Instance-Optimal PAC Contextual Bandits

Zhaoqi Li*, Lillian Ratliff*, Houssam Nassif†, Kevin Jamieson*, Lalit Jain*

*University of Washington

†Amazon
Motivation
Motivation

Nov - Dec
Upcoming Sale
Motivation

Nov - Dec
Upcoming Sale

Icon of a computer screen with a graphic promoting an upcoming sale. Icons of a person, a guitar, and a basketball on the right side.
Motivation
Question: What is the best way to give personalized recommendations?
Motivation

Question: What is the best way to give personalized recommendations?
Contextual Bandit Setting

- At each time $t = 1, 2, \ldots$:
  - Context $c_t \in C$ arrives, $c_t \sim \nu \in \Delta_C$
  - Choose action $a_t \in A$
  - Receive reward $r_t$, $\mathbb{E}[r_t | c_t, a_t] = r(c_t, a_t) \in \mathbb{R}$

- Policy class $\Pi$, each $\pi \in \Pi$, $\pi : C \rightarrow A$
- Average reward: $V(\pi) := \mathbb{E}_{c \sim \nu}[r(c, \pi(c))]$
- Optimal policy: $\pi_* := \arg \max_{\pi \in \Pi} V(\pi)$

$(\epsilon, \delta) - \text{PAC Guarantee}$

Return $\hat{\pi}$ satisfying, $V(\hat{\pi}) \geq V(\pi_*) - \epsilon$ with probability greater than $1 - \delta$ in a minimum number of samples.
Regret Minimization vs. Policy Identification
Regret Minimization vs. Policy Identification

• Regret heavily studied:

\[ R_T = \sum_{t=1}^{T} r(c_t, \pi^*(c_t)) - r(c_t, a_t) \]
Regret Minimization vs. Policy Identification

- Regret heavily studied:
  \[ R_T = \sum_{t=1}^{T} r(c_t, \pi_\star(c_t)) - r(c_t, a_t) \]

- ILOVETOCONBANDITS [Agarwal et al. 2014] achieves \( R_T = O(\sqrt{|A|T \log(\Pi)}) \), computationally efficient
Regret Minimization vs. Policy Identification

- Regret heavily studied:

\[ R_T = \sum_{t=1}^{T} r(c_t, \pi_\ast(c_t)) - r(c_t, a_t) \]

- ILOVETOCONBANDITS [Agarwal et al. 2014] achieves \( R_T = O(\sqrt{|A|T \log(\Pi)}) \), computationally efficient

- Modification gives \((\epsilon, \delta)\)-PAC algorithm w/ sample complexity \( O(|A| \log(\Pi/\delta)/\epsilon^2) \), also see [Zanette et al. 2021]
Regret Minimization vs. Policy Identification

• Regret heavily studied:
  \[ R_T = \sum_{t=1}^{T} r(c_t, \pi_t(c_t)) - r(c_t, a_t) \]

• ILOVETOCONBANDITS [Agarwal et al. 2014] achieves \( R_T = O(\sqrt{|A|T \log(\Pi)}) \), computationally efficient

• Modification gives \((\epsilon, \delta)\)- PAC algorithm w/ sample complexity \( O(|A| \log(\Pi/\delta)/\epsilon^2) \), also see [Zanette et al. 2021]

Two Problems

a) **Minimax** Result! Does not adapt to hardness of instance.
Regret Minimization vs. Policy Identification

- Regret heavily studied:
  \[ R_T = \sum_{t=1}^{T} r(c_t, \pi_t(c_t)) - r(c_t, a_t) \]

- ILOVETOCONBANDITS [Agarwal et al. 2014] achieves \( R_T = O(\sqrt{|A| T \log(\Pi)}) \), computationally efficient

- Modification gives \((\epsilon, \delta)\)-PAC algorithm w/ sample complexity \( O(\sqrt{|A| \log(\Pi/\delta)/\epsilon^2}) \), also see [Zanette et al. 2021]

