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)
Three Ways AI Will Impact The Lending Industry

Jeff Levin
Forbes Councils Member
Forbes Real Estate Council

Jeff Levin is a real estate investor & founder of Specialty Lending Group, a private lending firm based in Washington, DC.
Three Ways AI Will Impact The Lending Industry

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

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
Three Ways AI Will Impact The Lending Industry

Jeff Levin
Forbes Councils Member
Forbes Real Estate Council
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.

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
Three Ways AI Will Impact The Lending Industry

Jeff Levin

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

How artificial intelligence can help us make judges less biased

Predicting which applicant is the best candidate

Applying for Your Next Job May Be an Automated Nightmare

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
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** (report ~ Distribution)
- **Strategic** (report ~ utility maximizer)
- **Adversarial** (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)

Repeated Interaction Protocol

For round $t \in [T]$:

1. Environment chooses $x_t \sim \mathcal{X} \subseteq ([0,1]^d, 1)$.

2. Learner chooses action $\alpha_t \in \mathcal{A} \subseteq [-1, 1]^{d+1}$ and commits to it.

3. Agent observes $\alpha_t, \sigma_t = (x_t, y_t), y_t \in \{-1, 1\}$.

4. Agent reports $z_t(\alpha_t; x_t)$ (potentially different from $x_t$).

5. Learner observes label of $z_t(\alpha_t; x_t)$ ($\hat{y}_t$); incurs binary classification loss:

$$\ell(\alpha_t, z_t(\alpha_t; x_t)) = 1\{\text{sgn}(\hat{y}_t \cdot \langle \alpha_t, z_t(\alpha_t; x_t) \rangle) = -1\}.$$
Model (2) – Agent’s Behavior

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

General form of utility functions + myopically rational:

$$u_t(\alpha, z_t(\alpha; x_t)) = \delta \cdot v_t(\alpha, z_t(\alpha; x_t)) - c_t(\alpha, z_t(\alpha; x_t))$$

Value $\in [0,1]$  
Cost $\in [0,1]$  

e.g., $1\{\langle \alpha, z_t(\alpha; x_t) \rangle \geq 0\} \cdot 1\{\hat{y}_t \neq y_t\}$  
e.g., $(x_t - z_t(\alpha; x_t))^2$
Model (1)

Repeated Interaction Protocol

For round $t \in [T]$:

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

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

Learner’s Goal: $\mathcal{R}(T) = \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \min_{\alpha \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha, r_t(\alpha)) \leq o(T)$
Learner’s Goal – Why Non-Trivial (1)

Minimize Stackelberg Regret

\[ R(T) = \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \min_{a \in A} \sum_{t=1}^{T} \ell(\alpha, r_t(\alpha)) \leq o(T) \]

Minimize external regret

\[ R(T) = \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \min_{a \in A} \sum_{t=1}^{T} \ell(\alpha, r_t(\alpha)) \]

and show how it relates to Stackelberg.

We prove that external and Stackelberg regret are strongly incompatible!

\[ R(T) = o(T) \rightarrow R(T) = \Omega(T) \text{ and } R(T) = o(T) \rightarrow R(T) = \Omega(T) \]
Minimize Stackelberg Regret

\[
\mathcal{R}(T) = \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \min_{\alpha \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha, r_t(\alpha)) \leq o(T)
\]

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

- Minimize **function** \( \ell(\alpha, r_t(\alpha)) \).

\[\ell(\alpha, r_t(\alpha)) \text{ is not even Lipschitz}\]
Inferring $\ell(\alpha', r_t(\alpha'))$ For $\alpha' \neq \alpha_t$ Without Observing $x_t$ (1)
Inferring $\ell(\alpha', r_t(\alpha'))$ For $\alpha' \neq \alpha_t$ Without Observing $x_t$ (2)

Agent’s Action Space

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

$\forall \alpha' \text{ s.t.}, \frac{\langle \alpha', r_t(\alpha_t) \rangle}{\|\alpha'\|_2} \leq -2\delta$: estimated label is always $-1$
Inferring $\ell(\alpha', r_t(\alpha'))$ For $\alpha' \neq \alpha_t$ Without Observing $x_t$ (2)

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

∀ $\alpha'$ s.t., $|\langle \alpha', r_t(\alpha_t) \rangle| \geq 4\sqrt{d}\delta$

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

For $\alpha' \neq \alpha_t$ Without Observing $x_t$ (2)
Translating to the Dual (i.e., the Learner’s) Space

∀ \alpha \text{ s.t., } \langle \alpha, r_t(\alpha_t) \rangle \geq 4\sqrt{d}\delta: estimated label is always 1

∀ \alpha' \text{ s.t., } \langle \alpha', 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

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

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

Gives adaptive discretization of the learner’s action space!
Putting it Together; the GRINDER Algorithm

For all \( t \in [T] \):

