 1]$ . We are interested in comparing the following two quantities:  $\text{vol}(\mathcal{K})$  and  $\text{vol}(\mathcal{K}_+^\mu)$ . By definition:

$$\text{vol}(\mathcal{K}_+) = \int_0^1 r(x)^{d-1} dx \quad \text{and} \quad \text{vol}(\mathcal{K}_+^\mu) = \int_\mu^1 r(x)^{d-1} dx \quad (\text{D.3.7})$$

where  $r(x) = g(x)^{\frac{1}{d-1}}$  and  $g(x)$  corresponds to the volume of the  $(d-1)$ -dimensional section  $\mathcal{K}_x = \mathcal{K} \cap \{\mathbf{x} | \langle \mathbf{x}, e_i \rangle = x\}$ . We now prove that  $\text{vol}(\mathcal{K}_+^\mu) \geq \frac{1}{e} \text{vol}(\mathcal{K}_+)$ . Combining this with Lemma D.13 gives the result. We denote by  $\rho$  the following ratio:

$$\rho = \frac{\int_\mu^1 r(x)^{d-1} dx}{\int_0^1 r(x)^{d-1} dx} \geq \frac{\int_{1/\delta}^1 r(x)^{d-1} dx}{\int_0^1 r(x)^{d-1} dx} \quad (\text{D.3.8})$$

We approximate function  $r(x)$  with function  $\tilde{r}$ :

$$\tilde{r}(x) = \begin{cases} r(x) & \text{if } 0 \leq x \leq \delta \\ (1-x) \cdot \frac{r(\delta)}{1-\delta} & \text{if } \delta < x \leq 1 \end{cases}$$

Note that since  $0 = \tilde{r}(1) \leq r(1)$  (because  $r(x)$  is a non-negative function) and  $r(x)$  is concave from Brunn's theorem (Lemma D.12), for functions  $r(x)$  and  $\tilde{r}(x)$  it holds that  $r(x) \geq \tilde{r}(x)$ . Using this approximation function  $\tilde{r}(x)$  along with the fact that function  $f(z) = \frac{z}{y+z}$  is *increasing* for any scalar  $y > 0$ , we can relax Equation (D.3.8) as follows:

$$\rho \geq \frac{\int_{1/\delta}^1 \tilde{r}(x)^{d-1} dx}{\int_0^{1/\delta} \tilde{r}(x)^{d-1} dx + \int_{1/\delta}^1 \tilde{r}(x)^{d-1} dx} \quad (\text{D.3.9})$$

Next, we use another approximation function  $\hat{r}(x) = (1-x) \cdot \frac{r(\delta)}{1-\delta}$ ,  $0 \leq x \leq 1$ ; this time in order to approximate function  $\tilde{r}(x)$ . For  $x \in [\delta, 1]$ :  $\tilde{r}(x) = \hat{r}(x)$ . For  $x \in [0, \delta]$  and since  $\tilde{r}(0) = r(0) = 0$  and

$\tilde{r}(x)$  is concave in  $x \in [0, \delta]$ ,  $\hat{r}(x) \geq \tilde{r}(x) = r(x)$ ,  $x \in [0, \delta]$ . Thus, Equation (D.3.9) can be relaxed to:

$$\begin{aligned}\rho &\geq \frac{\int_{1/d}^1 \hat{r}(x)^{d-1} dx}{\int_0^{1/d} \hat{r}(x)^{d-1} dx + \int_{1/d}^1 \hat{r}(x)^{d-1} dx} = \frac{\int_{1/d}^1 (1-x)^{d-1} \cdot \left(\frac{r(\delta)}{1-\delta}\right)^{d-1} dx}{\int_0^1 (1-x)^{d-1} \cdot \left(\frac{r(\delta)}{1-\delta}\right)^{d-1} dx} \\ &= \frac{\int_{1/d}^1 (1-x)^{d-1} dx}{\int_0^1 (1-x)^{d-1} dx} = \frac{-\frac{1}{d} \left(0 - \left(1 - \frac{1}{d}\right)^d\right)}{-\frac{1}{d} (0 - 1)} = \left(1 - \frac{1}{d}\right)^d \geq \frac{1}{2e}\end{aligned}$$

This concludes our proof. ■

We next state the cylindrification lemma (Lobel et al. (2018)), relates the volume of the convex body to the volume of its projection onto a subspace.

**Lemma D.14** (Cylindrification (Lobel et al., 2018, Lemma 6.1)). *Let  $\mathcal{K}$  be a convex body in  $\mathbb{R}^d$  such that  $w(\mathcal{K}, \mathbf{u}) \geq \delta'$  for every unit vector  $\mathbf{u}$ . Then, for every  $(d-1)$ -dimensional subspace  $L$  it holds that  $\text{vol}(\Pi_L \mathcal{K}) \leq \frac{d(d+1)}{\delta'} \text{vol}(\mathcal{K})$ .*

**Lemma D.15** (Epoch Based Projected Grünbaum). *Let  $\mathbf{H}^+(\tilde{\mathbf{h}}_\phi, \tilde{\omega}_\phi)$  be the halfspace returned from CORPV.SEPARATINGCUT. Then, for  $\delta = \frac{\varepsilon}{4(d+\sqrt{d})}$  and  $\mathcal{K}_{\phi+1} = \mathcal{K}_\phi \cap \mathbf{H}^+(\tilde{\mathbf{h}}_\phi, \tilde{\omega}_\phi)$ :*

$$\text{vol}(\Pi_{L_\phi} \mathcal{K}_{\phi+1}) \leq \left(1 - \frac{1}{2e^2}\right) \text{vol}(\Pi_{L_\phi} \mathcal{K}_\phi)$$

*Proof.* By Lemma 7.5, we know that CORPV.SEPARATINGCUT returned hyperplane  $(\tilde{\mathbf{h}}_\phi, \tilde{\omega}_\phi)$  orthogonal to all small dimensions, such that  $\text{dist}(\kappa_\phi^*, (\tilde{\mathbf{h}}_\phi, \tilde{\omega}_\phi)) \leq 3\bar{\nu} = 3 \cdot \frac{\varepsilon - 2\sqrt{d}\delta}{4\sqrt{d}}$ . Substituting  $\delta = \frac{\varepsilon}{4(d+\sqrt{d})}$  we get that:

$$\text{dist}(\kappa_\phi^*, (\tilde{\mathbf{h}}_\phi, \tilde{\omega}_\phi)) \leq \frac{(2\sqrt{d}+1)\varepsilon}{2\sqrt{d}(\sqrt{d}+1)} \leq \frac{1}{d}$$

where the last inequality uses the fact that  $\varepsilon \leq 1/\sqrt{d}$  and that  $\frac{2\sqrt{d}+1}{\sqrt{d}+1} \leq 2$ . Hence, the clause in the approximate Grünbaum lemma (Lemma D.11) holds and as a result, applying the approximate Grünbaum lemma with  $\mathcal{K} = \Pi_{L_\phi} \mathcal{K}_\phi$ , the lemma follows. ■

For completeness, we state the Perceptron mistake bound lemma (Novikoff, 1963).

**Lemma D.16.** Given a dataset  $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i \in [n]}$  with  $\mathbf{x}_i \in \mathbb{R}^d$  and  $y_i \in \{-1, +1\}$ , if  $\|\mathbf{x}_i\| \leq R$  and there exists a linear classifier  $\boldsymbol{\theta}$  such that  $\|\boldsymbol{\theta}\| = 1$  and  $y_i \cdot \langle \boldsymbol{\theta}, \mathbf{x}_i \rangle \geq \gamma$  for a scalar  $\gamma$ . Then, the number of mistakes that the Perceptron algorithm incurs in  $\mathcal{D}$  is upper bounded by  $(R/\gamma)^2$ .

# E

## Appendix for Chapter 9

### E.1 THE PRINCIPAL'S LEARNING PROBLEM

Up until now, we have assumed that the principal has full information on the parameters of the problem. In particular, the principal perfectly knows the underlying linear model  $\mathbf{w}^*$ , the cost matrices  $A_1$  and  $A_2$ , and the projection matrices  $\Pi_1$  and  $\Pi_2$ . In this section, we study how our principal can learn  $\mathbf{w}_{\text{SW}}$  from samples of agents' *modified* features.

To do so, we present two simple building blocks: one that uses a batch of observations to help us estimate  $\mathbf{w}^*$ , and one that, aims to estimate  $\Delta_g(\mathbf{w}) = A_g^{-1}\Pi_g\mathbf{w}$  for a given  $\mathbf{w}$ . We make the following commutativity assumption:

**Assumption E.1.** *For all  $g \in \{1, 2\}$ ,  $\Pi_g A_g^{-1} = A_g^{-1} \Pi_g$ .*

We remark that this assumption holds in several cases of interest. For example, this holds when  $A_g = \sigma_g \mathbb{I}_{d \times d}$  for some  $\sigma_g \geq 0$ , i.e. when the cost of an agent for modifying features is the same across all features and independent across features. This also happens when  $\Pi_g$  and  $A_g^{-1}$  are both diagonal, in which case they are simultaneously diagonalizable hence commute (for example, when  $\Pi_g$  is the projection to a subset of the features, and when manipulating one feature does not affect another feature for free).

Under Assumption E.1, Equation (9.3.7) can be rewritten as:

$$\mathbf{w}_{\text{sw}} = \frac{A_1^{-1}\Pi_1\mathbf{w}^* + A_2^{-1}\Pi_2\mathbf{w}^*}{\|A_1^{-1}\Pi_1\mathbf{w}^* + A_2^{-1}\Pi_2\mathbf{w}^*\|} = \frac{\Delta_1(\mathbf{w}^*) + \Delta_2(\mathbf{w}^*)}{\|\Delta_1(\mathbf{w}^*) + \Delta_2(\mathbf{w}^*)\|},$$

Accurate estimation of both  $\Pi_g\mathbf{w}^*$  and  $\Delta_g(\mathbf{w})$  for any given  $\mathbf{w}$  is sufficient for accurate estimation of  $\mathbf{w}_{\text{sw}}$ . The principal can then take a classical explore-then-exploit approach, in which she first sets aside a batch of agents in group  $g$  to estimate the parameters of the problem to her desired accuracy, then use the parameters she learned to incentivize optimal outcome improvement during the rest of the time horizon.

**Estimating  $\Pi_g\mathbf{w}^*$ .** To estimate  $\Pi_g\mathbf{w}^*$ , we use Algorithm E.1. The algorithm has access to  $n$  agents from group  $g$ . It consists of first posting an initial model of  $\mathbf{w} = 0$  w.l.o.g.<sup>\*</sup>, observing the agents' true, unmodified features and true labels (according to  $\mathbf{w}^*$ ), and using these observations to compute and output an estimate  $\bar{\mathbf{w}}$  of  $\Pi_g\mathbf{w}^*$ :

---

**ALGORITHM E.1: Estimating  $\Pi_g\mathbf{w}^*$**

---

Post  $\mathbf{w} = 0$

For  $i = 1, \dots, n$ , the principal observes agent  $i$ 's true feature vector  $\mathbf{x}_i$ , and his true label  $y_i$

Output  $\bar{\mathbf{w}} \triangleq \arg \min_{\mathbf{w}} \sum_{i=1}^n (\mathbf{x}_i^\top \Pi_g \mathbf{w} - y_i)^2$

---

For simplicity of exposition, we consider the case in which the noise in the label follows a Gaussian distribution, as per the below assumptions. We note however that our results classically extend to the sub-Gaussian case by classical recovery guarantees of linear least-square regression.

<sup>\*</sup>The choice of  $\mathbf{w} = 0$  in Algorithm E.1 is not crucial. In fact, picking any given  $\mathbf{w}$  induces the same distribution of feature vectors as  $\mathcal{S}_g$ , with its expectation shifted by a constant amount of  $\Delta_g(\mathbf{w})$ , after the first  $N_g$  unmodified observations. In turn, given  $N_g + n$  samples, the distribution of the last  $n$  feature vectors used for estimation remains full-rank in subspace  $\mathcal{S}_g$  and still has covariance matrix  $\Sigma_g$ . Therefore, the high-probability bound of Claim E.1 remains the same.

**Assumption E.2.** For every agent  $i$ ,  $y_i - \mathbf{x}_i^\top \mathbf{w}^* \sim \mathcal{N}(0, \sigma^2)$  where  $0 \leq \sigma^2 < \infty$ .

**Claim E.1.** Under Assumption E.2, with probability at least  $1 - \delta$ , the output  $\bar{\mathbf{w}}$  of Algorithm E.1 satisfies

$$\|\bar{\mathbf{w}} - \Pi_g \mathbf{w}^*\|_2 = O\left(\frac{\sigma^2 d \log(1/\delta)}{\lambda_g n}\right),$$

where  $\lambda_g$  denotes the smallest non-zero eigenvalue of  $\Sigma_g$ , the covariance matrix of distribution  $\mathcal{D}_g$ .