Two Problems

a) **Minimax** Result! Does not adapt to hardness of instance.
Regret Minimization vs. Policy Identification

• Regret heavily studied:

\[ R_T = \sum_{t=1}^{T} r(c_t, \pi_*(c_t)) - r(c_t, a_t) \]

• ILOVETOCONBANDITS [Agarwal et al. 2014] achieves \( R_T = O(\sqrt{|A| T \log(\Pi)}) \), computationally efficient

• Modification gives \((\epsilon, \delta)\)-PAC algorithm w/ sample complexity \( O(|A| \log(\Pi/\delta)/\epsilon^2) \), also see [Zanette et al. 2021]

Two Problems

a) Minimax Result! Does not adapt to hardness of instance.

b) Can construct an example, where any optimal regret algorithm won’t be instance optimal!
Challenges

- What is the statistical limits of learning, i.e. the instance-dependent lower bound?
- Can we design sampling procedure to achieve this?
- Computational efficiency - context space $\mathcal{C}$ and policy space $\Pi$ could be infinite!
Challenges

• What is the statistical limits of learning, i.e. the *instance-dependent* lower bound?

• Can we design sampling procedure to achieve this?

• Computational efficiency - context space $C$ and policy space $\Pi$ could be *infinite*!
Challenges

• What is the statistical limits of learning, i.e. the instance-dependent lower bound?

• Can we design sampling procedure to achieve this?

• Computational efficiency - context space $C$ and policy space $\Pi$ could be infinite!

**Question:** what is possible?
Our Contribution

• Show the first instance-dependent lower bound for PAC contextual bandit

• Present a simple algorithm that achieves this lower bound

• Design a computational efficient algorithm that also achieves this lower bound
Towards Lower Bound: Estimators

- Linear contextual bandit setting (agnostic setting could be reduced to linear setting):
  - feature map: \( \phi : C \times A \rightarrow \mathbb{R}^d \) such that \( r(c, a) = \langle \phi(c, a), \theta^* \rangle \) for \( \theta^* \in \Theta \subseteq \mathbb{R}^d \)
  
- Given dataset \( \mathcal{D} = \{(c_t, a_t, r_t)\}_{t=1}^n \) where \( a_t \sim p_{c_t} \in \Delta_A \),

\[
\mathbb{E}[\phi(c_t, a_t)r_t] = \mathbb{E}_{c,a}[\phi(c, a)\phi(c, a)^\top \theta^*] = \sum_c \nu_c \sum_a p_{c,a} \phi(c, a)\phi(c, a)^\top \theta^* 
\]
Towards Lower Bound: Estimators

- Linear contextual bandit setting (agnostic setting could be reduced to linear setting):
  - feature map: $\phi : \mathcal{C} \times \mathcal{A} \rightarrow \mathbb{R}^d$ such that $r(c, a) = \langle \phi(c, a), \theta^* \rangle$ for $\theta^* \in \Theta \subset \mathbb{R}^d$
  - Given dataset $\mathcal{D} = \{(c_t, a_t, r_t)\}_{t=1}^n$ where $a_t \sim p_{c_t} \in \Delta_A$,

$$
\mathbb{E}[\phi(c_t, a_t)r_t] = \mathbb{E}_{c, a}[\phi(c, a)\phi(c, a)^\top \theta^*] = \sum_{c} \nu_c \sum_{a} p_{c,a} \phi(c, a)\phi(c, a)^\top \theta^*
$$

$A(p)$
Towards Lower Bound: Estimators

• Linear contextual bandit setting (agnostic setting could be reduced to linear setting):

  • feature map: \( \phi : C \times A \to \mathbb{R}^d \) such that \( r(c, a) = \langle \phi(c, a), \theta^* \rangle \) for \( \theta^* \in \Theta \subset \mathbb{R}^d \)

