Grinding the Space: Learning to Classify Against Strategic Agents

Chara Podimata, Harvard University

Joint work with
Yiling Chen (Harvard University), and Yang Liu (UC Santa Cruz)

Machine Learning (ML) for Decision Making that Matters

Three Ways AI Will Impact The Lending Industry

Jeff Levin Forbes Councils Member
Forbes Real Estate Council COUNCIL POST | Paid Program
Real Estate

POST WRITTEN BY

Jeff Levin

Jeff Levin is a real estate investor & founder of **Specialty Lending Group**, a private lending firm based in Washington, DC.

Machine Learning (ML) for Decision Making that Matters

Three Ways AI Will Impact The Lending Industry

Jeff Levin Forbes Councils Member
Forbes Real Estate Council COUNCIL POST | Paid Program
Real Estate

POST WRITTEN BY

Jeff Levin

Jeff Levin is a real estate investor & founder of **Specialty Lending Group**, a private lending firm based in Washington, DC.

POLICY | SCIENCE | US & WORLD

How artificial intelligence can help us make judges less biased

Predicting which judges are likely to be biased could give them the opportunity to consider more carefully

By Angela Chen | @chengela | Jan 17, 2019, 12:07pm EST

Machine Learning (ML) for Decision Making that Matters

Three Ways AI Will Impact The Lending Industry

Jeff Levin Forbes Councils Member
Forbes Real Estate Council COUNCIL POST | Paid Program
Real Estate

POST WRITTEN BY

Jeff Levin

Jeff Levin is a real estate investor & founder of **Specialty Lending Group**, a private lending firm based in Washington, DC.

POLICY | SCIENCE | US & WORLD

How artificial intelligence can help us make judges less biased

Predicting which judges are likely to be biased could give them the opportunity to consider more carefully

By Angela Chen | @chengela | Jan 17, 2019, 12:07pm EST

Health & Wellness

Why two insurance companies want to pay Hongkongers to work out

Insurers seek ways to make sure customers are fitter, and therefore healthier and less likely to make claims, in wake of data showing around half of Hong Kong adults don't exercise enough

Machine Learning (ML) for Decision Making that Matters

Three Ways AI Will Impact The Lending Industry

Jeff Levin Forbes Councils Member
Forbes Real Estate Council COUNCIL POST | Paid Program
Real Estate

POST WRITTEN BY

Jeff Levin

Jeff Levin is a real estate investor & founder of **Specialty Lending C**
lending firm based in Washington, DC.

POLICY | SCIENCE | US & WORLD

How artificial intelligenc n help us make judges les

Predicting whic the opp

ARTIFICIAL INTELLIGENCE

Applying for Your Next Job May Be an Automated Nightmare

n & Wellness

Why two insurance companies want to pay Hongkongers to work out

Insurers seek ways to make sure customers are fitter, and therefore healthier and less likely to make claims, in wake of data showing around half of Hong Kong adults don't exercise enough

Health & Wellness

Why two insurance companies want to pay Hongkongers to work out

Insurers seek ways to make sure customers are fitter, and therefore healthier and less likely to make claims, in wake of data showing around half of Hong Kong adults don’t exercise enough

ML Against Strategic Agents

Strength** of agent according to what they **report

Stochastic	Strategic	Adversarial
(report ~ Distribution)	(report ~ utility maximizer)	(**any** report)

Learner's Goal:

Either induce **truthfulness** (→ learn as if 'clean' dataset) or

Learn by exploiting the **structure of the agents' behavior**

This work: online learning in repeated classification settings against strategic agents.

Model (1)

$$y_t = h^*(\boldsymbol{x}_t), \hat{y}_t = h^*(\boldsymbol{z}_t(\alpha_t; \boldsymbol{x}_t))$$

$h^*(\cdot)$: **ideal** classifier

Repeated Interaction Protocol

For round $t \in [T]$:

1. Environment chooses $\boldsymbol{x}_t \sim \mathcal{X} \subseteq ([0,1]^d, 1)$.
2. Learner chooses action $\alpha_t \in \mathcal{A} \subseteq [-1,1]^{d+1}$ and commits to it.
3. Agent observes α_t, $\sigma_t = (\boldsymbol{x}_t, y_t)$, $y_t \in \{-1, 1\}$.
4. Agent reports $\boldsymbol{z}_t(\alpha_t; \boldsymbol{x}_t)$ (potentially different from $\boldsymbol{x}_t$).
5. Learner observes label of $\boldsymbol{z}_t(\alpha_t; \boldsymbol{x}_t)$ ($\hat{y}_t$); incurs binary classification loss:

$$\ell\big(\alpha_t, \boldsymbol{z}_t(\alpha_t; \boldsymbol{x}_t)\big) = 1\{\mathrm{sgn}(\hat{y}_t \cdot \langle \alpha_t, \boldsymbol{z}_t(\alpha_t; \boldsymbol{x}_t)\rangle) = -1\}.$$

Model (2) – Agent's Behavior

Intuition from a spamming email setting: $y_t = \hat{y}_t$

General form of utility functions + myopically rational:

$$u_t\big(\alpha, \boldsymbol{z}_t(\alpha_t; \boldsymbol{x}_t)\big) = \delta \cdot \underbrace{v_t\big(\alpha, \boldsymbol{z}_t(\alpha_t; \boldsymbol{x}_t)\big)}_{\text{Value} \in [0,1]} - \underbrace{\mathrm{c}_t\big(\alpha, \boldsymbol{z}_t(\alpha_t; \boldsymbol{x}_t)\big)}_{\text{Cost} \in [0,1]}$$

e.g., $1\{\langle \alpha, \boldsymbol{z}_t(\alpha; \boldsymbol{x}_t)\rangle \geq 0\} \cdot 1\{\hat{y}_t \neq y_t\}$

e.g., $\big(x_t - \boldsymbol{z}_t(\alpha; \boldsymbol{x}_t)\big)^2$

Model (1)

$$y_t = h^*(\boldsymbol{x}_t), \hat{y}_t = h^*(\boldsymbol{z}_t(\alpha_t; \boldsymbol{x}_t))$$

$$\boldsymbol{r}_t(\alpha_t) = \arg\max_{\boldsymbol{z}_t \in \mathcal{X}; \boldsymbol{x}_t} u_t(\alpha, \boldsymbol{z}_t(\alpha_t, \boldsymbol{x}_t))$$

Repeated Interaction Protocol

For round $t \in [T]$:

1. Environment chooses $\boldsymbol{x}_t \sim \mathcal{X} \subseteq ([0,1]^d, 1)$.
2. Learner chooses action $\alpha_t \in \mathcal{A} \subseteq [-1,1]^{d+1}$ and commits to it.
3. Agent observes α_t, $\sigma_t = (\boldsymbol{x}_t, y_t)$, $y_t \in \{-1, 1\}$.
4. Agent reports $\boldsymbol{r}_t(\alpha_t)$ (potentially different from $\boldsymbol{x}_t$).
5. Learner observes label of $\boldsymbol{r}_t(\alpha_t)$ ($\hat{y}_t$); incurs binary classification loss:

$$\ell(\alpha_t, \boldsymbol{r}_t(\alpha_t)) = 1\{\mathrm{sgn}(\hat{y}_t \cdot \langle \alpha_t, \boldsymbol{r}_t(\alpha_t) \rangle) = -1\}.$$

Learner's Goal: $\mathcal{R}(T) = \sum_{t=1}^{T} \ell(\alpha_t, \boldsymbol{r}_t(\alpha_t)) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha, \boldsymbol{r}_t(\alpha)) \leq o(T)$

Learner's Goal – Why Non-Trivial (1)

Minimize Stackelberg Regret

$$\mathcal{R}(T) = \sum_{t=1}^{T} \ell\big(\alpha_t, \boldsymbol{r}_t(\alpha_t)\big) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell\big(\alpha, \boldsymbol{r}_t(\alpha)\big) \leq o(T)$$

Minimize **external** regret $R(T) = \sum_{t=1}^{T} \ell\big(\alpha_t, \boldsymbol{r_t}(\alpha_t)\big) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell\big(\alpha, \boldsymbol{r_t}(\alpha_t)\big)$ and show how it relates to Stackelberg.