*Proof.* Without loss of generality, we restrict attention to the subspace  $\mathcal{S}_g$  induced by the support of the distribution of features  $\mathcal{D}_g$ . Let  $\Sigma_g$  be the covariance matrix of distribution  $\mathcal{D}_g$ ; by definition of  $\mathcal{D}_g$ ,  $\Sigma_g$  is full-rank with smallest eigenvalue  $\lambda_g$  in  $\mathcal{S}_g$ . By the classical recovery results on least-square regression, since  $\mathbb{E}[y_i | \mathbf{x}_i] = \mathbf{x}_i^\top (\Pi_g \mathbf{w}^*)$  by assumption, we obtain that

$$\|\bar{\mathbf{w}} - \Pi_g \mathbf{w}^*\|_2 = O\left(\frac{\sigma^2 d \log(1/\delta)}{\lambda_g n}\right).$$

This concludes the proof. ■

**Estimating  $\Delta_g(\mathbf{w})$ .** Algorithm E.2 has access to  $2n + N_g$  agents from group  $g$ , takes as an input a vector  $\mathbf{w}$ , and outputs an estimate of  $\Delta_g(\mathbf{w})$ .

---

#### ALGORITHM E.2: Estimating $\Delta_g(\mathbf{w})$

---

Post  $\mathbf{w}_1 = 0$

For  $i = 1, \dots, n$ , the principal observes agent  $i$ 's true feature vector  $\mathbf{x}_i$ , and his true label  $y_i$

Post  $\mathbf{w}_2 = \mathbf{w}$

For  $i = n+1, \dots, n+N_g$ , agent  $i$  plays true feature vector  $\mathbf{x}_i$

For  $i = n+N_g+1, \dots, 2n+N_g$ , the principal observes agent  $i$ 's modified feature vector

$\hat{\mathbf{x}}_i = \mathbf{x}_i + \Delta_g(\mathbf{w})$

Output  $\bar{\Delta}_g \triangleq \frac{1}{n} \left( \sum_{i=n+N_g+1}^{2n+N_g} \hat{\mathbf{x}}_i - \sum_{i=1}^n \hat{\mathbf{x}}_i \right)$

---

**Claim E.2.** Let us assume that for all  $i$ ,  $\|\mathbf{x}_i\|_\infty \leq 1$ . Then, with probability at least  $1 - \delta$ , the output  $\bar{\Delta}_g$  of Algorithm E.2 satisfies

$$\|\bar{\Delta}_g - \Delta_g(\mathbf{w})\|_2 \leq \sqrt{\frac{d \log(d/2\delta)}{n}}.$$

*Proof.* First, we note that

$$\begin{aligned}\overline{\Delta_g} &\triangleq \frac{1}{n} \left( \sum_{i=n+N_g+1}^{2n+N_g} \widehat{\mathbf{x}}_i - \sum_{i=1}^n \widehat{\mathbf{x}}_i \right) = \frac{1}{n} \left( \sum_{i=n+N_g+1}^{2n+N_g} \mathbf{x}_i + n\Delta_g(\mathbf{w}) - \sum_{i=1}^n \mathbf{x}_i + \right) \\ &= \Delta_g(\mathbf{w}) + \frac{1}{n} \left( \sum_{i=n+N_g+1}^{2n+N_g} \mathbf{x}_i - \sum_{i=1}^n \mathbf{x}_i + \right).\end{aligned}$$

In turn, we have that

$$\|\overline{\Delta_g} - \Delta_g(\mathbf{w})\|_2 = \frac{1}{n} \left\| \sum_{i=n+N_g+1}^{2n+N_g} \mathbf{x}_i - \sum_{i=1}^n \mathbf{x}_i \right\|_2.$$

We have that

$$\sum_{i=n+N_g+1}^{2n+N_g} \mathbf{x}_i - \sum_{i=1}^n \mathbf{x}_i = \sum_{i=1}^n Z_i$$

where  $Z_i = \mathbf{x}_{i+n+N_g} - \mathbf{x}_i$ . In turn,  $Z_i$  is a random vector with mean  $\mathbb{E}[Z_i] = 0$  and covariance matrix  $2\Sigma_g$ , noting that  $\mathbf{x}_i$  and  $\mathbf{x}_{i+n+N_g}$  are drawn independently. Further,  $|Z_i(k)| \leq 2$ . By Hoeffding's inequality, we have that with probability at least  $1 - \frac{\delta}{d}$ , for a given  $k \in [d]$ ,

$$\left| \sum_{i=1}^n Z_i(k) \right| \leq \sqrt{2n \log(d/2\delta)}.$$

By union bound, we have that with probability at least  $1 - \delta$ , this holds simultaneously for all  $k \in [d]$ , with directly yields

$$\left\| \sum_{i=1}^n Z_i \right\| = \sqrt{\sum_{k=1}^d \left( \sum_{i=1}^n Z_i(k) \right)^2} \leq \sqrt{2dn \log(d/2\delta)}.$$

This immediately leads to the result. ■

## E.2 SUPPLEMENTARY MATERIAL FOR SECTION 9.3

**Lemma E.1.** Let  $Q \in \mathbb{R}^{d \times d}$  a symmetric PD matrix and  $c$  a vector in  $\mathbb{R}^d$ . Then, the optimization problem:

$$\begin{aligned} & \max_{x \in \mathbb{R}^d} c^\top x \\ & \text{s.t., } x^\top Q x \leq b \end{aligned}$$

has unique solution:

$$x = \frac{b Q^{-1} c}{\sqrt{c^\top Q^{-1} c}}$$

*Proof.* We first compute the Lagrangian:

$$L(x, \lambda) = -c^\top x + \frac{\lambda}{2} (x^\top Q x - b) \quad (\text{E.2.1})$$

We can then find the KKT conditions:

$$-c + \lambda Q x = 0 \quad (\text{E.2.2})$$

$$\lambda \geq 0 \quad (\text{E.2.3})$$

$$\lambda (x^\top Q x - b) = 0 \quad (\text{E.2.4})$$

$$x^\top Q x \leq b \quad (\text{E.2.5})$$

At maximum it must be the case that  $\lambda > 0$  (from Eq. (E.2.3)) and hence, combining Eqs. (E.2.5) and (E.2.4) we get  $x^\top Q x = b$ . Due to the fact that  $\lambda > 0$ , then from Eq. (E.2.2), solving in terms of  $x$  and using the fact that  $Q$  is symmetric positive definite we get:

$$x = \frac{1}{\lambda} Q^{-1} c \quad (\text{E.2.6})$$

Substituting the above in equation  $x^\top Q x = b$  we obtain:

$$x^\top Q x = \frac{1}{\lambda^2} c^\top Q^{-1} c = b \quad (\text{E.2.7})$$

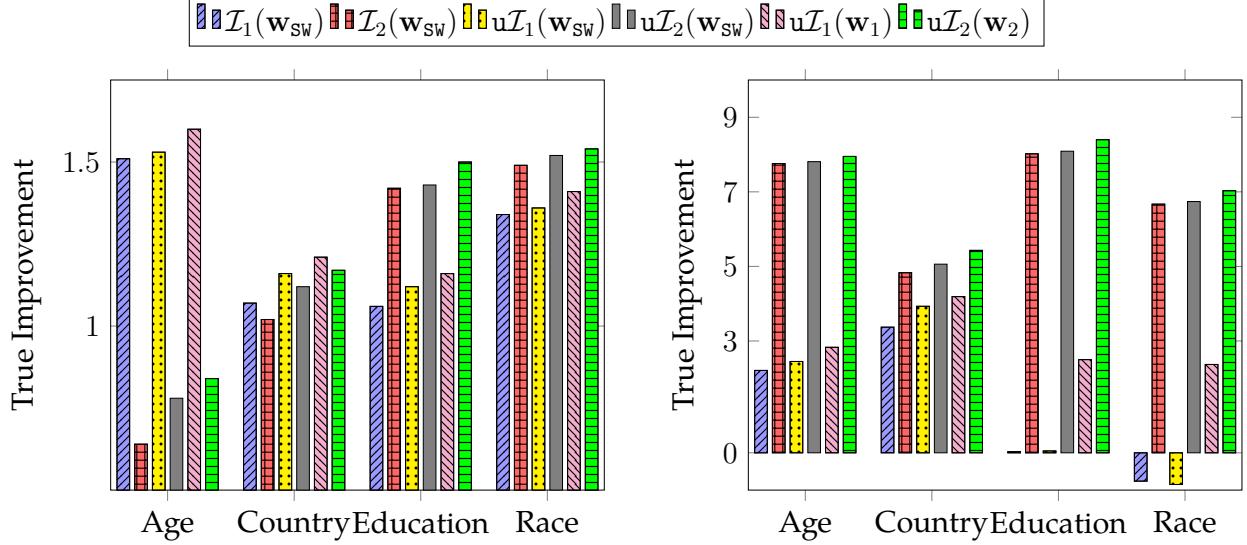


Figure E.1: Left, Right: evaluation on ADULT dataset when  $w^*$  is drawn uniformly at random and when  $A_g$ 's are drawn uniformly at random, respectively. The breakdown in subgroups  $G_1, G_2$  is the same as in Table 9.2. Recall that  $\mathcal{I}_g(w_{SW})$ ,  $u\mathcal{I}_g(w_{SW})$ ,  $u\mathcal{I}_g(w_g)$ , denote the total, the per-unit, and the optimal per-unit improvement for subgroup  $g$  in equilibrium, respectively.

Solving this in terms of  $\lambda$  gives  $\lambda = \frac{1}{b} \sqrt{c^\top Q^{-1} c}$ . Substituting  $\lambda$  in Eq. (E.2.6) we get the result. The proof is completed by the fact that the objective function is convex and the feasible set is concave; hence the global optimum is found at a KKT point. ■

### E.3 SUPPLEMENTARY MATERIAL FOR SECTION 9.5

In this section, we study the impact of disparities in access to information about the model not just on their own, but in conjunction with cost disparities and asymmetries of the scoring rule. To do so, we provide additional experimental results on both the ADULT and the TAIWAN-CREDIT dataset.

In Figures E.1 and E.2, we study disparities in improvement on the ADULT and the TAIWAN-CREDIT datasets respectively. While we keep using the same data  $X_g$  and projection matrices  $\Pi_g$  as in Section 9.5, we now consider non-identity cost matrices, and a non-symmetric  $w^*$ . To do so, we both draw  $A_g$  and  $w^*$  uniformly at random; for  $A_g$ 's the uniform distribution is taken over  $[-1, 1]$ , while for  $w^*$  it is taken over  $[0, 1]$ , in both cases coefficient by coefficient.

We first note that the scale of the true improvements may differ from those of Figure 9.1; indeed, this happens in Figure E.1, and is significantly more pronounced in Figure E.2. This comes

from the fact that changing the value and magnitude of both  $w^*$  and the  $A_g$ 's may lead to different magnitudes of improvements. More importantly, we remark that the presence of disparities in the cost matrices and asymmetries in the most accurate rule  $w^*$  *exacerbate* the disparities in outcomes across subgroups. This is particularly true of the “Education” feature on the right plot in Figure E.1. There, we observe that while the optimal per-unit improvement for subgroup 1 remains around 2 and seems unaffected by the new cost and  $w^*$ , the improvements for this subgroup when optimizing the social welfare across subgroups are significantly lower. In fact, because both the total and per-unit improvements are low, it seems that the disparities we observe are not only due to the fact that the learner may be putting less weight on subgroup 1 is spent on directions in which both subgroups have information, but that are only “good” and easy to modify for subgroup 2.

We further observe that the addition of non-identity cost matrices and non-symmetric  $w^*$  can lead to a *degradation* of outcomes in one of the subgroups, when the principal optimizes over the joint social welfare. This is most visible on the right plot in Figure E.1, where the total and per-unit improvements for subgroup 1 are negative. This matches the relatively counter-intuitive observation of Section 9.4.1 that optimizing for the social welfare of both subgroups may hurt the welfare of one of them.

Finally, by comparing the results across age groups in the left and right plot, we observe a significant reversal of the disparities of improvements across groups, with group 2 having significantly worse improvements in the first plot and group 1 significantly worse improvements in the second plot. This paints a *nuanced* picture that shows that the amount of information that a group has about the scoring rule used by the principal is not the only factor of importance. While having more information is important, how this information interacts with the true model  $w^*$  and the strategic behavior of the agents matters; having a lot of information in directions that have little effect on an agents' true label, or in directions that are very costly for some agents to modify, does not help them when it comes to improving their true labels.

#### E.4 GENERALIZING TO MULTIPLE SUBGROUPS

Let  $\mathcal{G}$  denote the set of all subgroups, i.e.,  $\mathcal{G} = \{1, 2, \dots, m\}$ . As is customary in the literature, we use  $\mathcal{G}_{-j}$  to denote all the subgroups apart from subgroup  $j$ , i.e.,  $\mathcal{G}_{-j} = \{1, 2, \dots, j-1, j+1, \dots, m\}$ .