  • Given dataset \( D = \{(c_t, a_t, r_t)\}_{t=1}^n \) where \( a_t \sim p_{c_t} \in \Delta_A \),

\[
\mathbb{E}[\phi(c_t, a_t)r_t] = \mathbb{E}_{c,a}[\phi(c, a)\phi(c, a)^\top \theta^*] = \sum_c \nu_c \sum_a p_{c,a} \phi(c, a)\phi(c, a)^\top \theta^* \\
\Rightarrow \hat{\theta} = \frac{1}{n} A(p)^{-1} \sum_{t=1}^n \phi(c_t, a_t)r_t
\]
Towards Lower Bound: Estimators

- Linear contextual bandit setting (agnostic setting could be reduced to linear setting):
  - feature map: \( \phi : C \times A \rightarrow \mathbb{R}^d \) such that \( r(c, a) = \langle \phi(c, a), \theta^* \rangle \) for \( \theta^* \in \Theta \subseteq \mathbb{R}^d \)

- Given dataset \( D = \{(c_t, a_t, r_t)\}_{t=1}^n \) where \( a_t \sim p_{c_t} \in \Delta_A \),

\[
\mathbb{E}[\phi(c_t, a_t)r_t] = \mathbb{E}_{c,a}[\phi(c, a)\phi(c, a)^\top \theta^*] = \sum_c \nu_c \sum_a p_{c,a} \phi(c, a)\phi(c, a)^\top \theta^* = \sum_c \frac{\nu_c}{\text{A}(p)_c} \sum_a p_{c,a} \phi(c, a)^\top \theta^* = \sum_c \frac{\nu_c}{\text{A}(p)_c} \phi(c) \quad \text{A(p)}
\]

\[
\therefore \hat{\theta} = \frac{1}{n} A(p)^{-1} \sum_{t=1}^n \phi(c_t, a_t) r_t
\]

IPS estimate!
A Lower Bound
A Lower Bound

• For each $\pi \in \Pi$, define the gap $\Delta(\pi) := V(\pi^*) - V(\pi)$
A Lower Bound

• For each \( \pi \in \Pi \), define the gap \( \Delta(\pi) := V(\pi^*) - V(\pi) \)

• Let \( \phi_{\pi} := \mathbb{E}_{c \sim \nu}[\phi(c, \pi(c))] \), an estimate \( \hat{\Delta}(\pi) = \hat{V}(\pi^*) - \hat{V}(\pi) = \langle \phi_{\pi^*} - \phi_{\pi}, \hat{\theta} \rangle \)

\[
Var(\hat{\Delta}(\pi)) = (\phi_{\pi^*} - \phi_{\pi})^T Var(\hat{\theta}) (\phi_{\pi^*} - \phi_{\pi}) = \frac{\|\phi_{\pi^*} - \phi_{\pi}\|_A^2}{n} (p)^{-1}
\]
A Lower Bound

• For each $\pi \in \Pi$, define the gap $\Delta(\pi) := V(\pi_*) - V(\pi)$

• Let $\phi_\pi := \mathbb{E}_{c \sim \nu}[\phi(c, \pi(c))]$, an estimate $\hat{\Delta}(\pi) = \hat{V}(\pi_*) - \hat{V}(\pi) = \left\langle \phi_{\pi_*} - \phi_\pi, \hat{\theta} \right\rangle$

$$Var(\hat{\Delta}(\pi)) = (\phi_{\pi_*} - \phi_\pi)\top Var(\hat{\theta})(\phi_{\pi_*} - \phi_\pi) = \frac{\|\phi_{\pi_*} - \phi_\pi\|^2_{A(p)^{-1}}}{n}$$