We prove that external and Stackelberg regret are **strongly incompatible**!

$$R(T) = o(T) \rightarrow \mathcal{R}(T) = \Omega(T) \text{ and } \mathcal{R}(T) = \mathrm{o}(T) \rightarrow R(T) = \Omega(T)$$

Learner's Goal – Why Non-Trivial (2)

Minimize Stackelberg Regret

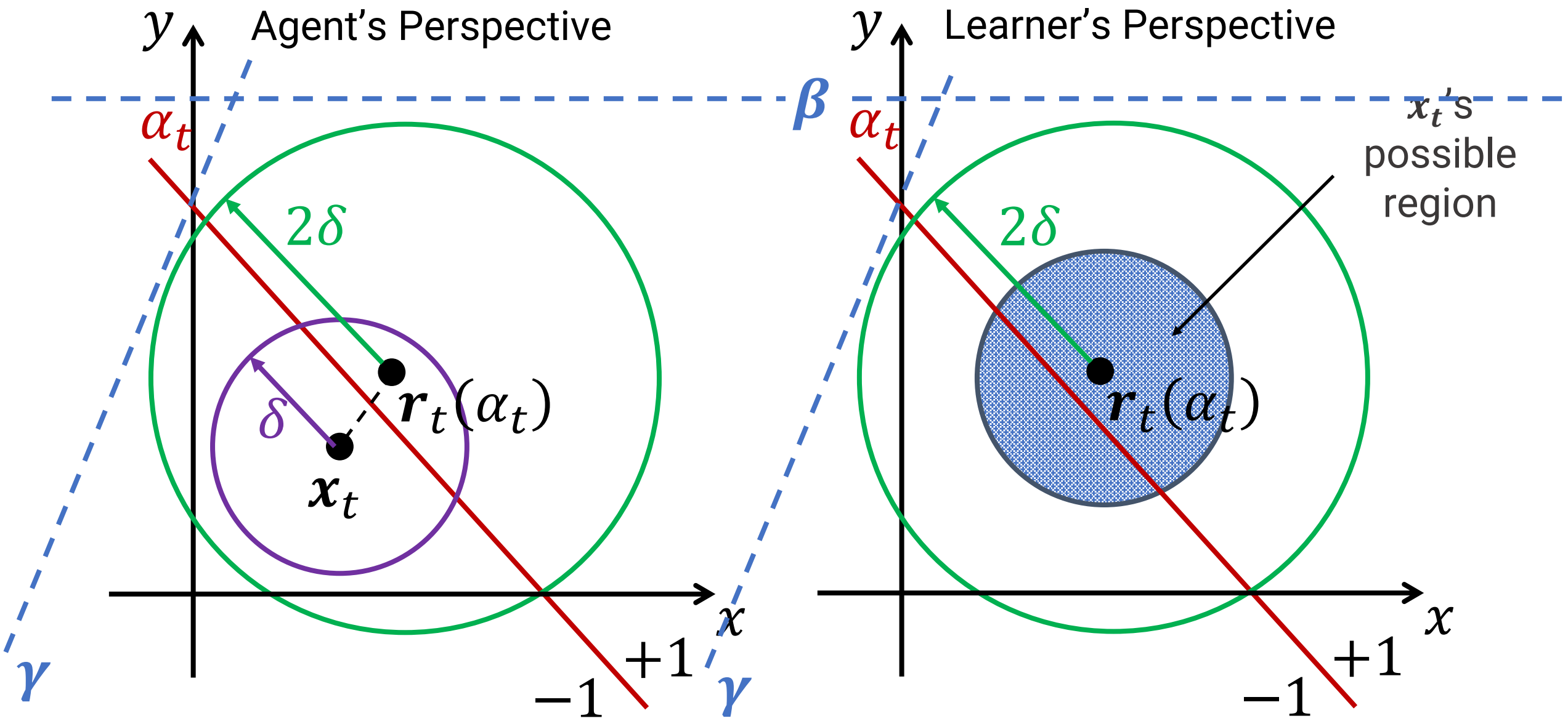
$$\mathcal{R}(T) = \sum_{t=1}^{T} \ell\big(\alpha_t, \boldsymbol{r}_t(\alpha_t)\big) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell\big(\alpha, \boldsymbol{r}_t(\alpha)\big) \leq o(T)$$

Minimize **external** regret $R(T) = \sum_{t=1}^{T} \ell\big(\alpha_t, \boldsymbol{r_t}(\alpha_t)\big) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell\big(\alpha, \boldsymbol{r_t}(\alpha_t)\big)$ and show how it relates to Stackelberg.

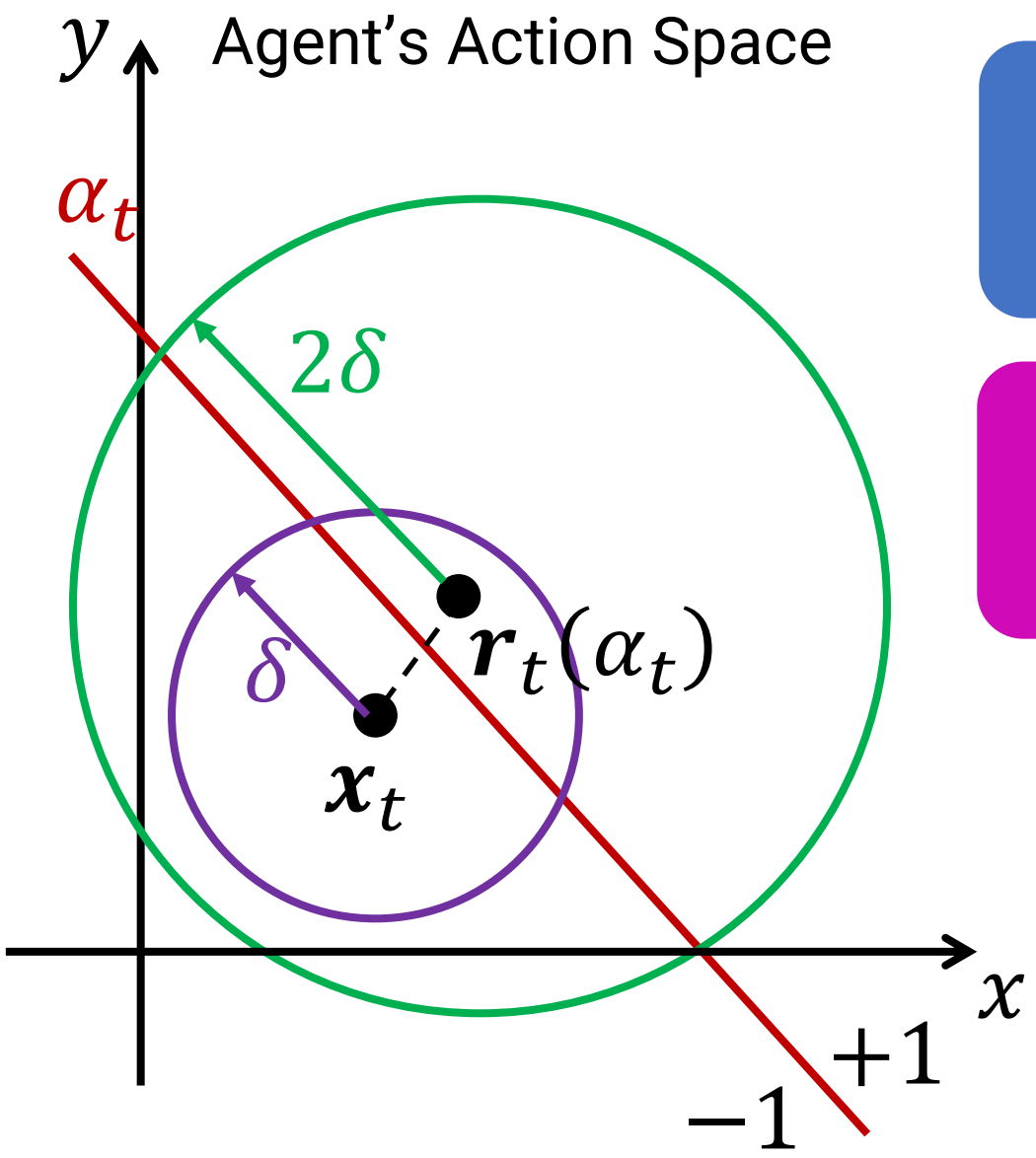
Minimize **function** $\ell\big(\alpha, \boldsymbol{r_t}(\alpha)\big)$.

$\ell\big(\alpha, \boldsymbol{r_t}(\alpha)\big)$ is **not** even Lipschitz

Inferring $\ell(\alpha', \boldsymbol{r}_t(\alpha'))$ For $\alpha' \neq \alpha_t$ Without Observing $\boldsymbol{x}_t$ (1)