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

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

3. Observe \((r_t(\alpha_t), y_t)\) & 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:

\[
\forall p: q_{t+1}(p) \propto \lambda(p) \exp(-\eta \sum_{t=1}^{t} \hat{\ell}(p, r_t(p))).
\]

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

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

\( \lambda(A) \): Lebesgue measure of space \( A \)
Putting it Together; the GRINDER Algorithm

For all $t \in [T]$:

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

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

3. Observe $(r_t(\alpha_t), y_t)$ & grind $A$ to $P_t^u, P_t^l, P_t^m$.

4. Update all polytopes with their IPS loss estimate
   (note: upper, and lower $\rightarrow$ full info!)

5. Update sampling probabilities:
   
   $\forall p$: $q_{t+1}(p) \propto \lambda(p) \exp(-\eta \sum_{t=1}^{t} \hat{\ell}(p, r_t(p)))$.

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

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

$\lambda(A)$: Lebesgue measure of space $A$
Putting it Together; the GRINDER Algorithm

For all $t \in [T]$:

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

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

3. Observe $(r_t(\alpha_t), y_t)$ & grind $A$ 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 $\rightarrow$ full info!)

5. Update sampling probabilities:
   \[ \forall p: q_{t+1}(p) \propto \lambda(p) \exp(-\eta \sum_{t=1}^t \hat{\ell}(p, r_t(p))) . \]

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

$\hat{\ell}(p, r_t(p)) = \frac{\mathbb{1}_{y_t = -1}}{\Pr[\text{update } p]}$
Putting it Together; the GRINDER Algorithm

For all $t \in [T]$:  

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

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

3. Observe $(r_t(\alpha_t), y_t)$ & grind $A$ to $P_t^u, P_t^l, P_t^m$.

4. Update all polytopes with their IPS loss estimate (note: upper, and lower $\rightarrow$ full info!)

5. Update sampling probabilities:

   $\forall p$: $q_{t+1}(p) \propto \lambda(p) \exp(-\eta \sum_{t=1}^{t} \hat{\ell}(p, r_t(p)))$.

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

$\hat{\ell}(p, r_t(p)) = \frac{1\{y_t = -1\}}{\Pr[update \ p]}$
Putting it Together; the GRINDER Algorithm

For all $t \in [T]$:

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

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

3. Observe $(r_t(\alpha_t), y_t)$ & grind $A$ to $\mathcal{P}^u_t, \mathcal{P}^l_t, \mathcal{P}^m_t$.

4. Update all polytopes with their IPS loss estimate (note: upper, and lower $\rightarrow$ full info!)

5. Update sampling probabilities:

$$\forall p: q_{t+1}(p) \propto \lambda(p) \exp(-\eta \sum_{t=1}^{\hat{\ell}(p, r_t(p))} \hat{\ell}(p, r_t(p))).$$

$\hat{\ell}(p, r_t(p)) = \frac{1\{y_t = 1\}}{\Pr[\text{update } p]}$
Putting it Together; the GRINDER Algorithm

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

For all \( t \in [T] \):

1. Choose polytope \( p_t \sim \pi_t(p) = (1 - \gamma)q_t(p) + \frac{\gamma \lambda(p)}{\lambda(A)} \)
2. Commit to \( \alpha_t \sim \text{Unif}(p_t) \)
3. Observe \((r_t(\alpha_t), y_t)\) & grind \( A \) to \( P^u_t, P^l_t, P^m_t \).
4. Update all polytopes with their IPS loss estimate (note: upper, and lower \( \rightarrow \) full info!)
5. Update sampling probabilities:
   \[
   \forall p: \quad q_{t+1}(p) \propto \lambda(p) \exp(-\eta \sum_{t=1}^t \hat{\ell}(p, r_t(p))).
   \]

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

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

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

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

But unavoidable, up to an extent...

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

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

$\mathcal{R}(T) \to \infty$ 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 $\delta$-bounded strategic agents is: $\mathcal{R}(T) = \Omega \left( \sqrt{\log \left( \frac{\lambda(\mathcal{A})}{\lambda(p')} \right) \cdot T} \right)$
In both cases, discrete GRINDER significantly outperforms EXP3, despite not having access to an omnipotent oracle.
In both cases, discrete GRINDER significantly outperforms EXP3, despite not having access to an omnipotent oracle.
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)

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

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

1. Unbiased Estimator:
   \[ E_{\mathcal{D}_t}[\hat{\ell}(\alpha, r_t(\alpha))] = \ell(\alpha, r_t(\alpha)) \]

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

Reasoning about the way you update \( \hat{\ell}(\alpha, r_t(\alpha)) \) according to \( \mathcal{D}_t \)
Define polytope sets:

- \( P_t \in \mathcal{P}^u_{t, \sigma_t} \), if \( \forall \alpha \in P_t: \langle \alpha, 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, 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

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

[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, 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]