**Theorem [Li et al. 2022]** Let $\tau$ be the stopping time of the algorithm. Any $(0, \delta)$-PAC algorithm satisfies $\tau \geq \rho_{\Pi,0} \log(1/2.4\delta)$ with high probability where

$$\rho_{\Pi,0} = \min_{p_c \in \Delta_A} \max_{\forall c \in C, \pi \in \Pi \setminus \pi_*} \frac{\|\phi_{\pi_*} - \phi_\pi\|^2_{A(p)^{-1}}}{\Delta(\pi)^2}.$$
Our algorithm
Our algorithm
Our algorithm

Input: II
Our algorithm

Input: $\Pi$
Initialize $\Pi_1 = \Pi$
Our algorithm

**Input:** \( \Pi \)

Initialize \( \Pi_1 = \Pi \)

for \( l = 1,2,\ldots \)
Input: Π
Initialize Π₁ = Π
for l = 1, 2, ...
    1. Choose \( p_c^{(l)} \in \Delta_A \), \( \forall c \in C \) and \( n_l \) such that
Our algorithm

Input: $\Pi$
Initialize $\Pi_1 = \Pi$
for $l = 1, 2, \ldots$

1. Choose $p_{c}^{(l)} \in \Delta_A, \forall c \in C$ and $n_l$ such that

$$\min_{p_c \in \Delta_A, \forall c \in C} \max_{\pi \in \Pi} \left( -\Delta(\pi) + \sqrt{\frac{\|\phi - \phi_{\hat{\pi}_{l-1}}\|^2_{A(P)^{-1}} \log(1/\delta)}{n_l}} \right) \leq 2^{-l}$$
Our algorithm

Input: \( \Pi \)
Initialze \( \Pi_1 = \Pi \)
for \( l = 1, 2, \ldots \)

1. Choose \( p_c^{(l)} \in \Delta_A, \forall c \in C \) and \( n_l \) such that

\[
\min_{p_c \in \Delta_A} \max_{\forall c \in C, \pi \in \Pi} \left( -\Delta(\pi) + \sqrt{n_l \left\| \phi_\pi - \phi_{\hat{\pi}_{l-1}} \right\|^2_{A(p)^{-1}} \log(1/\delta)} \right) \leq 2^{-l}
\]
**Our algorithm**

**Input:** $\Pi$

Initialize $\Pi_1 = \Pi$

for $l = 1,2,\ldots$

1. Choose $p_c^{(l)} \in \Delta_A$, $\forall c \in C$ and $n_l$ such that

\[
\min_{p_c \in \Delta_A, \forall c \in C} \max_{\pi \in \Pi} \left( -\Delta(\pi) + \sqrt{\frac{\|\phi_{c} - \phi_{\hat{n}_{l-1}}\|_A^2}{n_l} \log(1/\delta)} \right) \leq 2^{-l}
\]

2. For $t \in [n_l]$, for each context $c_t$, sampling $a_t \sim p_c^{(l)}$ and compute IPS estimate $\hat{\Delta}(\pi, \hat{n}_{l-1})$ for each $\pi \in \Pi$
Our algorithm

**Input:** $\Pi$

 Initialize $\Pi_1 = \Pi$

 for $l = 1, 2, \ldots$

 1. Choose $p_c^{(l)} \in \Delta_A$, $\forall c \in C$ and $n_l$ such that

     $$\min_{p_c \in \Delta_A, \forall c \in C} \max_{\pi \in \Pi} \left( -\Delta(\pi) + \frac{\sqrt{||\phi_\pi - \phi_{\pi_l-1}||_A^2 \log(1/\delta)}}{n_l} \right) \leq 2^{-l}$$

 2. For $t \in [n_l]$, for each context $c_t$, sampling $a_t \sim p_{c_t}^{(l)}$ and compute IPS estimate $\hat{\Delta}(\pi, \hat{\pi}_{l-1})$ for each $\pi \in \Pi$

 3. Update
Our algorithm

Input: Π

Initialize Π₁ = Π
for l = 1, 2, ...