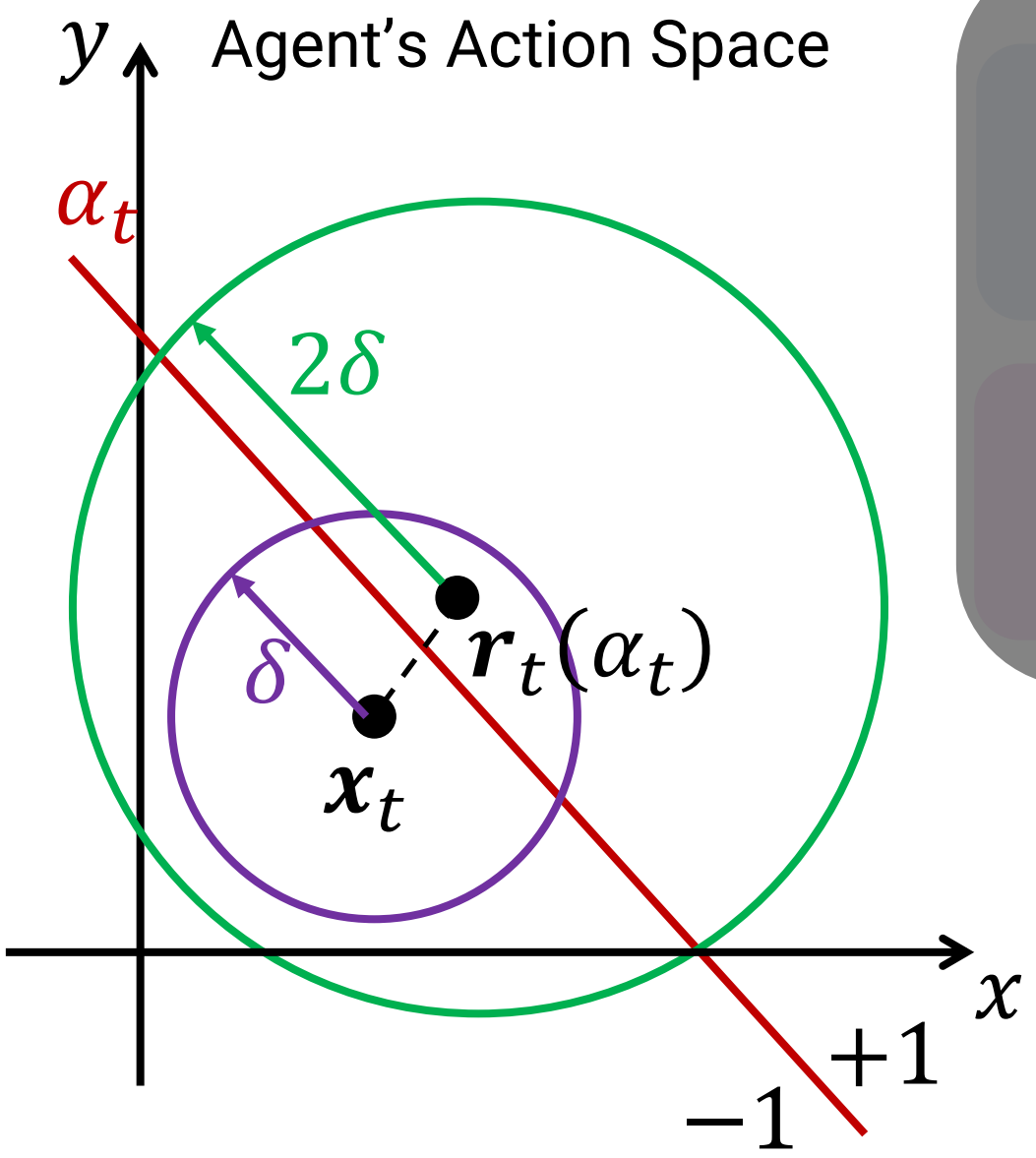
Inferring $\ell(\alpha', \boldsymbol{r}_t(\alpha'))$ For $\alpha' \neq \alpha_t$ Without Observing $\boldsymbol{x}_t$ (2)



$\forall \alpha'$ s.t., $\frac{\langle \alpha', \boldsymbol{r}_t(\alpha_t) \rangle}{\|\alpha'\|_2} \geq 2\delta$: estimated label is always 1

$\forall \alpha'$ s.t., $\frac{\langle \alpha', \boldsymbol{r}_t(\alpha_t) \rangle}{\|\alpha'\|_2} \leq -2\delta$: estimated label is always -1

Inferring $\ell(\alpha', \boldsymbol{r}_t(\alpha'))$ For $\alpha' \neq \alpha_t$ Without Observing $\boldsymbol{x}_t$ (2)



$\forall \alpha'$ s.t., $\frac{\langle \alpha', \boldsymbol{r}_t(\alpha_t) \rangle}{\|\alpha'\|_2} \geq 2\delta$: estimated label is always 1

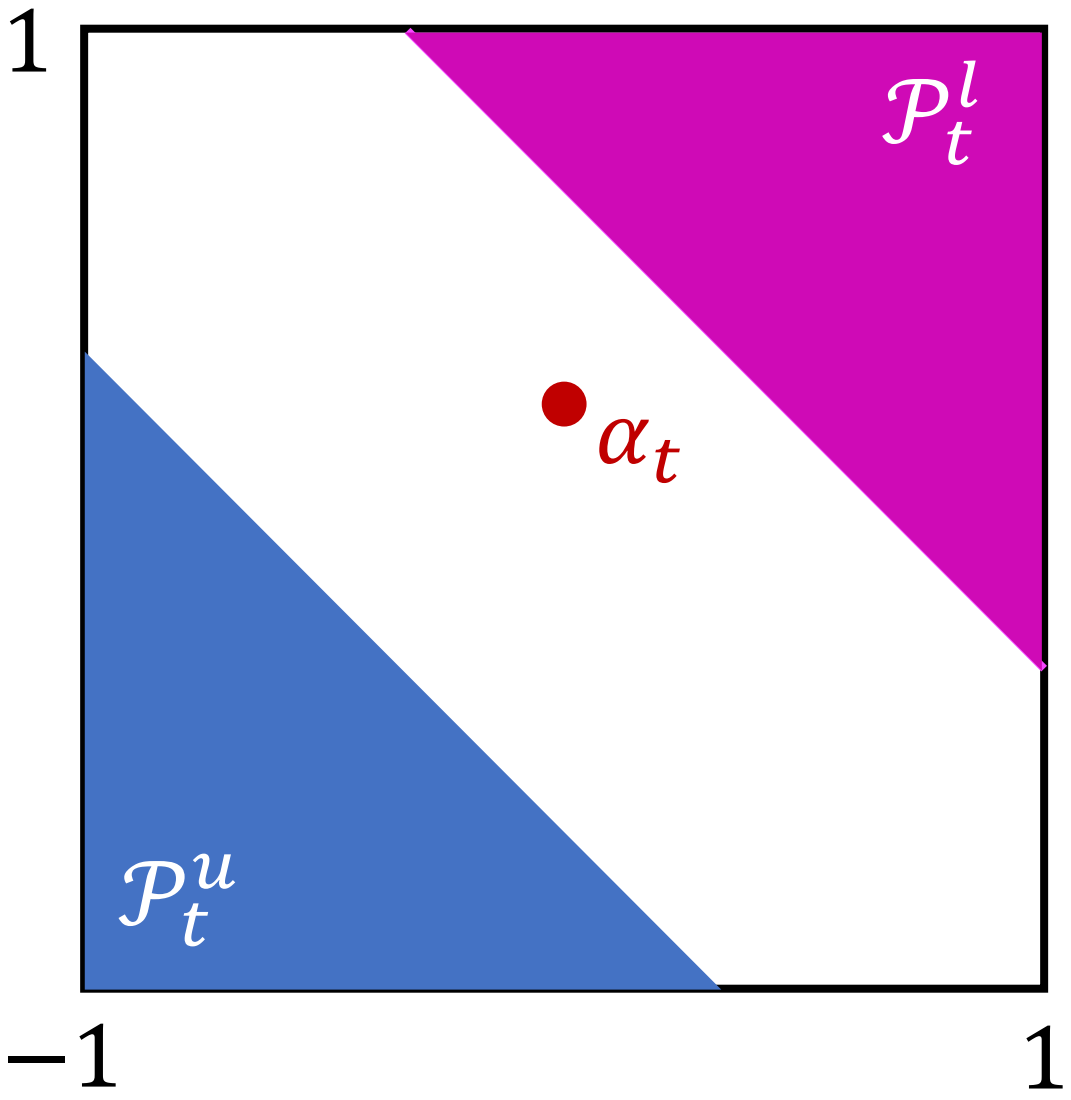
$\forall \alpha'$ s.t., $|\langle \alpha', \boldsymbol{r}_t(\alpha_t) \rangle| \geq 4\sqrt{d}\delta$

$\forall \alpha'$ s.t., $\frac{\langle \alpha', \boldsymbol{r}_t(\alpha_t) \rangle}{\|\alpha'\|_2} \leq -2\delta$: estimated label is always -1