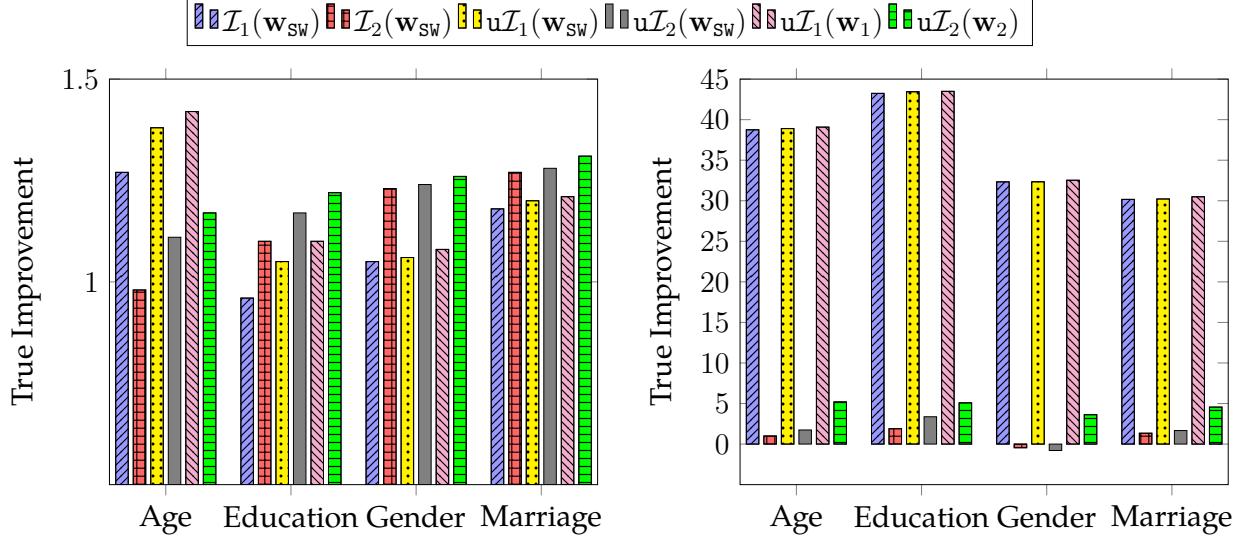


Figure E.2: Left, Right: evaluation on TAIWAN-CREDIT dataset when  $\mathbf{w}^*$  is drawn uniformly at random and when  $A_g$ 's are drawn uniformly at random, respectively. The breakdown in subgroups  $G_1, G_2$  is the same as in Table 9.2. Recall that  $\mathcal{I}_g(\mathbf{w}_{\text{SW}})$ ,  $u\mathcal{I}_g(\mathbf{w}_{\text{SW}})$ ,  $u\mathcal{I}_g(\mathbf{w}_g)$ , denote the total, the per-unit, and the optimal per-unit improvement for subgroup  $g$  in equilibrium, respectively.

In order to explain how the theorem and proposition statements change when  $m > 2$ , we first outline how the principal's equilibrium rule changes as a result of the presence of  $m > 2$  subgroups. Due to the fact that the estimated rule for each group  $g \in \mathcal{G}$  is:  $\mathbf{w}_{\text{est}}(g) = \Pi_g \mathbf{w}$ , then from extending Lemma 9.4 we have that the principal's equilibrium rule becomes:

$$\mathbf{w}_{\text{sw}} = \frac{(\Pi_1 A_1^{-1} + \dots + \Pi_m A_m^{-1}) \mathbf{w}^*}{\| (\Pi_1 A_1^{-1} + \dots + \Pi_m A_m^{-1}) \mathbf{w}^* \|} \quad (\text{E.4.1})$$

We first analyze the do-no-harm objective for the case that  $m > 2$  subgroups are present in the population. The analogue of Theorem 9.1 for  $m > 2$  subgroups follows.

### Theorem E.1

In equilibrium, there is no negative externality for subgroup  $g$  and any  $\mathbf{w}^*$  if and only if for all  $g \in \mathcal{G}$ , the matrix  $(\sum_{i \in \mathcal{G}} A_i^{-1} \Pi_i) \Pi_g A_g^{-1} + A_g^{-1} \Pi_g (\sum_{i \in \mathcal{G}} A_i^{-1} \Pi_i)$  is PSD.

This means that we can still guarantee that there is no negative externality for any of the subgroups in equilibrium in the two cases of interest, namely:

1. when the cost matrices are proportional to each other, i.e.,  $A_i = c_{ij} \cdot A_j$  for all  $(i, j) \in \mathcal{G}^2$  and

some scalars  $c_{ij} > 0$  (analogue of Proposition 9.1 for  $m > 2$  subgroups).

2. when the subspaces  $\mathcal{S}_1, \dots, \mathcal{S}_m$  are orthogonal (analogue of Proposition 9.2 for  $m > 2$ ).

To derive the aforementioned results, the only change in the proofs of Theorem 9.1, and Propositions 9.1 and 9.2 is that  $\mathbf{w}_{\text{SW}}$  should be substituted with the expression in Equation (E.4.1).

We proceed to discussing total improvement for  $m > 2$  subgroups. We find it useful to present a slightly generalized version of the *overlap proxy*.

**Definition E.1.** Given a scoring rule  $\mathbf{w} \in \mathbb{R}^d$  and projections  $\Pi_1, \dots, \Pi_m \in \mathbb{R}^d$ , we define the overlap proxy between any two groups  $G_i, G_k$  with respect to  $\mathbf{w}$  to be:  $r_{i,k}(\mathbf{w}) \triangleq \|\Pi_i \mathbf{w} - \Pi_k \mathbf{w}\|$ .

Using this definition, we can state the direct generalization of Lemma 9.6.

**Lemma E.2.** Let  $\text{diff}_{j,k} \triangleq |\mathcal{I}_j(\mathbf{w}) - \mathcal{I}_k(\mathbf{w})|$  be the disparity in total improvement across subgroups when the principal's rule is  $\mathbf{w}$ . In equilibrium, if  $A_j = A_k = \mathbb{I}_{d \times d}$ , then:  $\text{diff}_{j,k}(\mathbf{w}_{\text{SW}}) \leq r_{j,k}(\mathbf{w}^*)$ . Further, the equality holds if and only if  $\Pi_j \mathbf{w}^*$  and  $\Pi_k \mathbf{w}^*$  are co-linear.

The analogue of Theorem 9.3 for  $m > 2$  becomes:

### Theorem E.2

In equilibrium, the groups obtain equal total improvement for all  $\mathbf{w}^*$  if and only if  $A_1^{-1} \Pi_1 A_1^{-1} = A_2^{-1} \Pi_2 A_2^{-1} = \dots = A_m^{-1} \Pi_m A_m^{-1}$ .

Finally, we turn our attention to the per-unit improvement, and we state the analogue of Theorem 9.4 for  $m > 2$  subgroups. This analogue is again derived using Equation (E.4.1) for  $\mathbf{w}_{\text{SW}}$ .

### Theorem E.3

In equilibrium, subgroup  $g$  gets optimal per-unit improvement if and only if:

$$\left\langle A_g^{-1} \frac{\Pi_g A_g^{-1} \mathbf{w}^*}{\|\Pi_g A_g^{-1} \mathbf{w}^*\|_2} - A_g^{-1} \frac{\Pi_g (\Pi_1 A_1^{-1} + \Pi_2 A_2^{-1} + \dots + \Pi_m A_m^{-1}) \mathbf{w}^*}{\|\Pi_g (\Pi_1 A_1^{-1} + \Pi_2 A_2^{-1} + \dots + \Pi_m A_m^{-1}) \mathbf{w}^*\|_2}, \mathbf{w}^* \right\rangle = 0.$$

Note that this means that, in equilibrium, optimal per-unit outcome improvement is guaranteed if there exists  $c_g > 0$ , such that:

$$\Pi_g (A_g^{-1} \Pi_g)^\top \mathbf{w}^* = c_g \Pi_g (A_1^{-1} \Pi_1 + \dots + A_m^{-1} \Pi_m)^\top \mathbf{w}^*$$

Two notable examples for which this condition holds are:

1. when all of  $\mathcal{S}_1, \dots, \mathcal{S}_m$  are orthogonal to each other
2. when  $A_i = c_{ij} \cdot A_j$  and  $\Pi_i = \Pi_j$ .

# F

## Appendix for Chapter 10

### F.1 NOTES ON CHAPTER 10.4.1

If one is interested in optimizing the *sum* of utilities at each iteration rather than the *average*, then if all iterations have the same number of batches  $|I|$ , this simply amounts to rescaling everything by  $|I|$ , which would lead to an  $|I|$  blow up in the regret.

If different periods have different number of batches and  $I_{\max}$  is the maximum number of batches per iteration, then we can always pad the extra batches with all zero rewards. This would amount to again multiplying the regret by  $I_{\max}$  and would change the unbiased estimates at each period

to be scaled by the number of iterations in that period:

$$\tilde{u}_t(b) = \frac{|I_t|}{I_{\max}} \sum_{o \in O} \frac{\Pr_t[o|b] \cdot \Pr_t[o|b_t]}{\Pr_t[o]} (Q_t(b, o) - 1) \quad (\text{F.1.1})$$

and then we would invoke the same algorithm. This essentially puts more weight on iterations with more auctions, so that the "step-size" of the algorithm depends on how many auctions were run during that period. It is easy to see that the latter modification would lead to regret  $4I_{\max}\sqrt{T \log(|B|)}$  in the sponsored search auction application.

## F.2 STANDARD PROOF FOR THE REGRET OF EXPONENTIAL WEIGHTS UPDATE

**Lemma F.1.** *The exponential weights update with an estimate  $\tilde{u}_t(\cdot) \leq 0$  such that for any  $b \in B$  and  $t$ ,  $|\mathbb{E}[\tilde{u}_t(b)] - (u_t(b) - 1)| \leq \kappa$ , achieves expected regret on the form:*

$$R(T) \leq \frac{\eta}{2} \sum_{t=1}^T \sum_{b \in B} \pi_t(b) \cdot \mathbb{E}[\tilde{u}_t(b)^2] + \frac{1}{\eta} \log(|B|) + 2\kappa T$$

*Proof.* Let  $b^* = \arg \max_{b \in B} \mathbb{E} \left[ \sum_{t=1}^T u_t(b) \right]$ . Following the standard analysis of the exponential weights update algorithm Arora et al. (2012) and the fact that  $\forall x \leq 0, e^x \leq 1 + x + \frac{x^2}{2}$ , we have:

$$\begin{aligned} \mathbb{E} \left[ \sum_{t=1}^T \tilde{u}_t(b^*) \right] &\leq \sum_{t=1}^T \sum_{b \in B} \pi_t(b) \mathbb{E}[\tilde{u}_t(b)] + \frac{\eta}{2} \sum_{t=1}^T \sum_{b \in B} \pi_t(b) \cdot \mathbb{E}[\tilde{u}_t(b)^2] + \frac{1}{\eta} \log(|B|) \\ &\leq \sum_{t=1}^T \sum_{b \in B} \pi_t(b)(u_t(b) - 1 + \kappa) + \frac{\eta}{2} \sum_{t=1}^T \sum_{b \in B} \pi_t(b) \cdot \mathbb{E}[\tilde{u}_t(b)^2] + \frac{1}{\eta} \log(|B|) \\ &= \mathbb{E} \left[ \sum_{t=1}^T u_t(b_t) \right] + \frac{\eta}{2} \sum_{t=1}^T \sum_{b \in B} \pi_t(b) \cdot \mathbb{E}[\tilde{u}_t(b)^2] + \frac{1}{\eta} \log(|B|) + \kappa T - T \end{aligned}$$

which implies that

$$\begin{aligned} R(T) &= \mathbb{E} \left[ \sum_{t=1}^T u_t(b^*) \right] - \mathbb{E} \left[ \sum_{t=1}^T u_t(b_t) \right] \leq \mathbb{E} \left[ \sum_{t=1}^T \tilde{u}_t(b^*) \right] - \mathbb{E} \left[ \sum_{t=1}^T u_t(b_t) \right] + \kappa T + T \\ &\leq \frac{\eta}{2} \sum_{t=1}^T \sum_{b \in B} \pi_t(b) \cdot \mathbb{E}[\tilde{u}_t(b)^2] + \frac{1}{\eta} \log(|B|) + 2\kappa T \end{aligned}$$

■

REMARK. Let the estimator  $\tilde{u}_t(b)$  be unbiased for any  $t$  and any  $b \in B$ , then the expected regret is

$$R(T) \leq \frac{\eta}{2} \sum_{t=1}^T \sum_{b \in B} \pi_t(b) \cdot \mathbb{E} [\tilde{u}_t(b)^2] + \frac{1}{\eta} \log(|B|)$$

# G

## Appendix for Chapter 11

### G.1 GENERALIZATION FOR UNKNOWN REPLENISHING ARM

In this section, we show how the algorithm and the analysis for general  $\lambda$  changes once the replenishing arm is not known or has baseline reward that is not within  $[1 - 2\epsilon, 1 - \epsilon]$ .

**Lemma G.1.** *Any instance of  $K$  arms with reward tuples  $(r_i, b_i)_{i \in [K]}$  and an initial state  $q_0$  is equivalent to an instance with reward tuples  $(r'_i, b'_i) = (cr_i, b_i/c)$ ,  $\forall i \in [K]$  and initial state  $q'_0 = q_0/c$  for  $c > 0$ .*

*Proof.* To see this, note that the expected reward picked up after  $T$  rounds by a sequence of actions

$\{I_t\}_{t \in [T]}$  is equal to:

$$\begin{aligned}
\sum_{t \in [T]} q_t r_{I_t} &= \sum_{t \in [T]} (1 - \lambda)^t q_0 r_{I_t} + \lambda \sum_{t \in [T]} \sum_{s \in [t-1]} (1 - \lambda)^{t-1-s} b_{I_s} r_{I_t} \\
&= \sum_{t \in [T]} (1 - \lambda)^t \frac{q_0}{c} r_{I_t} c + \lambda \sum_{t \in [T]} \sum_{s \in [t-1]} (1 - \lambda)^{t-1-s} \frac{b_{I_s}}{c} r_{I_t} c \\
&= \sum_{t \in [T]} (1 - \lambda)^t q'_0 r'_{I_t} + \lambda \sum_{t \in [T]} \sum_{s \in [t-1]} (1 - \lambda)^{t-1-s} b'_{I_s} r'_{I_t}
\end{aligned}
\tag{Lemma 11.1}$$

This concludes our proof. ■

Next, we show how to choose  $c$  in order to guarantee that there exists an arm whose baseline reward is inside  $[1 - 2\varepsilon, 1 - \varepsilon]$ . This is the “replenishing” arm in the general case.