1. Choose \( p_c^{(l)} \in \Delta_A, \forall c \in C \) and \( n_l \) such that

\[
\min_{p_c \in \Delta_A, \forall c \in C} \max_{\pi \in \Pi} \left( -\Delta(\pi) + \sqrt{\frac{\|\phi_{\pi} - \phi_{\hat{\pi}_{l-1}}\|^2}{n_l} \log(1/\delta)} \right) \leq 2^{-l}
\]

2. For \( t \in [n_l] \), for each context \( c_t \), sampling \( a_t \sim p_c^{(l)} \) and compute IPS estimate \( \hat{\Delta}(\pi, \hat{\pi}_{l-1}) \) for each \( \pi \in \Pi \)

3. Update

\[
\hat{\pi}_l = \arg \min_{\pi \in \Pi} \hat{\Delta}(\pi, \hat{\pi}_{l-1})
\]
Our algorithm

Input: $\Pi$
Initialize $\Pi_1 = \Pi$
for $l = 1, 2, \ldots$

1. Choose $p_c^{(l)} \in \Delta_A$, $\forall c \in C$ and $n_l$ such that

$$\min_{p_c \in \Delta_A, \forall c \in C} \max_{\pi \in \Pi} \left( -\Delta(\pi) + \sqrt{\frac{\|\phi_\pi - \phi_{\hat{\pi}_{l-1}}\|_{A(p)}^2 \log(1/\delta)}{n_l}} \right) \leq 2^{-l}$$

2. For $t \in [n_l]$, for each context $c_t$, sampling $a_t \sim p^{(l)}_{c_t}$ and compute IPS estimate

$$\hat{\Delta}(\pi, \hat{\pi}_{l-1})$$

for each $\pi \in \Pi$

3. Update

$$\hat{\pi}_l = \arg \min_{\pi \in \Pi} \hat{\Delta}(\pi, \hat{\pi}_{l-1})$$

Theorem [Li et al. 2022] The above algorithm returns an $(\epsilon, \delta)$-PAC policy with at most $O(\rho_{\Pi,\epsilon} \log(|\Pi|/\delta) \log_2(1/\epsilon))$ samples.
Our algorithm
Our algorithm
Our algorithm

\[ \varepsilon_1 = 2^{-1} \]
Our algorithm

\[-\Delta(\pi)\]

\[\epsilon_2 = 2^{-2}\]
Our algorithm

For \( \varepsilon \)-good policy, our criteria forces the estimation to be \( 2\varepsilon \)-good!
Our algorithm

For $\epsilon$-good policy, our criteria forces the estimation to be $2\epsilon$-good!

Returning the empirical best policy at the end $\Rightarrow$ at least $2\epsilon$-good
Towards an efficient algorithm

**Input:** \( \Pi \)

Initialize \( \Pi_1 = \Pi \)

for \( l = 1, 2, \ldots \):

1. Choose \( p_c^{(l)} \) and \( n_l \) such that

\[
\min_{p_c \in \Delta_A} \max_{\forall c \in \mathcal{C}} \max_{\pi \in \Pi} \left( -\Delta(\pi) + \sqrt{\frac{\|\phi_{\pi} - \phi_{\hat{\pi}_{l-1}}\|^2}{n_l}} \right) \leq 2^{-l}
\]

2. For \( t \in [n_l] \), for each context \( c_t \), sampling \( a_t \sim p_c^{(l)} \) and compute IPS estimate \( \hat{\Delta}(\pi, \hat{\pi}_{l-1}) \) for each \( \pi \in \Pi \)

3. Update

\[
\hat{\pi}_l = \arg\min_{\pi \in \Pi} \hat{\Delta}(\pi, \hat{\pi}_{l-1})
\]
Towards an efficient algorithm

**Input:** \( \Pi \)

Initialize \( \Pi_1 = \Pi \)

for \( l = 1, 2, \ldots \)