Learner observes y_t

$\ell(\alpha', r_t(\alpha'))$

Translating to the Dual (i.e., the Learner's) Space



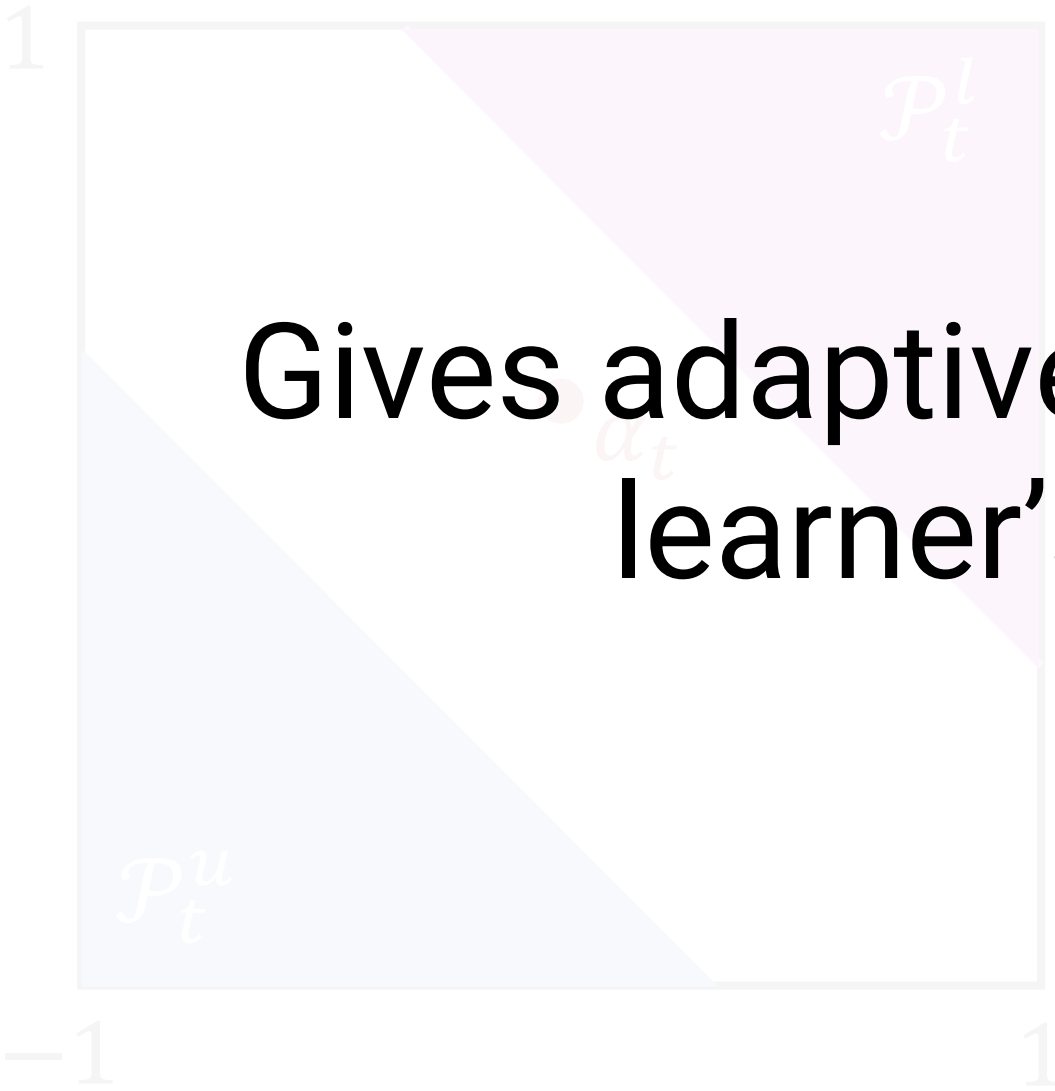
$\forall \alpha'$ s.t., $\langle \alpha', \boldsymbol{r_t}(\alpha_t) \rangle \geq 4\sqrt{d}\delta$:
estimated label is always 1

$\forall \alpha'$ s.t., $\langle \alpha', \boldsymbol{r_t}(\alpha_t) \rangle \leq -4\sqrt{d}\delta$:
estimated label is always -1

Learner observes y_t

$\ell(\alpha', r_t(\alpha'))$

Translating to the Dual (i.e., the Learner's) Space



$\forall a'$ s.t., $\langle a', r_t(a_t) \rangle \geq 4\sqrt{d}\delta$:
estimated label is always 1

$\forall a'$ s.t., $\langle a', r_t(a_t) \rangle \leq -4\sqrt{d}\delta$:
estimated label is always -1

Learner observes y_t

$\ell(a', r_t(a'))$

Gives adaptive discretization of the learner's action space!

Putting it Together; the GRINDER Algorithm

For all $t \in [T]$:

1. Choose polytope $p_t \sim \pi_t(p) = (1-\gamma)q_t(p) + \frac{\gamma\lambda(p)}{\lambda(\mathcal{A})}$

2. Commit to $\alpha_t \sim Unif(p_t)$

3. Observe $(\boldsymbol{r}_t(\alpha_t), y_t)$ & grind $\mathcal{A}$ to $\mathcal{P}_t^u, \mathcal{P}_t^l, \mathcal{P}_t^m$.

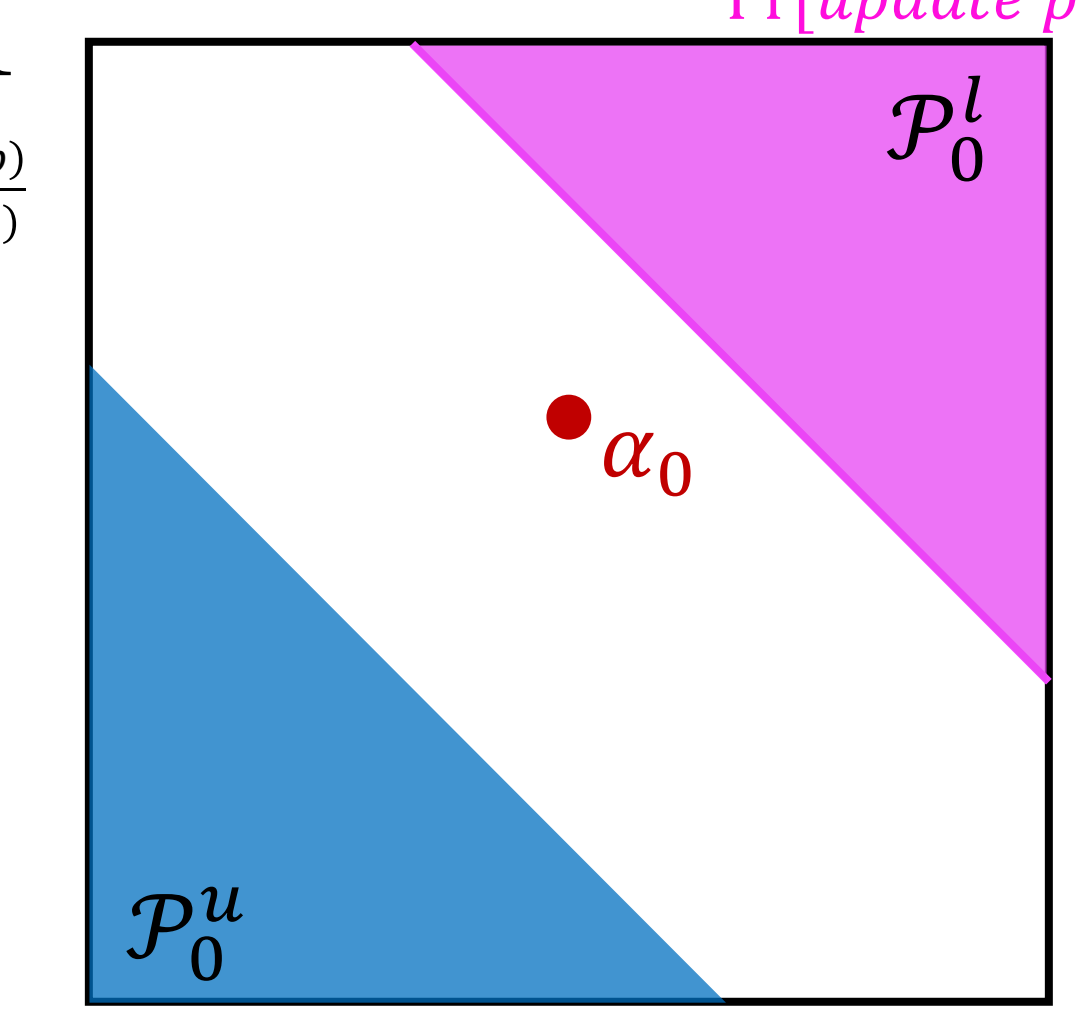
4. Update all polytopes with their IPS loss estimate

(note: upper, and lower → full info!)

5. Update sampling probabilities:

$$\forall p\colon\ q_{t+1}(p) \propto \lambda(p)\exp\left(-\eta\sum_{\tau=1}^{t}\hat{\ell}(p, \boldsymbol{r}_t(p))\right).$$

$$\hat{\ell}(p, r_t(p)) = \frac{\mathbf{1}\{y_t = 1\}}{\Pr[update\ p]}$$



$$\hat{\ell}(p, r_t(p)) = \frac{\mathbf{1}\{y_t = -1\}}{\Pr[update\ p]}$$

$\lambda(A)$: Lebesgue measure of space A

Putting it Together; the GRINDER Algorithm

For all $t \in [T]$:

1. Choose polytope $p_t \sim \pi_t(p) = (1-\gamma)q_t(p) + \frac{\gamma\lambda(p)}{\lambda(\mathcal{A})}$

2. Commit to $\alpha_t \sim Unif(p_t)$

3. Observe $(\boldsymbol{r}_t(\alpha_t), y_t)$ & grind $\mathcal{A}$ to $\mathcal{P}_t^u, \mathcal{P}_t^l, \mathcal{P}_t^m$.