**Lemma G.2.** *Let  $i^* = \arg \max_{i \in [K]} b_i$ . Then, for any  $\varepsilon > 0$  choosing  $c = b_{i^*} + \frac{\varepsilon \cdot b_{i^*}}{1 - \varepsilon}$  guarantees that  $b'_{i^*} \in [1 - 2\varepsilon, 1 - \varepsilon]$ .*

*Proof.* For the lower bound:

$$b'_{i^*} = \frac{b_{i^*}(1 - \varepsilon)}{b_{i^*}(1 + \varepsilon)} = \frac{1 - \varepsilon}{1 + \varepsilon} > 1 - 2\varepsilon \Leftrightarrow 1 - \varepsilon > 1 + \varepsilon - 2\varepsilon - 2\varepsilon^2 \Leftrightarrow 0 > -2\varepsilon^2$$

which is true. For the upper bound:

$$b'_{i^*} = \frac{b_{i^*}(1 - \varepsilon)}{b_{i^*}(1 + \varepsilon)} = \frac{1 - \varepsilon}{1 + \varepsilon} < 1 - \varepsilon$$
■

Moving forward we assume without loss of generality that our instance includes a replenishing arm, i.e., that there exists  $i_R \in [K]$  such that  $b_{i_R} \in [1 - 2\varepsilon, 1 - \varepsilon]$ . Note that this is indeed without loss of generality because of Lemmas G.1 and G.2. In this section, we prove the following guarantee regarding the regret incurred in the case of an unknown replenishing arm.

### Theorem G.1

Tuning  $\delta = 2\varepsilon$ ,  $M = K^2 \ln(T)/\varepsilon^2$  and

$$\varepsilon = \left( \frac{K \cdot \ln(T) \cdot \log(\lambda)}{T \cdot \log(1 - \lambda)} \right)^{1/3}$$

Algorithm G.1 incurs regret  $\text{REGRET}(T) = \mathcal{O} \left( \left( \frac{K \ln(T) \log(\lambda)}{\log(1 - \lambda)} \right)^{1/3} T^{2/3} \right)$

Let us define  $\bar{b}$  to be  $\bar{b} = \sum_{i \in [K]} b_i / K$ . Based on Lemma G.2, and the fact that  $b_i \geq 0, \forall i \in [K]$ , it holds that  $\bar{b} \geq (1 - \varepsilon)/K$ . This will be useful in our analysis below.

We first present the algorithm that achieves the desired regret guarantee for the case of an unknown replenishing arm.

---

#### Algorithm G.1: MAB Long-Term Effects with Unknown $i_R$

---

Set  $\varepsilon, \delta, M$  as stated in Theorem G.1.

Initialize rounds  $t = 1$ .

```

/* Explore in-the-void rewards and build their estimators: { $\hat{r}_i$ } $_{i \in [K]}$  */
```

3 **for** arm  $i \in [K]$  **do**

4     Initialize reward estimate  $\hat{r}_i = 0$ .

5     **for** blocks  $j \in [M]$  **do** // Restore the state to at least  $b_z - \varepsilon$

6         Choose an arm  $z \in [K]$  uniformly at random.     //  $z = \text{benchmark arm for state.}$

7         **for** pulls  $1, \dots, N(\lambda)$  **do**

8             Play arm  $z$ .

9             Update  $t \leftarrow t + 1$ .

10         Play arm  $i$ , observe reward  $R_j^i$ , and update:  $\hat{r}_i \leftarrow \hat{r}_i + \frac{R_j^i}{M}$ .     // Play  $i$  when  $q \approx b_z - \varepsilon$ .

11         Update  $t \leftarrow t + 1$ .

```

/* Explore baseline rewards and build estimators: { $\hat{b}_i$ } $_{i \in [K]}$  */
```

12 **for** arm  $i \in [K]$  **do**

13     Initialize state estimator  $\hat{v}_i = 0$ .

14     **for** blocks  $j \in [M]$  **do**

15         **for** pulls  $1, \dots, N(\lambda)$  **do**

16             Play arm  $i$ .

17             Update  $t \leftarrow t + 1$ .

18         Play arm  $i$ , observe reward  $S_j^i$ , and update:  $\hat{v}_i \leftarrow \hat{v}_i + \frac{S_j^i}{M}$ .     // Play  $i$  when  $q \approx b_i$

19         Compute baseline reward estimator:  $\hat{b}_i = \hat{v}_i / \hat{r}_i$ .

20 Feed  $(\hat{r}_i, \hat{b}_i)$  in the Dynamic Programming algorithm and play the solution until the end of  $T$ .

---

Our analysis follows a similar route as for the case of Theorem 11.1. Importantly, Lemma 11.5 remains unchanged and still holds verbatim. What changes is the lemma with the estimator  $\hat{r}_i, \forall i \in$

$[K]$  because now we have sampled uniformly at random a benchmark arm, rather than using the known replenishing arm.

**Lemma G.3.** *Let  $\bar{b} = \frac{1}{K} \sum_{i \in [K]} b_i$ . Then, for the in-the-void reward estimator of each arm in Line 10 of Algorithm G.1 and any scalar  $\delta$ , it holds that:*

$$\Pr [|\hat{r}_i - \bar{b} \cdot r_i| \geq \delta] \leq 2 \exp (-2M \cdot (\delta - \varepsilon)^2),$$

*Proof.* From Hoeffding's inequality on  $\hat{r}_i$ , we have that:

$$\Pr [|\hat{r}_i - \mathbb{E}[\hat{r}_i]| \geq \delta] \leq 2 \exp (-2M\delta^2) \quad (\text{G.1.1})$$

From Lemma 11.3, regardless of the starting state and the prior history, if an arm  $z$  is played repeatedly for  $N(\lambda)$  rounds, then at round  $t_j^i$  the system's state is at  $q_{t_j^i} \geq b_z - \varepsilon$ . So (by definition of our setting) and conditioning on event  $\mathcal{E}_z = \{\text{arm } z \text{ is chosen as benchmark}\}$  the expected reward at the right next round (i.e., Line 10 of Algorithm G.1) is

$$\mathbb{E}[R_j^i | \mathcal{E}_z] = q_{t_j^i} \cdot r_i \in [(b_z - \varepsilon) \cdot r_i, (b_z + \varepsilon) \cdot r_i],$$

This means that in expectation over the choice of  $z$  (which happens uniformly at random) we have:

$$\mathbb{E}[R_j^i] = \mathbb{E}[q_{t_j^i}] \cdot r_i \in [(\bar{b} - \varepsilon) \cdot r_i, (\bar{b} + \varepsilon) \cdot r_i]$$

As a result, by the linearity of expectation and using the definition of  $\hat{r}_i$ :

$$\mathbb{E}[\hat{r}_i] = \frac{\mathbb{E}[R_j^i]}{M} = \frac{\sum_{j \in [M]} q_{t_j^i} \cdot r_i}{M} = r_i \cdot \frac{\sum_{j \in [M]} q_{t_j^i}}{M} \Rightarrow \mathbb{E}[\hat{r}_i] \in [r_i \cdot (b_z - \varepsilon), r_i \cdot (b_z + \varepsilon)]$$

From Equation (G.1.1), we have that:

$$\begin{aligned}
2 \exp(-2M\delta^2) &\geq \Pr[\hat{r}_i - \mathbb{E}[\hat{r}_i] \geq \delta \text{ or } \hat{r}_i - \mathbb{E}[\hat{r}_i] \leq -\delta] \\
&\geq \Pr[\hat{r}_i \geq \bar{b} \cdot r_i + \varepsilon + \delta \text{ or } \hat{r}_i - \mathbb{E}[\hat{r}_i] \leq -\delta] & (\mathbb{E}[\hat{r}_i] \leq r_i \cdot b_z + \varepsilon) \\
&\geq \Pr[\hat{r}_i \geq \bar{b}r_i + \varepsilon + \delta \text{ or } \hat{r}_i \leq \bar{b}r_i - \varepsilon - \delta] & (\mathbb{E}[\hat{r}_i] \geq b_z \cdot r_i - \varepsilon) \\
&= \Pr[|\hat{r}_i - \bar{b}r_i| \geq \delta + \varepsilon]
\end{aligned}$$

Using as  $\delta' = \delta + \varepsilon$  in the latter gives the result. ■

Next, we show that the  $\hat{b}_i$  estimators that are built from the second part of the algorithm are good estimators, despite no assumptions on  $i_R$ .

**Lemma G.4.** *Let  $\bar{b} = \sum_z b_z / K$ . Then, for the baseline reward estimators of each arm  $i$  in Line 19 of Algorithm G.1 and any scalar  $\delta \geq 2\varepsilon$ , it holds that:*

$$\Pr\left[\left|\hat{b}_i - \frac{b_i}{\bar{b}}\right| \geq \delta\right] \leq 8 \exp(-2M \cdot (\varepsilon - \delta)^2)$$

*Proof.* We follow the steps of the proof of Lemma 11.6. Fix an arm  $i \in [K]$  and let us use  $e_v$  and  $e_r$  to denote the following quantities:  $e_v = \hat{v}_i - v_i$  and  $e_r = \hat{r}_i - \bar{b} \cdot r_i$  respectively. Then, we have that:

$$\begin{aligned}
\Pr\left[\left|\frac{\hat{v}_i}{\hat{r}_i} - \frac{v_i}{\bar{b} \cdot r_i}\right| \geq \delta\right] &= \Pr\left[\left|\frac{v_i + e_v}{\bar{b} \cdot r_i + e_r} - \frac{v_i}{\bar{b} \cdot r_i}\right| \geq \delta\right] \\
&= \Pr\left[\left|\frac{\bar{b}r_i e_v - e_r v_i}{\bar{b}r_i(\bar{b}r_i + e_r)}\right| \geq \delta\right] \\
&\leq \Pr\left[\left|\frac{e_v}{\bar{b}r_i + e_r}\right| + \left|b_i \frac{e_r}{\bar{b}r_i + e_r}\right| \geq \delta\right] \\
&\leq \underbrace{\Pr\left[\left|\frac{e_v}{\bar{b}r_i + e_r}\right| \geq \delta/2\right]}_{Q_1} + \underbrace{\Pr\left[b_i \cdot \left|\frac{e_r}{\bar{b} \cdot (\bar{b}r_i + e_r)}\right| \geq \delta/2\right]}_{Q_2} \tag{G.1.2}
\end{aligned}$$

where the first inequality is due to the triangle inequality and the fact that  $\Pr[a < c] \leq \Pr[b < c]$  for  $a \leq b$ , and the second inequality is due to the fact that when  $a + b \geq c$ , then  $\Pr[a + b \geq c] \leq \Pr[a \geq c/2] + \Pr[b \geq c/2]$ .

To upper bound  $Q_1$  and  $Q_2$ , we condition on the following event:  $\mathcal{E}'_i = \{|e_r| \leq \delta\}$ . Note that the probability with which the complement  $\mathcal{E}_i$  happens is given by Lemma G.3 and is:

$$\Pr[\mathcal{E}_i] \geq 2 \exp(-2M \cdot (\delta - \varepsilon)^2) \quad (\text{G.1.3})$$

Rewriting  $Q_1$ :

$$Q_1 = \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot |\bar{b}r_i + e_r| \right] \leq \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot \left| |\bar{b}r_i| - |e_r| \right| \right] \quad (\text{G.1.4})$$

Conditioning on  $\mathcal{E}'_i$  we get:

$$\begin{aligned} \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot \left| |\bar{b}r_i| - |e_r| \right| \middle| \mathcal{E}'_i \right] &\leq \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot |\bar{b}r_i - \delta| \right] \\ &\leq 2 \exp \left( -2M \cdot \left( \frac{\delta}{2} \cdot |\bar{b}r_i - \delta| - \varepsilon \right)^2 \right) \quad (\text{Lemma 11.5}) \\ &\leq 2 \exp(-2M \cdot (\varepsilon^2 - \varepsilon\delta)) \end{aligned} \quad (\text{G.1.5})$$

where the last inequality is due to the fact that  $|\bar{b}r_i - \delta| \leq 1$ . From the law of total probability:

$$\begin{aligned} Q_1 &= \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot |\bar{b}r_i + e_r| \middle| \mathcal{E}'_i \right] \cdot \Pr[\mathcal{E}'_i] + \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot |\bar{b}r_i + e_r| \middle| \mathcal{E}_i \right] \cdot \Pr[\mathcal{E}_i] \\ &\leq \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot \left| |\bar{b}r_i| - |e_r| \right| \middle| \mathcal{E}'_i \right] \cdot \Pr[\mathcal{E}'_i] + \Pr \left[ |e_v| \geq \frac{\delta}{2} \cdot \left| |\bar{b}r_i| - |e_r| \right| \middle| \mathcal{E}_i \right] \cdot \Pr[\mathcal{E}_i] \\ &\leq 2 \exp(M \cdot (\varepsilon^2 - \delta)) \cdot 1 + 1 \cdot 2 \exp(-2M \cdot (\delta - \varepsilon)^2) \end{aligned}$$

where the first inequality is due to Eq. (G.1.4) and the last one is due to Eqs. (G.1.3), (G.1.5).