1. Choose \( p_c^{(l)} \) and \( n_l \) such that

\[
\min_{p_c \in \Delta, \forall c \in C} \max_{\pi \in \Pi} \left( -\Delta(\pi) + \sqrt{\frac{\|\phi_{\pi} - \phi_{\hat{\pi}_{l-1}}\|_A^2}{n_l} \log(1/\delta)} \right) \leq 2^{-l}
\]

2. For \( t \in [n_l] \), for each context \( c_t \), sampling \( a_t \sim p_c^{(l)} \) and compute IPS estimate \( \hat{\Delta}(\pi, \hat{\pi}_{l-1}) \) for each \( \pi \in \Pi \)

3. Update

\[
\hat{\pi}_l = \arg \min_{\pi \in \Pi} \hat{\Delta}(\pi, \hat{\pi}_{l-1})
\]
Towards an efficient algorithm

**Input:** $\Pi$

Initialize $\Pi_1 = \Pi$

for $l = 1, 2, \ldots$

1. Choose $p_c^{(l)}$ and $n_l$ such that

   \[
   \min_{p_c \in \Delta_\mathcal{X}, \forall c \in \mathcal{C}} \max_{\pi \in \Pi} \left( -\Delta(\pi) + \sqrt{\left\| \phi_\pi - \hat{\phi}_{l-1}\right\|^2 n_l \left(1/\delta \right)} \right) \leq 2^{-l}
   \]

2. For $t \in [n_l]$, for each context $c_t$, sampling $a_t \sim p_c^{(l)}$ and compute IPS estimate $\hat{\Delta}(\pi, \hat{\pi}_{l-1})$ for each $\pi \in \Pi$

3. Update

   \[
   \hat{\pi}_l = \arg\min_{\pi \in \Pi} \hat{\Delta}(\pi, \hat{\pi}_{l-1})
   \]

---

not efficient since cannot hold on to $p_c$ for all $c$!
Dual Problem

- Consider the dual formulation:

\[
\text{Primal} \quad \min_{p \in \Delta_n, \forall c \in C} \max_{\pi \in \Lambda} \Delta(\pi, \pi^*) - \Delta(\pi, \pi^*) + \sqrt{\frac{||\phi_{\pi} - \phi_{\pi^*}||^2_{A(p)^{-1}} \log(1/\delta)}} \leq n
\]


**Dual Problem**

- Consider the dual formulation:

\[
\begin{align*}
\text{Primal} & \quad \min_{p_c \in \Delta_A, \forall c \in C} \max_{\pi \in \Pi} -\Delta(\pi, \pi^*) + \sqrt{\frac{\|\phi_\pi - \phi_{\pi^*}\|_{A(p)}^{-1} \log(1/\delta)}{n}} \\
\text{Dual} & \quad \max_{\lambda \in \Delta_\Pi, \gamma_\pi \geq 0} \min_{p_c \in \Delta_A, \forall c \in C} \min_{\pi \in \Pi} \sum_{\pi \in \Pi} \lambda_\pi \left( -\Delta(\pi, \pi^*) + \gamma_\pi \|\phi_\pi - \phi_{\pi^*}\|_{A(p)}^{-1} + \frac{\log(1/\delta)}{2\gamma_\pi n} \right).
\end{align*}
\]
Compute Action Distribution

- If we solve for $p_c$ for all $c$, we have an analytical solution:

$$\min_{p_c \in \Delta_a, \forall c \in C} \sum_{\pi \in \Pi} \lambda_{\pi} \gamma_{\pi} \| \phi_{\pi} - \phi_{\pi_c} \|_{\lambda(p)}^{-1} = E_{c \sim \nu} \left[ \left( \sum_{a \in A} \sqrt{\sum_{\pi \in \Pi} \lambda_{\pi} \gamma_{\pi} (1\{\pi(c) = a\} + 1\{\pi_s(c) = a\} - 21\{\pi(c) = \pi_s(c)\})} \right)^2 \right]$$

$$= E_{c \sim \nu} \left[ \left( \sum_{a \in A} \sqrt{(\lambda \odot \gamma)^T t_a^{(c)}} \right)^2 \right]$$
Compute Action Distribution