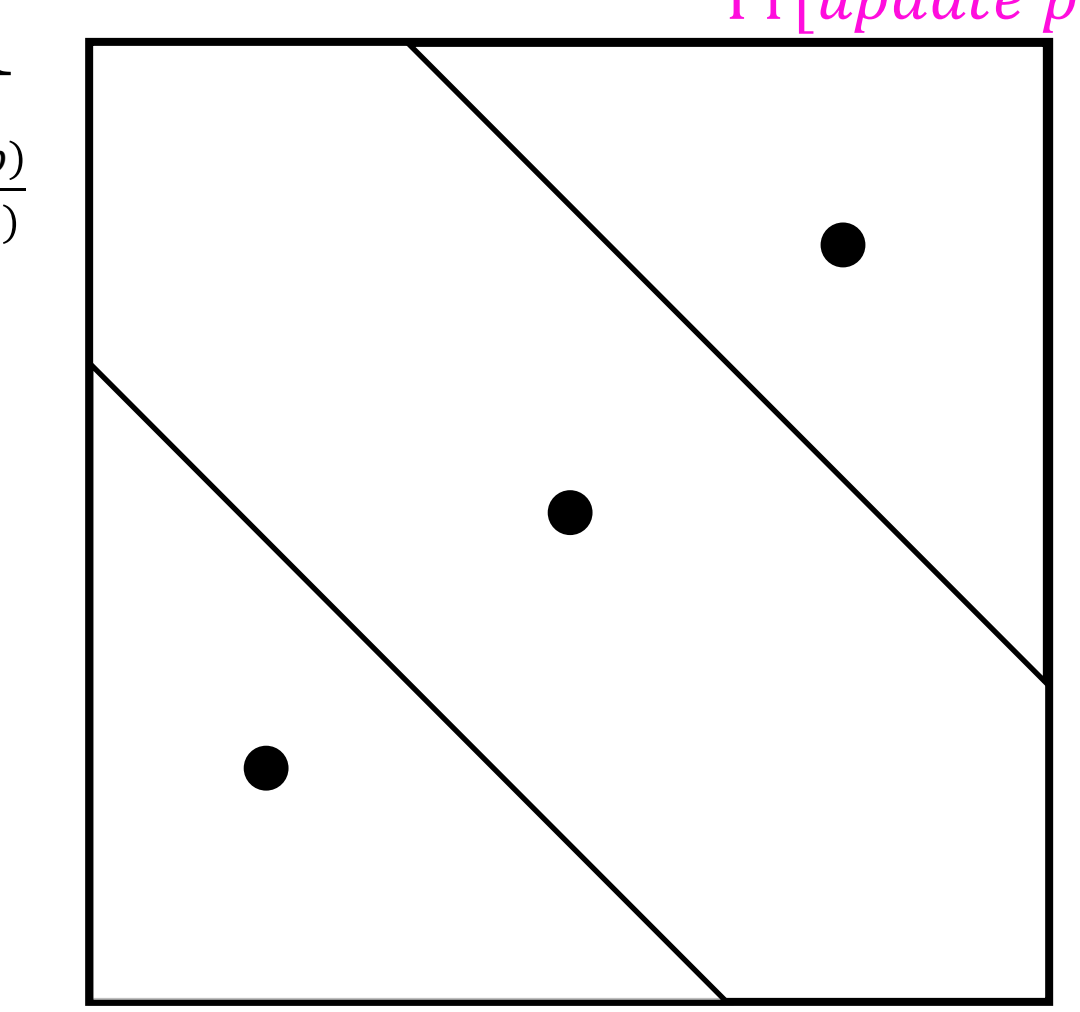
4. Update all polytopes with their IPS loss estimate

(note: upper, and lower → full info!)

5. Update sampling probabilities:

$$\forall p\colon\ q_{t+1}(p) \propto \lambda(p)\exp\left(-\eta\sum_{\tau=1}^{t}\hat{\ell}(p, \boldsymbol{r}_t(p))\right).$$

$$\hat{\ell}(p, r_t(p)) = \frac{\mathbf{1}\{y_t = 1\}}{\Pr[update\ p]}$$



$$\hat{\ell}(p, r_t(p)) = \frac{\mathbf{1}\{y_t = -1\}}{\Pr[update\ p]}$$

$\lambda(A)$: Lebesgue measure of space A

Putting it Together; the GRINDER Algorithm

For all $t \in [T]$:

1. Choose polytope $p_t \sim \pi_t(p) = (1-\gamma)q_t(p) + \frac{\gamma\lambda(p)}{\lambda(\mathcal{A})}$

2. Commit to $\alpha_t \sim Unif(p_t)$

3. Observe $(\boldsymbol{r}_t(\alpha_t), y_t)$ & grind $\mathcal{A}$ to $\mathcal{P}_t^u, \mathcal{P}_t^l, \mathcal{P}_t^m$.

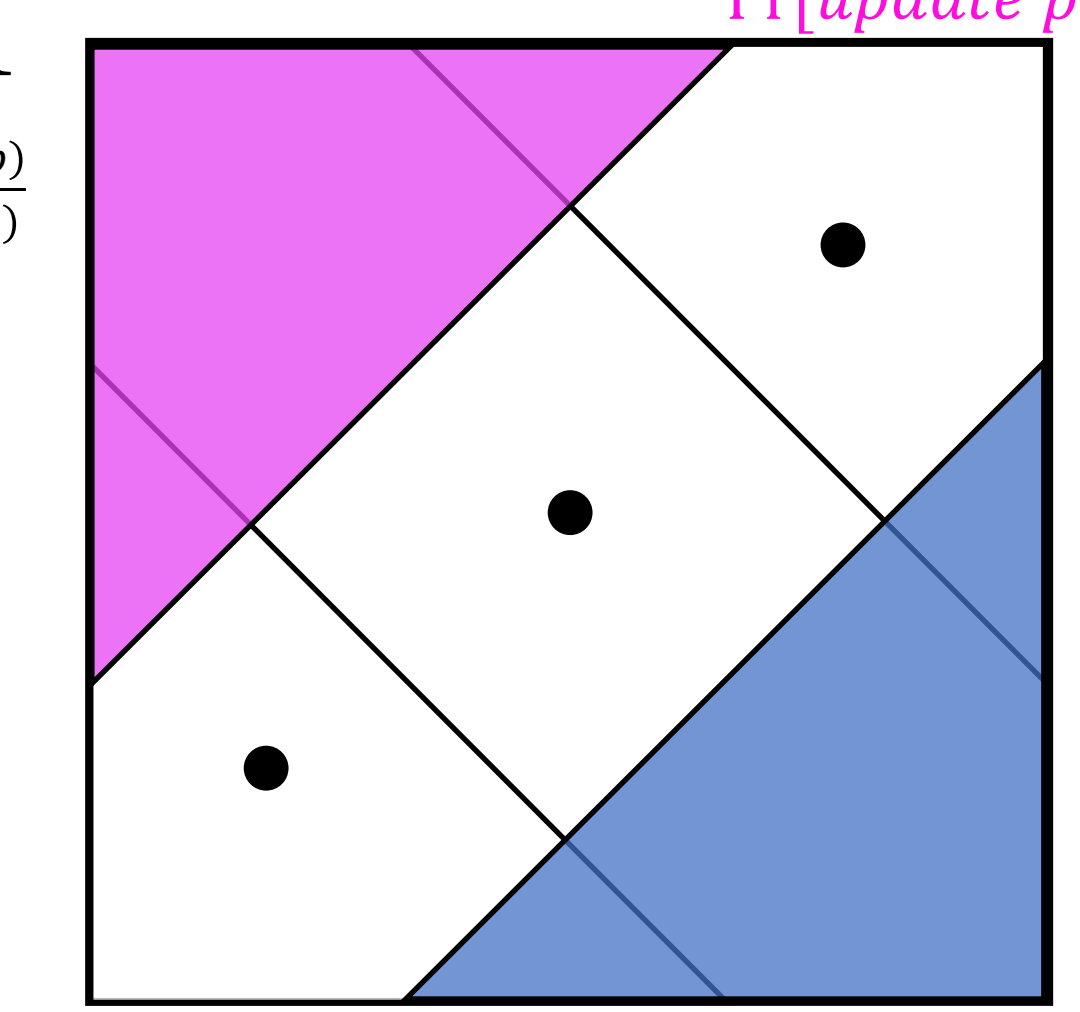
4. Update all polytopes with their IPS loss estimate

(note: upper, and lower → full info!)

5. Update sampling probabilities:

$$\forall p\colon\ q_{t+1}(p) \propto \lambda(p)\exp\left(-\eta\sum_{\tau=1}^{t}\hat{\ell}(p, \boldsymbol{r}_t(p))\right).$$

$$\hat{\ell}(p, r_t(p)) = \frac{\mathbf{1}\{y_t = 1\}}{\Pr[update\ p]}$$



$$\hat{\ell}(p, r_t(p)) = \frac{\mathbf{1}\{y_t = -1\}}{\Pr[update\ p]}$$

$\lambda(A)$: Lebesgue measure of space A

Putting it Together; the GRINDER Algorithm

For all $t \in [T]$:

1. Choose polytope $p_t \sim \pi_t(p) = (1-\gamma)q_t(p) + \frac{\gamma\lambda(p)}{\lambda(\mathcal{A})}$

2. Commit to $\alpha_t \sim Unif(p_t)$

3. Observe $(\boldsymbol{r}_t(\alpha_t), y_t)$ & grind $\mathcal{A}$ to $\mathcal{P}_t^u, \mathcal{P}_t^l, \mathcal{P}_t^m$.