We now turn our attention to  $Q_2$ :

$$Q_2 \leq \Pr \left[ |e_r| \geq \frac{\delta}{2(K - \varepsilon)} \cdot |\bar{b}r_i + e_r| \right] \leq \Pr \left[ |e_r| \geq \frac{\delta}{2K} \cdot |\bar{b}r_i + e_r| \right]$$

where the first inequality is due to the fact that  $\bar{b} \geq 1/K$ . Using exactly the same reasoning as

above, but now coupled with Lemma G.3 instead of Lemma 11.5 we have that:

$$Q_2 \leq 2 \exp(M \cdot (\varepsilon^2 - \varepsilon\delta/K)) + 2 \exp(-2M \cdot (\delta/K - \varepsilon)^2)$$

Adding the two upper bounds from  $Q_1$  and  $Q_2$  to Equation (G.1.2) we get the stated result. ■

We are now ready to prove Theorem G.1.

*Proof of Theorem G.1.* The proof follows directly the proof of Theorem 11.1 but we use the Lemmas that we stated above, for the estimators computed by Algorithm G.1. ■

# References

- Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *25th Advances in Neural Information Processing Systems (NIPS)*, pages 2312–2320, 2011. [122](#)
- Jacob Abernethy and Rafael M. Frongillo. A collaborative mechanism for crowdsourcing prediction problems. In *Advances in Neural Information Processing Systems*, 2011. [66](#)
- Jacob Abernethy, Yiling Chen, and Jennifer Wortman Vaughan. Efficient market making via convex optimization, and a connection to online learning. *ACM Transactions on Economics and Computation*, 1(2):12:1–12:38, 2013. [66](#)
- Jacob D. Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, 2008. [21](#), [122](#)
- Sachin Adlakha and Ramesh Johari. Mean field equilibrium in dynamic games with strategic complementarities. *Operations Research*, 61(4):971–989, 2013. [265](#)
- Alekh Agarwal, Dean P. Foster, Daniel J. Hsu, Sham M. Kakade, and Alexander Rakhlin. Stochastic convex optimization with bandit feedback. *SIAM Journal on Optimization*, 23(1):213–240, 2013. [122](#)
- Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1638–1646, Bejing, China, 22–24 Jun 2014. PMLR. [265](#)
- Arpit Agarwal, Shivani Agarwal, and Prathamesh Patil. Stochastic dueling bandits with adversarial corruption. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, 2021. [181](#)
- Rajeev Agrawal. The continuum-armed bandit problem. *SIAM J. Control and Optimization*, 33(6):1926–1951, 1995. [121](#)
- Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. The strategic perceptron. In *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*. ACM, 2021. [11](#), [29](#), [88](#), [238](#), [245](#)
- Noga Alon, Nicolo Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. From bandits to experts: A tale of domination and independence. *Advances in Neural Information Processing Systems*, 26, 2013. [261](#), [265](#)

Noga Alon, Nicolo Cesa-Bianchi, Ofer Dekel, and Tomer Koren. Online learning with feedback graphs: Beyond bandits. In *JMLR WORKSHOP AND CONFERENCE PROCEEDINGS*, volume 40. Microtome Publishing, 2015. [89](#), [102](#), [108](#), [265](#), [289](#)

Tal Alon, Magdalen Dobson, Ariel Procaccia, Inbal Talgam-Cohen, and Jamie Tucker-Foltz. Multi-agent evaluation mechanisms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1774–1781, 2020. [236](#)

Kareem Amin, Afshin Rostamizadeh, and Umar Syed. Learning prices for repeated auctions with strategic buyers. In *27th Annual Conference on Neural Information Processing Systems 2013.*, pages 1169–1177, 2013. [182](#)

Kareem Amin, Afshin Rostamizadeh, and Umar Syed. Repeated contextual auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, 2014. [182](#), [264](#)

Kareem Amin, Rachel Cummings, Lili Dworkin, Michael Kearns, and Aaron Roth. Online learning and profit maximization from revealed preferences. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. [264](#)

Idan Amir, Idan Attias, Tomer Koren, Roi Livni, and Yishay Mansour. Prediction with corrupted expert advice. *Proceedings of 32nd Advances in Neural Processing Systems (NeurIPS)*, 2020. [181](#)

William N Anderson Jr, E James Harner, and George E Trapp. Eigenvalues of the difference and product of projections. *Linear and Multilinear Algebra*, 17(3-4):295–299, 1985. [246](#)

Jose Apesteguia, Steffen Huck, and Jörg Oechssler. Imitation-theory and experimental evidence. *Journal of Economic Theory*, 136(1):217–235, 2007. [15](#)

David Applegate and Ravi Kannan. Sampling and integration of near log-concave functions. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 156–163, 1991. [231](#)

Raman Arora, Michael Dinitz, Teodor Vanislavov Marinov, and Mehryar Mohri. Policy regret in repeated games. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 6733–6742, 2018. [90](#)

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012. [68](#), [373](#)

Javed A Aslam and Aditi Dhagat. Searching in the presence of linearly bounded errors. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 486–493, 1991. [182](#)

Peter Auer and Chao-Kai Chiang. An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 116–120, 2016. [112](#)

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 322–331. IEEE, 1995. [98](#)

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002a. [117](#), [119](#)

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multi-armed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002b. 10, 22, 65, 66, 68, 73, 74, 114, 117, 119, 123, 261, 279, 300, 325, 326

Mohammad Gheshlaghi Azar, Alessandro Lazaric, and Emma Brunskill. Online stochastic optimization under correlated bandit feedback. In *31th Intl. Conf. on Machine Learning (ICML)*, pages 1557–1565, 2014. 121

Moshe Babaioff, Shaddin Dughmi, Robert D. Kleinberg, and Aleksandrs Slivkins. Dynamic pricing with limited supply. *ACM Trans. on Economics and Computation*, 3(1):4, 2015. Special issue for 13th ACM EC, 2012. 122

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *J. of the ACM*, 65(3):13:1–13:55, 2018. Preliminary version in FOCS 2013. 122

Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the sixteenth ACM conference on economics and computation*, pages 61–78, 2015. 89, 92

Santiago R Balseiro and Yonatan Gur. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science*, 65(9):3952–3968, 2019. 264

Santiago R Balseiro, Omar Besbes, and Gabriel Y Weintraub. Repeated auctions with budgets in ad exchanges: Approximations and design. *Management Science*, 61(4):864–884, 2015. 265

Gah-Yi Ban and N Bora Keskin. Personalized dynamic pricing with machine learning: High-dimensional features and heterogeneous elasticity. *Management Science*, 2021. 180

Albert Bandura. Observational learning. *The international encyclopedia of communication*, 2008. 15

Imre Bárány, Alfredo Hubard, and Jesús Jerónimo. Slicing convex sets and measures by a hyperplane. *Discrete & Computational Geometry*, 39(1):67–75, 2008. 48

Salvador Barberà, Dolors Berga, and Bernardo Moreno. Individual versus group strategy-proofness: When do they coincide? *Journal of Economic Theory*, 145(5):1648–1674, 2010. 52, 53

Hamsa Bastani and Mohsen Bayati. Online decision making with high-dimensional covariates. *Oper. Res.*, 68(1):276–294, 2020. 6, 13, 180

Soumya Basu, Rajat Sen, Sujay Sanghavi, and Sanjay Shakkottai. Blocking bandits. *Advances in Neural Information Processing Systems*, 32, 2019. 298

Soumya Basu, Orestis Papadigenopoulos, Constantine Caramanis, and Sanjay Shakkottai. Contextual blocking bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 271–279. PMLR, 2021. 298

Yahav Bechavod, Katrina Ligett, Aaron Roth, Bo Waggoner, and Steven Z Wu. Equal opportunity in online classification with partial feedback. *Advances in Neural Information Processing Systems*, 32, 2019. 112

Yahav Bechavod, Katrina Ligett, Steven Wu, and Juba Ziani. Gaming helps! learning from strategic interactions in natural dynamics. In *International Conference on Artificial Intelligence and Statistics*, pages 1234–1242. PMLR, 2021. 14, 89, 237

Yahav Bechavod, Chara Podimata, Zhiwei Steven Wu, and Juba Ziani. Information discrepancy in strategic learning. In *Proceedings of the 39th International Conference on Machine Learning, ICML 2022*. PMLR, 2022. 17

Omer Ben-Porat and Moshe Tennenholtz. Best response regression. In *Advances in Neural Information Processing Systems*, pages 1499–1508, 2017. 88

Omer Ben-Porat and Moshe Tennenholtz. Competing prediction algorithms. *arXiv preprint arXiv:1806.01703*, 2018. 88

Omar Besbes and Assaf Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57(6):1407–1420, 2009. 122, 181

Omar Besbes and Assaf Zeevi. On the minimax complexity of pricing in a changing environment. *Operations Research*, 59(1):66–79, 2011. 122

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations research*, 63(5):1227–1244, 2015. 181

Nicholas Bishop, Hau Chan, Debmalya Mandal, and Long Tran-Thanh. Adversarial blocking bandits. *Advances in Neural Information Processing Systems*, 33:8139–8149, 2020. 298

Daniel Björkegren, Joshua E. Blumenstock, and Samsun Knight. Manipulation-proof machine learning. *CoRR*, abs/2004.03865, 2020. 2

Duncan Black. *Theory of Committees and Elections*. Cambridge University Press, 1958. 30

David Blackwell. Controlled random walks. In *Proceedings of the international congress of mathematicians*, volume 3, pages 336–338, 1954. 21

David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956. 21

Avrim Blum, Vijay Kumar, Atri Rudra, and Felix Wu. Online learning in online auctions. *Theoretical Computer Science*, 324(2-3):137–146, 2004. 264

Avrim Blum, MohammadTaghi Hajiaghayi, Katrina Ligett, and Aaron Roth. Regret minimization and the price of total anarchy. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 373–382. ACM, 2008. 264

Avrim Blum, Nika Haghtalab, and Ariel D. Procaccia. Learning optimal commitment to overcome insecurity. In *28th Advances in Neural Information Processing Systems (NIPS)*, pages 1826–1834, 2014. 89

Avrim Blum, Yishay Mansour, and Jamie Morgenstern. Learning valuation distributions from partial observation. In *AAAI*, pages 798–804, 2015. 264

Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of data science*. Cambridge University Press, 2016. 190, 356

Arnoud V. Den Boer. Dynamic pricing and learning: Historical origins, current research, and new directions. *Surveys in Operations Research and Management Science*, 20(1), June 2015. 121

Ilija Bogunovic, Andreas Krause, and Jonathan Scarlett. Corruption-tolerant gaussian process bandit optimization. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020. 181

Mark Braverman and Sumegha Garg. The role of randomness and noise in strategic classification. In *1st Symposium on Foundations of Responsible Computing, FORC 2020, June 1-3, 2020, Harvard University, Cambridge, MA, USA (virtual conference)*, volume 156, pages 9:1–9:20, 2020. 29, 236

Felix Breuer. Uneven splitting of ham sandwiches. *Discrete & Computational Geometry*, 43(4):876–892, 2010. 48

Josef Broder and Paat Rusmevichientong. Dynamic pricing under a general parametric choice model. *Operations Research*, 60(4):965–980, 2012. 181

George W Brown, Alexander M Mood, et al. On median tests for linear hypotheses. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, volume 2, pages 159–166. University of California Press Berkeley, 1951. 44

Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 547–555. ACM, 2011. 88

Sébastien Bubeck. *Bandits Games and Clustering Foundations*. PhD thesis, Univ. Lille 1, 2010. 124

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012. 22, 77, 89, 124, 260

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. Online optimization in x-armed bandits. In *21st Advances in Neural Information Processing Systems (NIPS)*, pages 201–208, 2008. 12, 21, 89, 114, 115, 116, 117, 119, 121

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvari. Online Optimization in X-Armed Bandits. *J. of Machine Learning Research (JMLR)*, 12:1587–1627, 2011a. Preliminary version in NIPS 2008. 114, 345

Sébastien Bubeck, Gilles Stoltz, and Jia Yuan Yu. Lipschitz bandits without the lipschitz constant. In *22nd Intl. Conf. on Algorithmic Learning Theory (ALT)*, pages 144–158, 2011b. 114, 121

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012. 265

Sébastien Bubeck, Ofer Dekel, Tomer Koren, and Yuval Peres. Bandit convex optimization:  $\sqrt{T}$  regret in one dimension. In *28th Conf. on Learning Theory (COLT)*, pages 266–278, 2015. 21

Sébastien Bubeck, Yin Tat Lee, and Ronen Eldan. Kernel-based methods for bandit convex optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 72–85. ACM, 2017. 88, 122

Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: Stochastic and adversarial bandits. In *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23, pages 42.1–42.23, 2012. 112, 181

