Instance-Optimal PAC Algorithms for Contextual Bandits

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

*University of Washington

†Amazon

Contextual Bandit Setting

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

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

(ϵ, δ) – PAC Guarantee

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

Contributions:

- Show the first **instance-dependent** lower bound for PAC contextual bandits
- Design sampling procedure that achieves this lower bound
- Design a computationally efficient algorithm - allowing context space C and policy space Π to be **infinite**!

Regret Minimization Not Enough

- Regret heavily studied:
  - ILOVETOCONBANDITS [Agarwal et al. 2014] achieves $R_T = O(\sqrt{|A| T \log(\Pi)})$, computationally efficient
  - Modification gives (ϵ, δ)- PAC algorithm w/ sample complexity $O(|A| \log(\Pi/\delta)/\epsilon^2)$, also see [Zanette et al. 2021]

True for any policy class! Not capturing difficulty for learning π_*

<u>Two Problems</u>

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!

Agnostic Setting Reduces to Linear

- Lower bound motivated by best-arm identification in linear bandits [Fiez et al. 2019]

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

$$\begin{array}{cc} & a \\ c & \begin{pmatrix} & \vdots & \\ \cdots & r(c,a) & \\ & & \end{pmatrix} \end{array} \xrightarrow{\textbf{vectorize}} \theta^*$$

$$\Rightarrow r(c,a) = \left\langle \underbrace{\mathbf{vec}(e_c e_a^\top)}_{\phi(c,a)}, \theta^* \right\rangle$$

Contribution 1: A Lower Bound

- Let $\phi_\pi := \mathbb{E}_{c\sim\nu}[\phi(c,\pi(c))]$ and $A(p) = \sum_c \nu_c \sum_a p_{c,a}\phi(c,a)\phi(c,a)^\top$,

Theorem [Li et al. 2022] Let τ 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

variance

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

gap

- This bound is better than the sample complexity bound based on disagreement coefficients [Foster et al. 2020] and decision-estimation coefficients [Foster et al. 2021]

Contribution 2: An Instance-Optimal Algorithm

- In each round, Choose $p_c \in \triangle_{\mathsf{A}}, \forall c \in \mathsf{C}$ and n such that

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

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

Contribution 3: An Efficient Algorithm

- Consider the dual formulation of the design of the previous algorithm:

Primal $$\min_{p_c \in \triangle_{\mathsf{A}}, \forall c \in \mathsf{C}} \max_{\pi \in \Pi} -\Delta(\pi, \pi_*) + \sqrt{\frac{\|\phi_\pi - \phi_{\pi_*}\|^2_{A(p)^{-1}} \log(1/\delta)}{n}}$$

Dual $$\max_{\lambda \in \triangle_\Pi} \min_{\gamma_\pi \geq 0} \boxed{\min_{p_c \in \triangle_{\mathsf{A}}, \forall c \in \mathsf{C}} \sum_{\pi \in \Pi} \lambda_\pi \left(-\Delta(\pi, \pi_*) + \gamma_\pi \|\phi_\pi - \phi_{\pi_*}\|^2_{A(p)^{-1}} + \frac{\log(1/\delta)}{2\gamma_\pi n} \right).}$$

analytical solution ⇒ implicitly maintain p_c for all $c \in \mathsf{C}$ simultaneously!

- The dual objective is concave in λ and locally strongly convex in γ, so the saddle point problem can be solved

- Frank-Wolfe subroutine gives us a sparse yet good enough solution λ

Thank you!