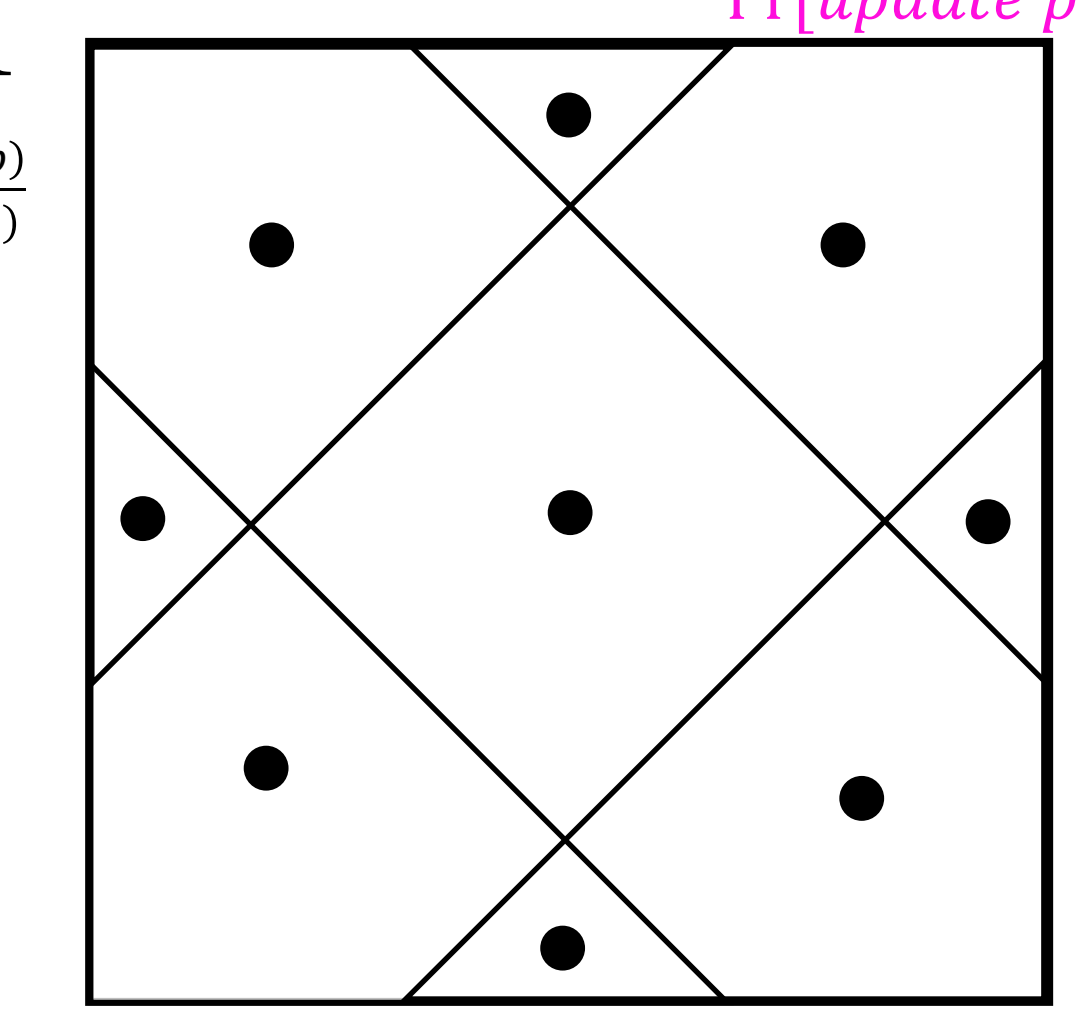
4. Update all polytopes with their IPS loss estimate

(note: upper, and lower → full info!)

5. Update sampling probabilities:

$$\forall p\colon\ q_{t+1}(p) \propto \lambda(p)\exp\left(-\eta\sum_{\tau=1}^{t}\hat{\ell}(p, \boldsymbol{r}_t(p))\right).$$

$$\hat{\ell}(p, r_t(p)) = \frac{\mathbf{1}\{y_t = 1\}}{\Pr[update\ p]}$$



$$\hat{\ell}(p, r_t(p)) = \frac{\mathbf{1}\{y_t = -1\}}{\Pr[update\ p]}$$

$\lambda(A)$: Lebesgue measure of space A

Putting it Together; the GRINDER Algorithm

For all $t \in [T]$:

1. Choose polytope $p_t \sim \pi_t(p) = (1-\gamma)q_t(p) + \frac{\gamma\lambda(p)}{\lambda(\mathcal{A})}$

2. Commit to $\alpha_t \sim Unif(p_t)$

3. Observe $(\boldsymbol{r}_t(\alpha_t), y_t)$ & grind $\mathcal{A}$ to $\mathcal{P}_t^u, \mathcal{P}_t^l, \mathcal{P}_t^m$.

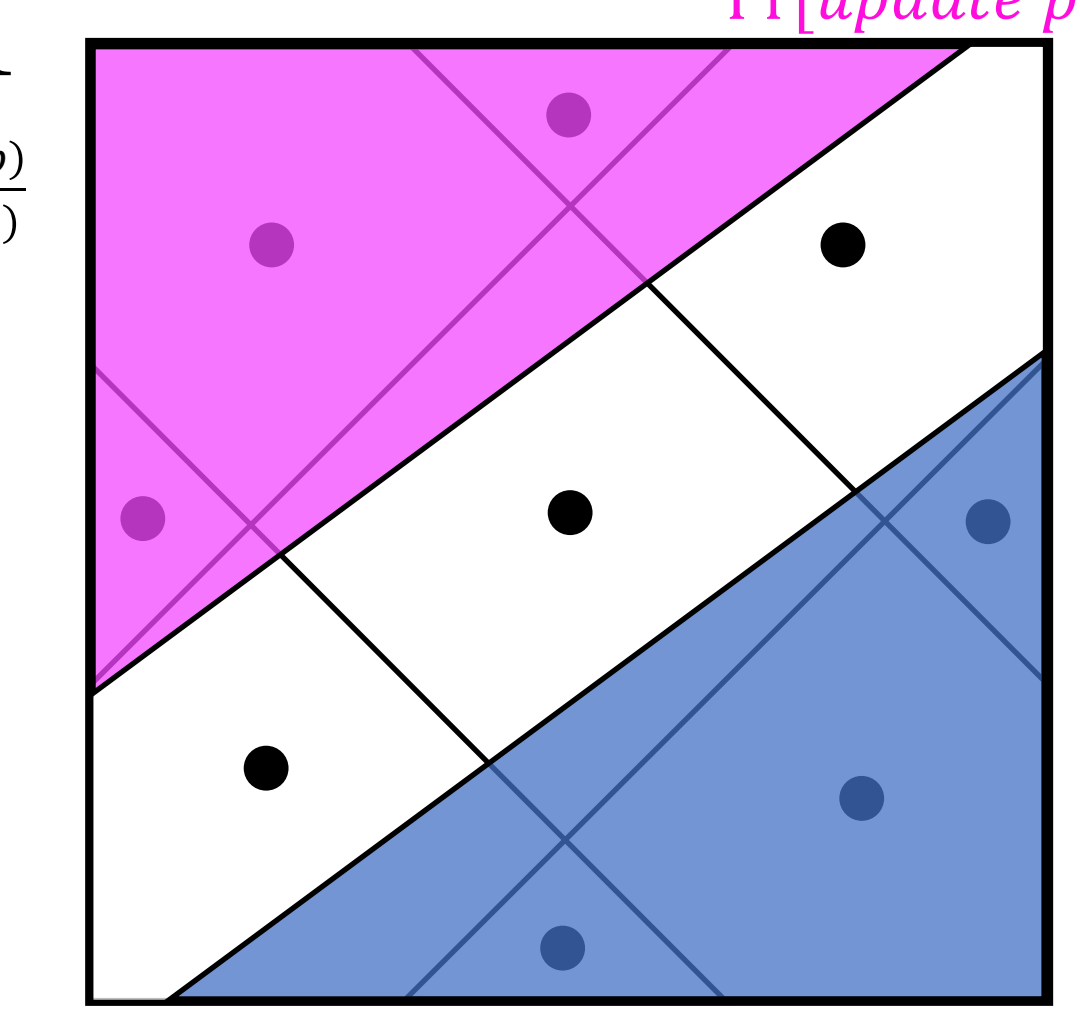
4. Update all polytopes with their IPS loss estimate

(note: upper, and lower → full info!)

5. Update sampling probabilities:

$$\forall p\colon\ q_{t+1}(p) \propto \lambda(p)\exp\left(-\eta \sum_{\tau=1}^{t} \hat{\ell}(p, \boldsymbol{r}_t(p))\right).$$

$$\hat{\ell}(p, r_t(p)) = \frac{\mathbf{1}\{y_t = 1\}}{\Pr[update\ p]}$$



$$\hat{\ell}(p, r_t(p)) = \frac{\mathbf{1}\{y_t = -1\}}{\Pr[update\ p]}$$

$\lambda(A)$: Lebesgue measure of space A

Putting it Together; the GRINDER Algorithm

For all $t \in [T]$:

1. Choose polytope $p_t \sim \pi_t(p) = (1-\gamma)q_t(p) + \frac{\gamma\lambda(p)}{\lambda(\mathcal{A})}$

2. Commit to $\alpha_t \sim Unif(p_t)$

3. Observe $(\boldsymbol{r}_t(\alpha_t), y_t)$ & grind $\mathcal{A}$ to $\mathcal{P}_t^u, \mathcal{P}_t^l, \mathcal{P}_t^m$.