- If we solve for $p_c$ for all $c$, we have an analytical solution:

$$
\min_{p_c \in \Delta, \forall c \in C} \sum_{\pi \in \Pi} \lambda_{\pi} \gamma_{\pi} \| \phi_{\pi} - \phi_{\pi_{\gamma}} \|^2_{\gamma(p)^{-1}} = \mathbb{E}_{c \sim \nu} \left[ \left( \sum_{a \in A} \sqrt{\sum_{\pi \in \Pi} \lambda_{\pi} \gamma_{\pi} (1 \{ \pi(c) = a \} + 1 \{ \pi_{\gamma}(c) = a \} - 21 \{ \pi(c) = \pi_{\gamma}(c) \})} \right)^2 \right]
$$

$$
= \mathbb{E}_{c \sim \nu} \left[ \left( \sum_{a \in A} (\lambda \gamma)^{T_{\pi_{\gamma}}(c)} \right)^2 \right]
$$

Implicitly maintain $p_c$ for all $c \in C$ simultaneously!
Compute Action Distribution

- If we solve for $p_c$ for all $c$, we have an analytical solution:

$$
\min_{p_c \in \Delta, \forall c \in C} \sum_{\pi \in \Pi} \lambda^c \gamma^c \| \phi_{\pi} - \phi_{\pi^*} \|^2 = \mathbb{E}_{c \sim \nu} \left[ \left( \sum_{a \in A} \sqrt{\sum_{\pi \in \Pi} \lambda^c \gamma^c (1 \{ \pi(c) = a \} + 1 \{ \pi^*(c) = a \} - 21 \{ \pi(c) = \pi^*(c) \})} \right)^2 \right]
$$

$$
= \mathbb{E}_{c \sim \nu} \left[ \left( \sum_{a \in A} \sqrt{\lambda \gamma^T t_a^{(c)}} \right)^2 \right]
$$

Implicitly maintain $p_c$ for all $c \in C$ simultaneously!

- Dual becomes

$$
\max_{\lambda \in \Delta_{\Pi}} \min_{\gamma \in \Pi} \sum_{\pi \in \Pi} \lambda^c \left( -\Delta(\pi, \pi^*) + \frac{\log(1/\delta)}{\gamma^c n} \right) + \mathbb{E}_{c \sim \nu} \left[ \left( \sum_{a \in A} \sqrt{\lambda \gamma^T t_a^{(c)}} \right)^2 \right]
$$
Compute Action Distribution

• If we solve for \( p_c \) for all \( c \), we have an analytical solution:

\[
\min_{p_c \in \Delta, \forall c \in C} \sum_{\pi \in \Pi} \lambda_{\pi} \gamma_{\pi} \sum_{\pi \in \Pi} \lambda_{\pi} \gamma_{\pi} \|\phi_{\pi} - \phi_{\pi,\lambda(p)}\|^2 = \mathbb{E}_{c \sim \nu} \left[ \left( \sum_{a \in A} \sqrt{\sum_{\pi \in \Pi} \lambda_{\pi} \gamma_{\pi} (\mathbf{1}_{\{\pi(c) = a\}} + \mathbf{1}_{\pi_{s}(c) = a} - 2 \mathbf{1}_{\{\pi(c) = \pi_{s}(c)\}})} \right)^2 \right] \\
=: \mathbb{E}_{c \sim \nu} \left[ \left( \sum_{a \in A} (\lambda \otimes \gamma)^{T} t_{a}^{(c)} \right)^2 \right] \\
\text{Implicitly maintain } p_c \text{ for all } c \in C \text{ simultaneously!}
\]