- Adam Bull. Adaptive-treed bandits. *Bernoulli J. of Statistics*, 21(4):2289–2307, 2015. [114](#), [121](#)
- Yang Cai, Constantinos Daskalakis, and Christos Papadimitriou. Optimum statistical estimation with strategic data sources. In *Conference on Learning Theory*, pages 280–296. PMLR, 2015. [29](#), [88](#)
- Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, Maria Kyropoulou, Brendan Lucier, Renato Paes Leme, and Éva Tardos. Bounding the inefficiency of outcomes in generalized second price auctions. *Journal of Economic Theory*, 156:343–388, 2015. [264](#)
- Ioannis Caragiannis, Ariel D. Procaccia, and Nisarg Shah. Truthful univariate estimators. In *33rd Intl. Conf. on Machine Learning (ICML)*, pages 127–135, 2016. [29](#), [64](#)
- Felipe Caro and Jérémie Gallien. Inventory management of a fast-fashion retail network. *Operations research*, 58(2):257–273, 2010. [9](#)
- Felipe Caro, Jérémie Gallien, Miguel Díaz, Javier García, José Manuel Corredoira, Marcos Montes, José Antonio Ramos, and Juan Correa. Zara uses operations research to reengineer its global distribution process. *Interfaces*, 40(1):71–84, 2010. [9](#)
- Leonardo Cella and Nicolò Cesa-Bianchi. Stochastic bandits with delay-dependent payoffs. In *International Conference on Artificial Intelligence and Statistics*, pages 1168–1177. PMLR, 2020. [298](#)
- Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006. ISBN 978-0-521-84108-5. [285](#)
- Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P Helmbold, Robert E Schapire, and Manfred K Warmuth. How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485, 1997. [10](#), [65](#)
- Nicolo Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Regret minimization for reserve prices in second-price auctions. *IEEE Transactions on Information Theory*, 61(1):549–564, 2015. [264](#)
- Nicolò Cesa-Bianchi, Pierre Gaillard, Claudio Gentile, and Sébastien Gerchinovitz. Algorithmic chaining and the role of partial feedback in online nonparametric learning. In *30th Conf. on Learning Theory (COLT)*, pages 465–481, 2017. [122](#)
- Nicolo Cesa-Bianchi, Tommaso Cesari, and Vianney Perchet. Dynamic pricing with finitely many unknown valuations. In *Algorithmic Learning Theory*, pages 247–273. PMLR, 2019. [181](#)
- Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *49th ACM Symp. on Theory of Computing (STOC)*, pages 47–60, 2017. [64](#)
- Shuchi Chawla, Jason D. Hartline, and Denis Nekipelov. Mechanism design for data science. In *ACM Conference on Economics and Computation, EC '14, Stanford , CA, USA, June 8-12, 2014*, pages 711–712, 2014. [264](#)
- Xi Chen and Yining Wang. Robust dynamic pricing with demand learning in the presence of outlier customers. *working paper*, 2020. [181](#)
- Xi Chen, Akshay Krishnamurthy, and Yining Wang. Robust dynamic assortment optimization in the presence of outlier customers. *arXiv:1910.04183*, 2019. [181](#)
- Xi Chen, Zachary Owen, Clark Pixton, and David Simchi-Levi. A statistical learning approach to personalization in revenue management. *Management Science*, 2021. [181](#)

Yatong Chen, Jialu Wang, and Yang Liu. Strategic recourse in linear classification. *arXiv preprint arXiv:2011.00355*, 2020a. 236

Yiling Chen, Chara Podimata, Ariel D Procaccia, and Nisarg Shah. Strategyproof linear regression in high dimensions. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 9–26. ACM, 2018. 17, 32, 88

Yiling Chen, Yang Liu, and Chara Podimata. Learning strategy-aware linear classifiers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b. 11, 14, 17, 29, 238, 245

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Hedging the drift: Learning to optimize under non-stationarity. *Management Science*, 2021. 181

Alon Cohen, Tamir Hazan, and Tomer Koren. Online learning with feedback graphs without the graphs. In *International Conference on Machine Learning*, pages 811–819, 2016. 89, 265, 266

Maxime Cohen, Ilan Lobel, and Renato Paes Leme. Feature-based dynamic pricing. *Management Science*, 2020. 13, 14, 122, 180, 183, 185, 212, 214, 217

Richard Cole and Tim Roughgarden. The sample complexity of revenue maximization. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 243–252. ACM, 2014. 264

Rachel Cummings, Stratis Ioannidis, and Katrina Ligett. Truthful linear regression. In *28th Conf. on Learning Theory (COLT)*, pages 448–483, 2015. 29, 88

Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Adebayo Olusola Adetunmbi, Opeyemi Emmanuel Ajibawa, et al. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6):e01802, 2019. 5

Yuval Dagan, Yuval Filmus, Daniel Kane, and Shay Moran. The entropy of lies: playing twenty questions with a liar. *arXiv preprint arXiv:1811.02177*, 2018. 182

Meir Dan-Cohen. Decision rules and conduct rules: On acoustic separation in criminal law. *Harvard Law Review*, pages 625–677, 1984. 322

Varsha Dani, Thomas P. Hayes, and Sham Kakade. The Price of Bandit Information for Online Optimization. In *20th Advances in Neural Information Processing Systems (NIPS)*, 2007. 21

Varsha Dani, Thomas P. Hayes, and Sham Kakade. Stochastic Linear Optimization under Bandit Feedback. In *21th Conf. on Learning Theory (COLT)*, pages 355–366, 2008. 122

Geoffroy De Clippel, Herve Moulin, and Nicolaus Tideman. Impartial division of a dollar. *Journal of Economic Theory*, 139(1):176–191, 2008. 53

Thomas S. Dee, Will Dobbie, Brian A. Jacob, and Jonah Rockoff. The causes and consequences of test score manipulation: Evidence from the new york regents examinations. *American Economic Journal: Applied Economics*, 11(3):382–423, July 2019. doi: 10.1257/app.20170520. 2

Ofer Dekel and Elad Hazan. Better rates for any adversarial deterministic mdp. In *International Conference on Machine Learning*, pages 675–683. PMLR, 2013. 299

Ofer Dekel, Felix Fischer, and Ariel D. Procaccia. Incentive compatible regression learning. *Journal of Computer and System Sciences*, 76(8):759–777, 2010. [9](#), [27](#), [28](#), [29](#), [30](#), [61](#), [64](#), [88](#)

Arnoud V den Boer. Dynamic pricing with multiple products and partially specified demand distribution. *Mathematics of operations research*, 39(3):863–888, 2014. [181](#)

Arnoud V den Boer and Bert Zwart. Simultaneously learning and optimizing using controlled variance pricing. *Management science*, 60(3):770–783, 2014. [181](#)

Peerapong Dhangwatnotai, Tim Roughgarden, and Qiqi Yan. Revenue maximization with a single sample. *Games and Economic Behavior*, 91:318–333, 2015. [264](#)

Nishanth Dikkala and Éva Tardos. Can credit increase revenue? In *Web and Internet Economics - 9th International Conference, WINE 2013, Cambridge, MA, USA, December 11-14, 2013, Proceedings*, pages 121–133, 2013. [264](#)

Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 55–70, 2018. [11](#), [14](#), [29](#), [88](#), [89](#), [92](#), [238](#), [245](#), [337](#), [342](#)

David Dranove, Daniel Kessler, Mark McClellan, and Mark Satterthwaite. Is more information better? the effects of report cards on health care providers. *Journal of Political Economy*, 111(3):555–588, 2003. ISSN 00223808, 1537534X. [2](#)

Alexey Drutsa. Horizon-independent optimal pricing in repeated auctions with truthful and strategic buyers. In *Proceedings of the 26th International Conference on World Wide Web*, pages 33–42, 2017. [182](#)

Devdatt P Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *BRICS Report Series*, 3(25), 1996. [150](#)

Ronen Eldan. Thin shell implies spectral gap up to polylog via a stochastic localization scheme. *Geometric and Functional Analysis*, 23(2):532–569, 2013. [218](#)

John H Elton and Theodore P Hill. A stronger conclusion to the classical ham sandwich theorem. *European Journal of Combinatorics*, 32(5):657–661, 2011. [47](#)

Hossein Esfandiari, Amin Karbasi, Abbas Mehrabian, and Vahab Mirrokni. Regret bounds for batched bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7340–7348, 2021. [313](#)

Eyal Even-Dar, Michael Kearns, Yishay Mansour, and Jennifer Wortman. Regret to the best vs. regret to the average. *Machine Learning Journal*, 72:21–37, 2008. [72](#)

Michal Feldman, Tomer Koren, Roi Livni, Yishay Mansour, and Aviv Zohar. Online pricing with strategic and patient buyers. In *Advances in Neural Information Processing Systems*, 2016. [182](#), [264](#)

Zhe Feng, Chara Podimata, and Vasilis Syrgkanis. Learning to bid without knowing your value. In *19th ACM Conf. on Economics and Computation (ACM-EC)*, pages 505–522, 2018. [17](#), [122](#)

F. Fischer and M. Klimm. Optimal impartial selection. *SIAM Journal on Computing*, 44(5):1263–1285, 2015. [53](#)

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics, 2005. 21, 88, 122

Rupert Freeman, David Pennock, Chara Podimata, and Jennifer Wortman Vaughan. No-regret and incentive-compatible online learning. In *International Conference on Machine Learning*, pages 3270–3279. PMLR, 2020. 17

Yoav Freund and Robert E Schapire. Game theory, on-line prediction and boosting. In *9th Conf. on Learning Theory (COLT)*, pages 325–332, 1996. 21

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997. 10, 65, 66, 68

Rafael Frongillo, Nicoás Della Penna, and Mark D. Reid. Interpreting prediction markets: A stochastic approach. In *Advances in Neural Information Processing Systems*, 2012. 66

Ganesh Ghalme, Vineet Nair, Itay Eilat, Inbal Talgam-Cohen, and Nir Rosenfeld. Strategic classification in the dark. In *International Conference on Machine Learning*, pages 3672–3681. PMLR, 2021. 235, 236

John C Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):148–164, 1979. 298

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. 67, 69

Alexander Goldenshluger and Assaf Zeevi. A linear response bandit problem. *Stochastic Systems*, 2013. 180

Negin Golrezaei, Patrick Jaillet, and Jason Cheuk Nam Liang. Incentive-aware contextual pricing with non-parametric market noise. *arXiv preprint arXiv:1911.03508*, 2019a. 182

Negin Golrezaei, Adel Javanmard, and Vahab Mirrokni. Dynamic incentive-aware learning: Robust pricing in contextual auctions. In *Advances in Neural Information Processing Systems*, pages 9759–9769, 2019b. 182

Negin Golrezaei, Vahideh H. Manshadi, Jon Schneider, and Shreyas Sekar. Learning product rankings robust to fake users. In *Twenty-Second ACM Conference on Economics and Computation (EC)*, 2021. 181

Andres Gonzalez-Lira and Ahmed Mushfiq Mobarak. Slippery Fish: Enforcing Regulation under Subversive Adaptation. IZA Discussion Papers 12179, Institute of Labor Economics (IZA), February 2019. 2

Google. AdWords Bid Simulator. [https://support.google.com/adwords/answer/2470105?hl=en&ref\\_topic=3122864](https://support.google.com/adwords/answer/2470105?hl=en&ref_topic=3122864), 2018a. [Online; accessed 15-February-2018]. 262, 276

Google. Bid Landscapes. <https://developers.google.com/adwords/api/docs/guides/bid-landscapes>, 2018b. [Online; accessed 15-February-2018]. 262, 276

Google. Bid Lanscapes. <https://developers.google.com/adwords/api/docs/reference/v201710/DataService.BidLandscape>, 2018c. [Online; accessed 15-February-2018]. 262, 276

Michael Greenstone, Guojun He, Ruixue Jia, and Tong Liu. Can technology solve the principal-agent problem? evidence from chinas war on air pollution. *SSRN Electronic Journal*, 01 2020. doi: 10.2139/ssrn.3638591. 2

Jean-Bastien Grill, Michal Valko, and Rémi Munos. Black-box optimization of noisy functions with unknown smoothness. In *28th Advances in Neural Information Processing Systems (NIPS)*, 2015. 114, 121

Anupam Gupta, Tomer Koren, and Kunal Talwar. Better algorithms for stochastic bandits with adversarial corruptions. In *Conference on Learning Theory*, 2019a. 177, 181

Samarth Gupta, Shreyas Chaudhari, Gauri Joshi, and Osman Yağan. Multi-armed bandits with correlated arms. *IEEE Transactions on Information Theory*, 67(10):6711–6732, 2021. 299

Vivek Gupta, Pegah Nokhiz, Chitradeep Dutta Roy, and Suresh Venkatasubramanian. Equalizing recourse across groups. *CoRR*, abs/1909.03166, 2019b. 236

Yonatan Gur, Assaf J. Zeevi, and Omar Besbes. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Annual Conference on Neural Information Processing Systems 2014*, pages 199–207, 2014. 181

András Gyorgy, Tamás Linder, and Gábor Lugosi. Efficient tracking of large classes of experts. *IEEE Transactions on Information Theory*, 58(11):6709–6725, 2012. 283, 284, 285

Nika Haghtalab, Nicole Immorlica, Brendan Lucier, and Jack Z. Wang. Maximizing welfare with incentive-aware evaluation mechanisms. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 160–166, 2020. 236

Yanjun Han, Zhengyuan Zhou, and Tsachy Weissman. Optimal no-regret learning in repeated first-price auctions. *CoRR*, abs/2003.09795, 2020. 122

James Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957. 21

Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pages 111–122, 2016. 11, 14, 29, 88, 245

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007. 21

Hoda Heidari, Michael J Kearns, and Aaron Roth. Tight policy regret bounds for improving and decaying bandits. In *IJCAI*, pages 1562–1570, 2016. 298

Hoda Heidari, Claudio Ferrari, Krishna P. Gummadi, and Andreas Krause. Fairness behind a veil of ignorance: A welfare analysis for automated decision making. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018. 237

Hoda Heidari, Vedant Nanda, and Krishna P. Gummadi. On the long-term impact of algorithmic decision policies: Effort unfairness and feature segregation through social learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2692–2701. PMLR, 2019. [237](#)

Chien-Ju Ho, Aleksandrs Slivkins, and Jennifer Wortman Vaughan. Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems. *J. of Artificial Intelligence Research*, 55:317–359, 2016. Preliminary version appeared in ACM EC 2014. [114](#), [121](#)

Henning Hohnhold, Deirdre O’Brien, and Diane Tang. Focusing on the long-term: It’s good for users and business. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1849–1858, 2015. [16](#), [17](#), [298](#)

Ron Holzman and Hervé Moulin. Impartial nominations for a prize. *Econometrica*, 81(1):173–196, 2013. [53](#)

Safwan Hossain and Nisarg Shah. The effect of strategic noise in linear regression. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS ’20, Auckland, New Zealand, May 9-13, 2020*, pages 511–519. International Foundation for Autonomous Agents and Multiagent Systems, 2020. [28](#), [29](#)

Jinli Hu and Amos Storkey. Multi-period trading prediction markets with connections to machine learning. In *International Conference on Machine Learning*, pages 1773–1781, 2014. [66](#)

Lily Hu and Yiling Chen. Fair classification and social welfare. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 535–545, 2020. [237](#)

Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 259–268, 2019. [14](#), [29](#), [236](#)

D. Bradford Hunt. Redlining. *Encyclopedia of Chicago*, 2005. [15](#)

Stratis Ioannidis and Patrick Loiseau. Linear regression as a non-cooperative game. In *International Conference on Web and Internet Economics*, pages 277–290. Springer, 2013. [29](#)

Krishnamurthy Iyer, Ramesh Johari, and Mukund Sundararajan. Mean field equilibria of dynamic auctions with learning. *ACM SIGecom Exchanges*, 10(3):10–14, 2011. [265](#)

Meena Jagadeesan, Celestine Mendler-Dünner, and Moritz Hardt. Alternative microfoundations for strategic classification. In *International Conference on Machine Learning*, pages 4687–4697. PMLR, 2021. [89](#)

Adel Javanmard and Hamid Nazerzadeh. Dynamic pricing in high-dimensions. *The Journal of Machine Learning Research*, 20(1):315–363, 2019. [180](#)

Kumar Joag-Dev and Frank Proschan. Negative association of random variables with applications. *The Annals of Statistics*, pages 286–295, 1983. [150](#)

Iain M Johnstone and Paul F Velleman. The resistant line and related regression methods. *Journal of the American Statistical Association*, 80(392):1041–1054, 1985. [27](#), [40](#), [43](#), [44](#)

Yash Kanoria and Hamid Nazerzadeh. Dynamic reserve prices for repeated auctions: Learning from bids - working paper. In *Web and Internet Economics - 10th International Conference, WINE 2014, Beijing, China, December 14-17, 2014. Proceedings*, page 232, 2014. [264](#)

Yash Kanoria and Hamid Nazerzadeh. Incentive-compatible learning of reserve prices for repeated auctions. *Operations Research*, 69(2):509–524, 2021. [182](#)

Richard M Karp and Robert Kleinberg. Noisy binary search and its applications. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 881–890, 2007. [182](#)

Mark G Kelly, David J Hand, and Niall M Adams. The impact of changing populations on classifier performance. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 367–371, 1999. [3](#)

Horst Kermer and AB Németh. Supporting spheres for families of independent convex sets. *Archiv der Mathematik*, 24(1):91–96, 1973. [48](#)

N Bora Keskin and Assaf Zeevi. Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research*, 62(5):1142–1167, 2014. [181](#)

N. Bora Keskin and Assaf Zeevi. Chasing demand: Learning and earning in a changing environment. *Math. Oper. Res.*, 42(2):277–307, 2017a. [122](#)

N. Bora Keskin and Assaf Zeevi. Chasing demand: Learning and earning in a changing environment. *Math. Oper. Res.*, 42(2):277–307, 2017b. [181](#)

Moein Khajehnejad, Behzad Tabibian, Bernhard Schölkopf, Adish Singla, and Manuel Gomez-Rodriguez. Optimal decision making under strategic behavior. *arXiv preprint arXiv:1905.09239*, 2019. [236](#)

Khashayar Khosravi, Renato Paes Leme, and Chara Podimata. Bandits with long-term effects. *Working Paper*, 2022. [17](#)

Jyrki Kivinen and Manfred K Warmuth. Averaging expert predictions. In *European Conference on Computational Learning Theory*, pages 153–167. Springer, 1999. [68](#)

Jon Kleinberg and Manish Raghavan. How do classifiers induce agents to invest effort strategically? *ACM Transactions on Economics and Computation (TEAC)*, 8(4):1–23, 2020. [14](#), [236](#)

Jon Kleinberg, Aleksandrs Slivkins, and Tom Wexler. Triangulation and embedding using small sets of beacons. *J. of the ACM*, 56(6), September 2009. Subsumes conference papers in *IEEE FOCS 2004* and *ACM-SIAM SODA 2005*. [120](#)

Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *18th Advances in Neural Information Processing Systems (NIPS)*, 2004. [114](#), [116](#), [121](#)

Robert Kleinberg and Nicole Immorlica. Recharging bandits. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 309–319. IEEE, 2018. [298](#)

Robert Kleinberg and Tom Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *Symposium on Foundations of Computer Science*. IEEE, 2003. [114](#), [119](#), [122](#), [173](#), [179](#), [181](#)

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 681–690. ACM, 2008a. [276](#)

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *40th ACM Symp. on Theory of Computing (STOC)*, pages 681–690, 2008b. [12](#), [89](#), [114](#), [115](#), [116](#), [117](#), [119](#), [121](#)

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Bandits and experts in metric spaces. *J. of the ACM*, 66(4):30:1–30:77, May 2019. Merged and revised version of conference papers in ACM STOC 2008 and ACM-SIAM SODA 2010. Also available at <http://arxiv.org/abs/1312.1277>. [21](#), [114](#), [121](#), [173](#), [345](#)

Robert D Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems*, pages 697–704, 2005. [276](#)

Levente Kocsis and Csaba Szepesvari. Bandit Based Monte-Carlo Planning. In *17th European Conf. on Machine Learning (ECML)*, pages 282–293, 2006. [121](#)

Tomer Koren, Roi Livni, and Yishay Mansour. Bandits with movement costs and adaptive pricing. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1242–1268, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR. [264](#)

Akshay Krishnamurthy, John Langford, Aleksandrs Slivkins, and Chicheng Zhang. Contextual bandits with continuous actions: Smoothing, zooming, and adapting. *J. of Machine Learning Research (JMLR)*, 27(137):1–45, 2020. Preliminary version at *COLT 2019*. [114](#), [121](#)

Akshay Krishnamurthy, Thodoris Lykouris, Chara Podimata, and Robert Schapire. Contextual search in the presence of irrational agents. In *53rd Annual Symposium on Theory of Computing, STOC 2021*, 2021. [17](#), [122](#), [217](#)

David Kurokawa, Omer Lev, Jamie Morgenstern, and Ariel D Procaccia. Impartial peer review. In *24th Intl. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 582–588, 2015. [53](#)

Nicolas S Lambert, John Langford, Jennifer Wortman, Yiling Chen, Daniel Reeves, Yoav Shoham, et al. Self-financed wagering mechanisms for forecasting. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, pages 170–179, 2008. [10](#), [66](#), [70](#)

Nicolas S Lambert, John Langford, Jennifer Wortman Vaughan, Yiling Chen, Daniel M Reeves, Yoav Shoham, and David M Pennock. An axiomatic characterization of wagering mechanisms. *Journal of Economic Theory*, 156:389–416, 2015. [10](#), [66](#), [71](#)

Search Engine Land. Bing ads launches bid landscape, a keyword level bid simulator tool. <https://searchengineland.com/bing-ads-launches-bid-landscape-keyword-level-bid-simulator-tool-187219>, 2014. [Online; accessed 15-February-2018]. [xi](#), [263](#)

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020. [22](#), [89](#)

Yin Tat Lee and Santosh Vempala. *Techniques in Optimization and Sampling*. (book in progress), 2021. [218](#), [219](#), [231](#)

Renato Paes Leme and Jon Schneider. Contextual search via intrinsic volumes. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 268–282, 2018. [217](#)

Liu Leqi, Fatma Kilinc Karzan, Zachary Lipton, and Alan Montgomery. Rebounding bandits for modeling satiation effects. *Advances in Neural Information Processing Systems*, 34, 2021. [298](#)

Joshua Letchford, Vincent Conitzer, and Kamesh Munagala. Learning and approximating the optimal strategy to commit to. In *2nd Symp. on Algorithmic Game Theory (SAGT)*, pages 250–262, 2009. [89](#)

Nir Levine, Koby Crammer, and Shie Mannor. Rotting bandits. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3074–3083, 2017. [298](#)

Yingkai Li, Edmund Y Lou, and Liren Shan. Stochastic linear optimization with adversarial corruption. *arXiv:1909.02109*, 2019. [181](#)

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994. [10](#), [21](#), [65](#)

Allen Liu, Renato Paes Leme, and Jon Schneider. Optimal contextual pricing and extensions. In *Symposium on Discrete Algorithms*, 2021. [14](#), [122](#), [180](#), [214](#), [216](#), [217](#)

Jinyan Liu, Zhiyi Huang, and Xiangning Wang. Learning optimal reserve price against non-myopic bidders. In *Annual Conference on Neural Information Processing Systems 2018*, 2018. [182](#)

Lydia T Liu, Ashia Wilson, Nika Haghtalab, Adam Tauman Kalai, Christian Borgs, and Jennifer Chayes. The disparate equilibria of algorithmic decision making when individuals invest rationally. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 381–391, 2020. [236](#)

Wei Liu and Sanjay Chawla. A game theoretical model for adversarial learning. In *2009 IEEE International Conference on Data Mining Workshops*, pages 25–30. IEEE, 2009. [88](#)

Ilan Lobel, Renato Paes Leme, and Adrian Vladu. Multidimensional binary search for contextual decision-making. *Operations Research*, 2018. [13](#), [122](#), [178](#), [180](#), [183](#), [187](#), [194](#), [195](#), [202](#), [203](#), [214](#), [216](#), [217](#), [348](#), [349](#), [354](#), [358](#), [360](#)

Andrea Locatelli and Alexandra Carpentier. Adaptivity to smoothness in x-armed bandits. In *31th Conf. on Learning Theory (COLT)*, pages 1463–1492, 2018. [121](#)

László Lovász and Santosh Vempala. The geometry of logconcave functions and sampling algorithms. *Random Structures & Algorithms*, 30(3):307–358, 2007. [219](#)

Thodoris Lykouris, Vasilis Syrgkanis, and Éva Tardos. Learning and efficiency in games with dynamic population. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 120–129. Society for Industrial and Applied Mathematics, 2016. [286](#)

Thodoris Lykouris, Vahab S. Mirrokni, and Renato Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Symposium on Theory of Computing*, 2018. [13](#), [177](#), [179](#), [181](#), [184](#), [192](#), [205](#), [350](#)

Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme. Bandits with adversarial scaling. In *International Conference on Machine Learning*, pages 6511–6521. PMLR, 2020. [298](#)

Thodoris Lykouris, Max Simchowitz, Aleksandrs Slivkins, and Wen Sun. Corruption-robust exploration in episodic reinforcement learning. In *Annual Conference on Learning Theory*, 2021. [181](#)

Odalric-Ambrym Maillard and Rémi Munos. Online Learning in Adversarial Lipschitz Environments. In *European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 305–320, 2010. [121](#)

Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *NIPS*, pages 684–692, 2011. [261](#), [265](#)

Jieming Mao, Renato Paes Leme, and Jon Schneider. Contextual pricing for lipschitz buyers. In *Advances in Neural Information Processing Systems*, 2018. [182](#)

Janusz Marecki, Gerry Tesauro, and Richard Segal. Playing repeated Stackelberg games with unknown opponents. In *11th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 821–828, 2012. [89](#)

John McCarthy. Measures of the value of information. *Proceedings of the National Academy of Sciences of the United States of America*, 42(9):654, 1956. [69](#)

Reshef Meir, Ariel D Procaccia, and Jeffrey S Rosenschein. On the limits of dictatorial classification. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 609–616, 2010. [29](#), [88](#)

Reshef Meir, Shaull Almagor, Assaf Michaely, and Jeffrey S. Rosenschein. Tight bounds for strategyproof classification. In *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, 2011, Volume 1-3*, pages 319–326, 2011. [29](#), [88](#)

Reshef Meir, Ariel D Procaccia, and Jeffrey S Rosenschein. Algorithms for strategyproof classification. *Artificial Intelligence*, 186:123–156, 2012. [28](#), [29](#), [88](#)