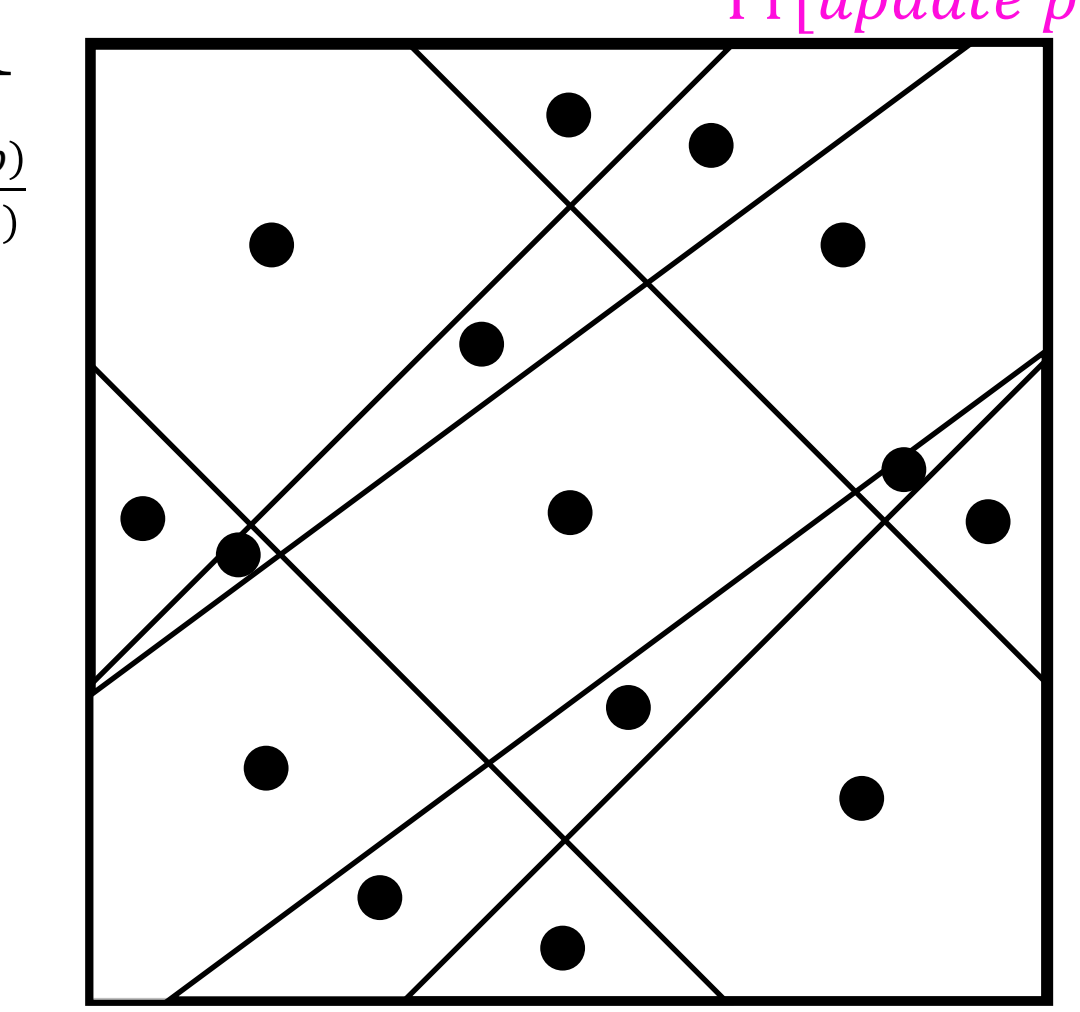
4. Update all polytopes with their IPS loss estimate

(note: upper, and lower → full info!)

5. Update sampling probabilities:

$$\forall p\colon\ q_{t+1}(p) \propto \lambda(p)\exp\left(-\eta\sum_{\tau=1}^{t}\hat{\ell}(p, \boldsymbol{r}_t(p))\right).$$

$$\hat{\ell}(p, r_t(p)) = \frac{\mathbf{1}\{y_t = 1\}}{\Pr[update\ p]}$$



$$\hat{\ell}(p, r_t(p)) = \frac{\mathbf{1}\{y_t = -1\}}{\Pr[update\ p]}$$

$\lambda(A)$: Lebesgue measure of space A

Results

GRINDER's regret: $O\left(\sqrt{\log\left(\frac{\lambda(\mathcal{A})\cdot T}{\lambda(p')}\right)\log\left(\frac{\lambda(\mathcal{A})}{\lambda(p')}\right)T}\right)$

$\lambda(S)$: Lebesgue measure of S
p': polytope with the smallest Lebesgue measure

Unfortunate dependence: $\mathcal{R}(T) \to \infty$ if $\log\left(\frac{\lambda(\mathcal{A})}{\lambda(p')}\right) \to 0$

But unavoidable, up to an extent…

The Stackelberg regret of learning a linear classifier against myopically rational δ - bounded strategic agents is: $\mathcal{R}(T) = \Omega\left(\sqrt{\log\left(\frac{\lambda(\mathcal{A})}{\lambda(p')}\right)\cdot T}\right)$

An Unfortunate Dependence

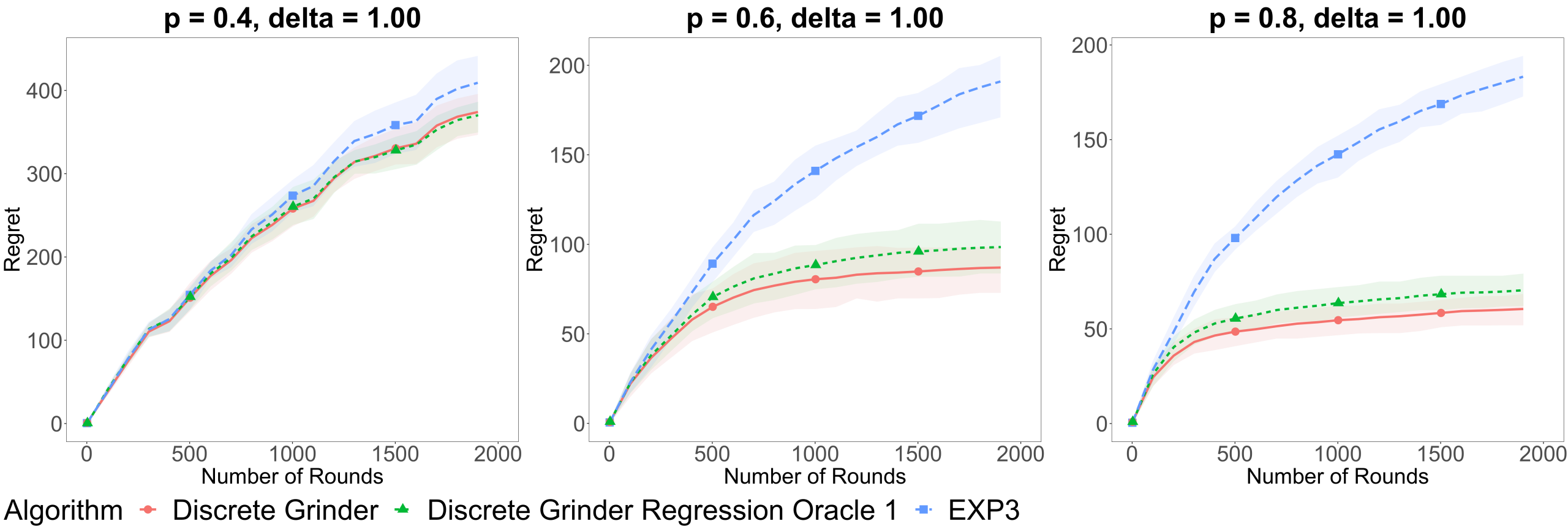
GRINDER: $\mathcal{R}(T) = O\left(\sqrt{\log\left(\frac{\lambda(\mathcal{A})\cdot T}{\lambda(p')}\right)\log\left(\frac{\lambda(\mathcal{A})}{\lambda(p')}\right)T}\right)$

$\mathcal{R}(T) \to \infty$ if $\log\left(\frac{\lambda(\mathcal{A})}{\lambda(p')}\right) \to 0$