• Dual becomes

\[
\max \min_{\lambda \in \Delta_{\Pi}} \sum_{\pi \in \Pi} \lambda_{\pi} \left( -\Delta(\pi, \pi_{s}) + \frac{\log(1/\delta)}{\gamma_{\pi} n} \right) + \mathbb{E}_{c \sim \nu} \left[ \left( \sum_{a \in A} (\lambda \otimes \gamma)^{T} t_{a}^{(c)} \right)^2 \right] \\
\text{concave in } \lambda \text{ and locally strongly convex in } \gamma!
\]
Frank-Wolfe

- Gives us a sparse yet good enough solution $\lambda$
- Plug in solution $\lambda$ in the closed-form gives us $p_c \in \Delta_A$
Thanks!
Towards an efficient algorithm

- **argmax** oracle: given \((c_1, s_1), \ldots, (c_n, s_n) \in \mathcal{C} \times \mathbb{R}^{|\mathcal{A}|}\), returns \[
\arg \max_{\pi \in \Pi} \sum_{t=1}^{n} s_t(\pi(c_t))
\]

- Can be computed using cost-sensitive classification
A Lower Bound

- Choose action distribution $p$ such that:

$$\max_{\pi \in \Pi \setminus \pi_*} \frac{\| \phi_{\pi_*} - \phi_{\pi} \|_{A(p)^{-1}}^2}{(\Delta(\pi) \lor \epsilon)^2} \leq \frac{n}{2 \log(1/\delta)}$$
Agnostic Setting Reduces to Linear

- What if we do not assume linear structure of reward function?

We can reduce it to the previous setting by constructing \( \phi \)!

- Let \( \theta^* \in \mathbb{R}^{|C| \times |A|} \) where \( [\theta^*]_{c,a} = r(c,a) \)

\[
\begin{pmatrix}
  \vdots \\
  (c, a) \\
  \vdots \\
\end{pmatrix}
\xrightarrow{\text{vectorize}}
\theta^*
\]
Agnostic Setting Reduces to Linear

\[ r(c, a) = \langle \text{vec}(e_c e_a^T), \theta^* \rangle \]

\[ \phi(c, a) \]

\[ \|\phi_{\pi^*} - \phi_{\pi}\|_{A(p)^{-1}}^2 = \sum_c \nu_c \sum_a \frac{1}{p_{c,a}} (1\{\pi(c) = a\} - 1\{\pi^*(c) = a\})^2 = \mathbb{E}_{c \sim \nu} \left[ \left( \frac{1}{p_{c,\pi(c)}} + \frac{1}{p_{c,\pi^*(c)}} \right) 1\{\pi^*(c) \neq \pi(c)\} \right]. \]
Agnostic Setting Reduces to Linear

\[ r(c, a) = \langle \text{vec}(e_c e_a^T), \theta^* \rangle \]

\[ \phi(c, a) \]

\[ \|\phi_{\pi^*} - \phi_\pi\|_{A(p)}^2 = \sum_c \nu_c \sum_a \frac{1}{p_{c,a}} \left( \mathbf{1}\{ \pi(c) = a \} - \mathbf{1}\{ \pi^*(c) = a \} \right)^2 = \mathbb{E}_{c \sim \mathcal{U}} \left[ \left( \frac{1}{p_{c,\pi(c)}} + \frac{1}{p_{c,\pi^*(c)}} \right) \mathbf{1}\{ \pi^*(c) \neq \pi(c) \} \right]. \]

\[ \rho_{\Pi,\epsilon} := \min_{p_c \in \Delta_A, \forall c \in C} \max_{\pi \in \Pi \setminus \pi^*} \frac{\mathbb{E}_{c \sim \mathcal{U}} \left[ \left( \frac{1}{p_{c,\pi(c)}} + \frac{1}{p_{c,\pi^*(c)}} \right) \mathbf{1}\{ \pi^*(c) \neq \pi(c) \} \right]}{(\mathbb{E}_{c \sim \mathcal{U}}[ r(c, \pi^*(c)) - r(c, \pi(c)) ] \lor \epsilon)^2}. \]