Microsoft. BingAds, Bid Landscapes. <https://advertise.bingads.microsoft.com/en-us/resources/training/bidding-and-traffic-estimation>, 2018. [Online; accessed 15-February-2018]. [262](#), [276](#)

John Miller, Smitha Milli, and Moritz Hardt. Strategic classification is causal modeling in disguise. In *International Conference on Machine Learning*, pages 6917–6926. PMLR, 2020. [237](#)

Smitha Milli, John Miller, Anca D Dragan, and Moritz Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 230–239, 2019. [29](#), [236](#)

Stanislav Minsker. Estimation of extreme values and associated level sets of a regression function via selective sampling. In *26th Conf. on Learning Theory (COLT)*, pages 105–121, 2013. [114](#)

Yonatan Mintz, Anil Aswani, Philip Kaminsky, Elena Flowers, and Yoshimi Fukuoka. Nonstationary bandits with habituation and recovery dynamics. *Operations Research*, 68(5):1493–1516, 2020. [298](#)

Mehryar Mohri and Andrés Munoz Medina. Revenue optimization against strategic buyers. In *NIPS*, 2015. 182

Mehryar Mohri and Andres Munoz. Optimal regret minimization in posted-price auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, pages 1871–1879, 2014. 264

Mehryar Mohri and Andres Munoz Medina. Optimal regret minimization in posted-price auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, 2014. 182

Hervé Moulin. On strategy-proofness and single-peakedness. *Public Choice*, 35:437–455, 1980. 28, 30, 52, 56, 57, 58, 64

Rémi Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In *25th Advances in Neural Information Processing Systems (NIPS)*, pages 783–791, 2011. 114, 121

Rémi Munos and Pierre-Arnaud Coquelin. Bandit algorithms for tree search. In *23rd Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2007. 121

Andres Munoz and Sergei Vassilvitskii. Revenue optimization with approximate bid predictions. *Advances in Neural Information Processing Systems*, 30, 2017. 264

Mila Nambiar, David Simchi-Levi, and He Wang. Dynamic learning and pricing with model misspecification. *Management Science*, 65(11):4980–5000, 2019. 180

Subhash C Narula and John F Wellington. Interior analysis for the minimum sum of absolute errors regression. *Technometrics*, 27(2):181–188, 1985. 59, 60

Albert B Novikoff. On convergence proofs for perceptrons. Technical report, STANFORD RESEARCH INST MENLO PARK CA, 1963. 360

Robert Nowak. Generalized binary search. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pages 568–574. IEEE, 2008. 182

Robert Nowak. Noisy generalized binary search. In *Advances in neural information processing systems*, pages 1366–1374, 2009. 182

Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. In *Advances in Neural Information Processing Systems*, pages 577–585, 2016. 67

Ronald Ortner. Online regret bounds for markov decision processes with deterministic transitions. In *International Conference on Algorithmic Learning Theory*, pages 123–137. Springer, 2008. 299

Ronald Ortner and Daniil Ryabko. Online regret bounds for undiscounted continuous reinforcement learning. *Advances in Neural Information Processing Systems*, 25, 2012. 299

Michael Ostrovsky and Michael Schwarz. Reserve prices in internet advertising auctions: A field experiment. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 59–60. ACM, 2011. 264

Renato Paes Leme and Jon Schneider. Contextual search via intrinsic volumes. In *Symposium on Foundations of Computer Science*. IEEE, 2018. 122, 180, 214

Renato Paes Leme, Chara Podimata, and Jon Schneider. Corruption-robust contextual search through density updates. In *Conference on Learning Theory*. PMLR, 2022. 17

Sandeep Pandey, Deepak Agarwal, Deepayan Chakrabarti, and Vanja Josifovski. Bandits for Taxonomies: A Model-based Approach. In *SIAM Intl. Conf. on Data Mining (SDM)*, 2007. 121

Andrzej Pelc. Coding with bounded error fraction. *Ars Combinatoria*, 24:17–22, 1987. 182

Andrzej Pelc. Searching games with errorsfifty years of coping with liars. *Theoretical Computer Science*, 270(1-2):71–109, 2002. 182

Juan C. Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 7599–7609. PMLR, 2020. 89, 237

Juan Perote and J. Perote-Peña. The impossibility of strategy-proof clustering. *Economics Bulletin*, 4(23):1–9, 2003. 28, 29

Juan Perote and J. Perote-Peña. Strategy-proof estimators for simple regression. *Mathematical Social Sciences*, 47:153–176, 2004. 9, 27, 28, 29, 30, 40, 41, 43, 44, 88

Ciara Pike-Burke and Steffen Grunewalder. Recovering bandits. *Advances in Neural Information Processing Systems*, 32, 2019. 298

Chara Podimata and Alex Slivkins. Adaptive discretization for adversarial lipschitz bandits. In *Conference on Learning Theory*, pages 3788–3805. PMLR, 2021. 17, 89, 181

pwc report. Internet advertising revenue report. [https://www.iab.com/wp-content/uploads/2020/05/FY19-IAB-Internet-Ad-Revenue-Report\\_Final.pdf](https://www.iab.com/wp-content/uploads/2020/05/FY19-IAB-Internet-Ad-Revenue-Report_Final.pdf), 2020. Accessed: 06/09/2022. 6, 15

Sheng Qiang and Mohsen Bayati. Dynamic pricing with demand covariates. Available at SSRN 2765257, 2016. 180

Luis A. Rademacher. Approximating the centroid is hard. In *Proceedings of the Twenty-Third Annual Symposium on Computational Geometry*, SCG '07, page 302305, New York, NY, USA, 2007. 187

Mark D Reid and Robert C Williamson. Surrogate regret bounds for proper losses. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 897–904. ACM, 2009. 67, 69

Jason Rhuggenaath, Paulo Roberto de Oliveira da Costa, Yingqian Zhang, Alp Akcay, and Uzay Kaymak. Low-regret algorithms for strategic buyers with unknown valuations in repeated posted-price auctions. In *Machine Learning and Knowledge Discovery in Databases - European Conference*, 2020, 2020. 182

Ronald L. Rivest, Albert R. Meyer, Daniel J. Kleitman, Karl Winklmann, and Joel Spencer. Coping with errors in binary search procedures. *Journal of Computer and System Sciences*, 20(3):396–404, 1980. 182, 206

Giulia Romano, Gianluca Tartaglia, Alberto Marchesi, and Nicola Gatti. Online posted pricing with unknown time-discounted valuations. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021. 182

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 1958. 190

Aaron Roth, Jonathan Ullman, and Zhiwei Steven Wu. Watch and learn: Optimizing from revealed preferences feedback. In *Symposium on Theory of Computing*. ACM, 2016. 181

Aaron Roth, Aleksandrs Slivkins, Jonathan Ullman, and Zhiwei Steven Wu. Multidimensional dynamic pricing for welfare maximization. *ACM Transactions on Economics and Computation (TEAC)*, 2020. 181

Tim Roughgarden. Intrinsic robustness of the price of anarchy. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 513–522. ACM, 2009. 264

Tim Roughgarden and Okke Schrijvers. Online prediction with selfish experts. In *Advances in Neural Information Processing Systems*, pages 1300–1310, 2017. 10, 66, 70

Leonard J Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971. 67, 69

Jeffrey C Schlimmer and Richard H Granger. Beyond incremental processing: tracking concept drift. In *AAAI*, pages 502–507, 1986. 3

Yevgeny Seldin and Gábor Lugosi. An improved parametrization and analysis of the exp3++ algorithm for stochastic and adversarial bandits. *arXiv preprint arXiv:1702.06103*, 2017. 112

Yevgeny Seldin and Aleksandrs Slivkins. One practical algorithm for both stochastic and adversarial bandits. In *ICML*, pages 1287–1295, 2014. 112

Julien Seznec, Andrea Locatelli, Alexandra Carpentier, Alessandro Lazaric, and Michal Valko. Rotting bandits are no harder than stochastic ones. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, 2019. 298

Virag Shah, Ramesh Johari, and Ramesh Johari. Semi-parametric dynamic contextual pricing. In *Advances in Neural Information Processing Systems*, pages 2360–2370, 2019. 181, 182

Yonadav Shavit, Benjamin L. Edelman, and Brian Axelrod. Causal strategic linear regression. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8676–8686. PMLR, 2020. 14, 89, 237

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000. 3

Aleksandrs Slivkins. Multi-armed bandits on implicit metric spaces. In *25th Advances in Neural Information Processing Systems (NIPS)*, 2011. 114, 121

Aleksandrs Slivkins. Contextual bandits with similarity information. *J. of Machine Learning Research (JMLR)*, 15(1):2533–2568, 2014. Preliminary version in *COLT 2011*. 114, 116, 121

Aleksandrs Slivkins. Introduction to multi-armed bandits. *Found. Trends Mach. Learn.*, 12(1-2):1–286, 2019. 22, 89, 130

Aleksandrs Slivkins and Eli Upfal. Adapting to a changing environment: the brownian restless bandits. In *21st Annual Conference on Learning Theory*, 2008. 181

Aleksandrs Slivkins, Filip Radlinski, and Sreenivas Gollapudi. Ranked bandits in metric spaces: Learning optimally diverse rankings over large document collections. *J. of Machine Learning Research (JMLR)*, 14(Feb):399–436, 2013. Preliminary version in 27th ICML, 2010. 114, 121

Joel Spencer. Ulam’s searching game with a fixed number of lies. *Theoretical Computer Science*, 95(2):307–321, 1992. 182

Joel Spencer and Peter Winkler. Three thresholds for a liar. *Combinatorics, Probability & Computing*, 1:81–93, 1992. 182

Search Marketing Standard. Google adwords improves the bid simulator tool feature. <http://www.searchmarketingstandard.com/google-adwords-improves-the-bid-simulator-tool-feature>, 2014. [Online; accessed 15-February-2018]. xi, 263

Richard P Stanley et al. An introduction to hyperplane arrangements. *Geometric combinatorics*, 13:389–496, 2004. 225, 340

William Steiger and Jihui Zhao. Generalized ham-sandwich cuts. *Discrete & Computational Geometry*, 44(3):535–545, 2010. 47, 48, 49

Arthur H Stone and John W Tukey. Generalized sandwich theorems. *Duke Mathematical Journal*, 9(2):356–359, 1942. 47

Shohei Tamura and Shinji Ohseto. Impartial nomination correspondences. *Social Choice and Welfare*, 43(1):47–54, 2014. 53

Philip E. Tetlock and Dan Gardner. *Superforecasting: The Art and Science of Prediction*. Crown, 2015. 10

Lyn Thomas, Jonathan Crook, and David Edelman. *Credit scoring and its applications*. SIAM, 2017. 6

Stratis Tsirtsis and Manuel Gomez Rodriguez. Decisions, counterfactual explanations and strategic behavior. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 236

John W Tukey et al. *Exploratory data analysis*, volume 2. Reading, MA, 1977. 44

Stanisław M. Ulam. *Adventures of a mathematician*. Charles Scribner’s Sons, New York, NY, USA, 1976. 182

Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 10–19, 2019. 89, 236

Michal Valko, Alexandra Carpentier, and Rémi Munos. Stochastic simultaneous optimistic optimization. In *30th Intl. Conf. on Machine Learning (ICML)*, pages 19–27, 2013. 114, 121

Vladimir Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 56(2):153–173, 1998. 10, 21, 65

Volodimir G Vovk. Aggregating strategies. *Proc. of Computational Learning Theory*, 1990, 1990. 10, 65

Zizhuo Wang, Shiming Deng, and Yinyu Ye. Close the gaps: A learning-while-doing algorithm for single-product revenue management problems. *Operations Research*, 62(2):318–331, 2014. 122

Romain Warlop, Alessandro Lazaric, and Jérémie Mary. Fighting boredom in recommender systems with linear reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018. 298

Jonathan Weed, Vianney Perchet, and Philippe Rigollet. Online learning in repeated auctions. In *Conference on Learning Theory*, 2016. 122, 260, 263, 264, 272, 277, 278, 279

Chen-Yu Wei and Haipeng Luo. Non-stationary reinforcement learning without prior knowledge: an optimal black-box approach. In *Conference on Learning Theory, COLT 2021*, 2021. 181

Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, 25(A):287–298, 1988. 298

Gerhard Widmer and Miroslav Kubat. Effective learning in dynamic environments by explicit context tracking. In *European Conference on Machine Learning*, pages 227–243. Springer, 1993. 3

Jens Witkowski, Rupert Freeman, Jennifer Wortman Vaughan, David M Pennock, and Andreas Krause. Incentive-compatible forecasting competitions. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 66, 67, 80, 328

Wordstream. Bid management tools. <https://www.wordstream.com/bid-management-tools>, 2018. [Online; accessed 15-February-2018]. 262

Thomas Zaslavsky. Counting the faces of cut-up spaces. *Bulletin of the American Mathematical Society*, 81(5):916–918, 1975. 340

Anton Zhiyanov and A. Drutsa. Bisection-based pricing for repeated contextual auctions against strategic buyer. In *Proceedings of the Thirty-Seventh International Conference in Machine Learning*, 2020. 182

Julian Zimmert and Yevgeny Seldin. Tsallis-inf: An optimal algorithm for stochastic and adversarial bandits. *Journal of Machine Learning Research (JMLR)*, 2021. 177, 181

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *20th Intl. Conf. on Machine Learning (ICML)*, pages 928–936, 2003. 21