But unavoidable, up to an extent…

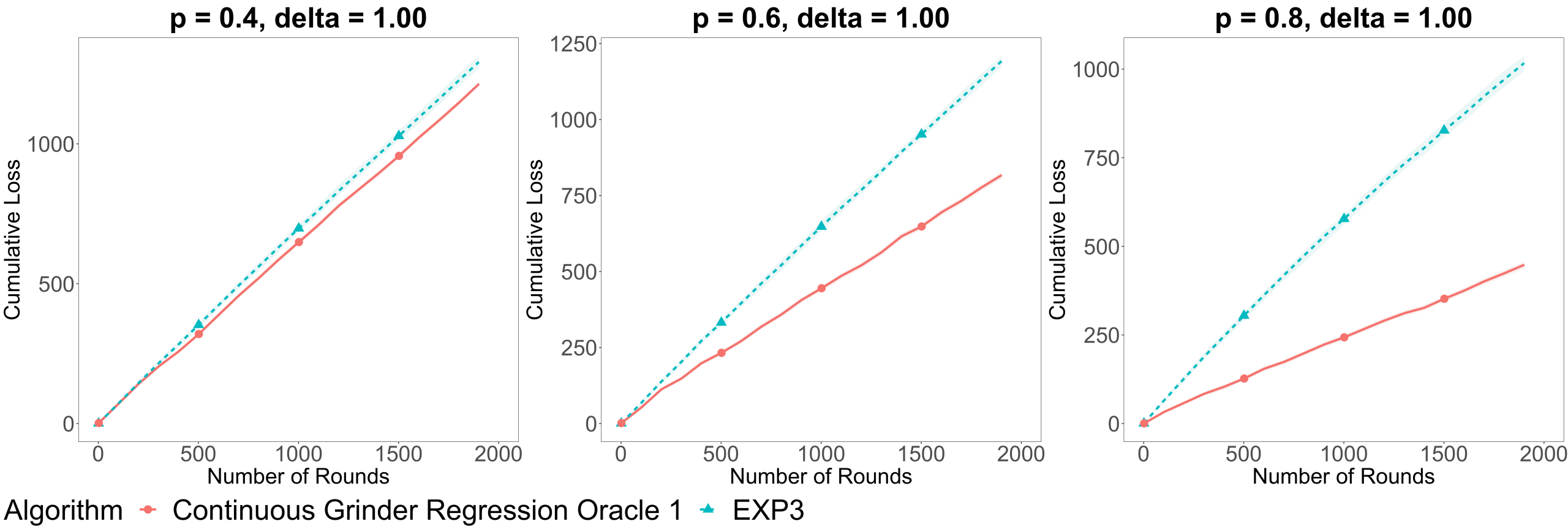
The Stackelberg regret of learning a linear classifier against myopically rational δ - bounded strategic agents is: $\mathcal{R}(T) = \Omega\left(\sqrt{\log\left(\frac{\lambda(\mathcal{A})}{\lambda(p')}\right)\cdot T}\right)$

Simulations – Discrete, Countable Action Set



In both cases, discrete GRINDER significantly outperforms EXP3, despite not having access to an omnipotent oracle.

Simulations – Continuous Action Set



In both cases, discrete GRINDER significantly outperforms EXP3, despite not having access to an omnipotent oracle.

Open Questions

- A lot of **technical** open questions directly related to our model in the paper. →
- More broadly:
  - Can we protect other ML tasks (like classification) against similar types of strategic agents?
  - What about non-myopic agents?
  - Connections with fairness literature: does strategic manipulation protection imply fair algorithms? Is the opposite true?


Thank You!

Proof Sketch (1)

$$\mathcal{R}(T) = O\left(\sqrt{\log\left(\frac{\lambda(\mathcal{A})\cdot T}{\lambda(p')}\right)\log\left(\frac{\lambda(\mathcal{A})}{\lambda(p')}\right) T}\right)$$

Distribution induced by 2-stage sampling process at round t: $\mathcal{D}_t$

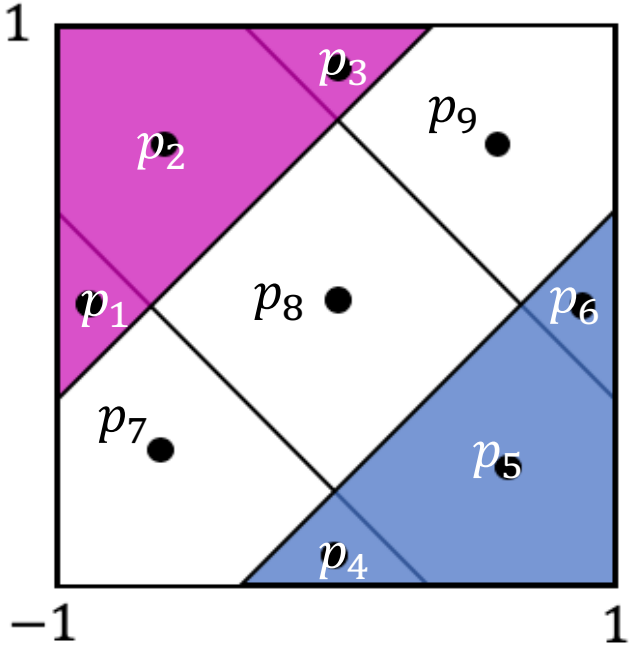
1. Unbiased Estimator:
   $E_{\mathcal{D}_t}\left[\hat{\ell}\left(\alpha, \boldsymbol{r_t}(\alpha)\right)\right] = \ell\left(\alpha, \boldsymbol{r_t}(\alpha)\right)$
2. Upper bound for 2nd moment:
   $E_{\mathcal{D}_t}\left[\hat{\ell}\left(\alpha, \boldsymbol{r_t}(\alpha)\right)^2\right] \leq \frac{1}{\Pr[\alpha]}$ in

Reasoning about the way you update $\hat{\ell}\left(\alpha, \boldsymbol{r_t}(\alpha)\right)$ according to $\mathcal{D}_t$

Proof Sketch (2)

Define polytope sets:

- $P_t \in \mathcal{P}^u_{t,\sigma_t}, if\ \forall \alpha \in P_t: \langle \alpha, \boldsymbol{x_t} \rangle \geq 4\sqrt{d}\,\delta$
- $P_t \in \mathcal{P}^l_{t,\sigma_t}, if\ \forall \alpha \in P_t: \langle \alpha, \boldsymbol{x_t} \rangle \leq -4\sqrt{d}\,\delta$
- $P_t \in \mathcal{P}^m_{t,\sigma_t}, if\ P_t \in \mathcal{P}_t \ \setminus (\mathcal{P}^u_{t,\sigma_t} \cup \mathcal{P}^l_{t,\sigma_t})$



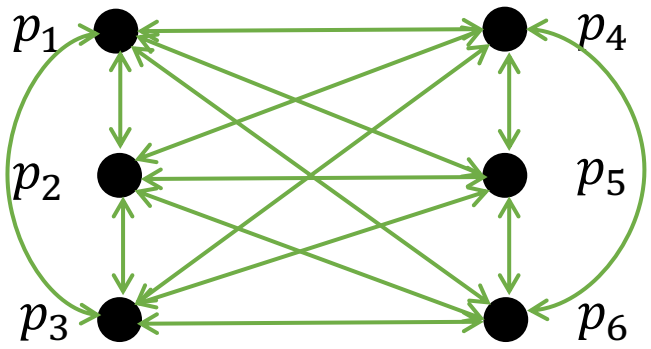
3. Bound the variance of our estimator

$$\pi_t(p) = (1-\gamma)q_t(p) + \frac{\gamma\lambda(p)}{\lambda(\mathcal{A})}$$

$$E_{\mathcal{D}_t}\left[\frac{1}{\Pr^{in}[\alpha_t]}\right] \leq 4\log\left(\frac{4\lambda(\mathcal{A})\cdot\left|\mathcal{P}^u_{t,\sigma_t} \cup \mathcal{P}^l_{t,\sigma_t}\right|}{\gamma\lambda(p')}\right) + \lambda(\mathcal{P}^m_{t,\sigma_t})$$

[Alon, Cesa-Bianchi, Dekel, Koren '15]: For weighted feedback graph with weights $w_i \geq \epsilon, \sum_i w_i \leq 1$:

$$\sum_{i\in[K]} \frac{w_i}{w_i + \sum_{j\in N^{in}(i)} w_j} \leq 4\alpha^G \ln\left(\frac{4K}{\alpha^G\,\epsilon}\right)$$



Related Work

- **Learning using data from strategic data sources:**
  - Strategic Classification: [Meir, Almagor, Michaely, Rosenschein '11], [Meir, Procaccia, Rosenschein '12], [Hardt, Meggido, Papadimitriou, Wooters '16], [Dong, Roth, Schutzman, Waggoner, Wu '18]
  - Strategic Linear Regression: [Perote, Perote-Pena '04], [Dekel, Fischer, Procaccia '10], [Cummings, Ioannidis, Ligett '15], [Cai, Daskalakis, Papadimitriou '15], [Chen, **P.** , Procaccia, Shah '18], [Hossain, Shah '19], [Ben-Porat, Tennenholtz '19]
  - Learning from revealed preferences: [Beigman and Vohra, '06], [Zadimoghaddam, Roth '12], [Balcan, Daniely, Mehta, Urner, Vazirani '14], [Roth, Ullman, Wu '16], [Jabbari, Rogers, Roth, Wu '16]…
- **Multi-Armed Bandits:**
  - Standard Models: [Cesa Bianchi, Bubeck '12], [Slivkins '19], [Lattimore, Szepesvari '19] and many more
  - Strategic Considerations: [Braverman, Mao, Schneider, Weinberg '19]
- **Learning in Stackelberg Security Games:** [Blum, Haghtalab, Procaccia '14], [Balcan, Blum, Haghtalab, Procaccia '15], [Peng, Shen, Tang, Zuo '19]