Improved Regret for Differentially Private Exploration in Linear MDP

Dung Daniel Ngo, **Giuseppe Vietri**, Zhiwei Steven Wu

ICML 2022

Carnegie Mellon University

Reinforcement Learning
Background

- **Setup**: An agent interacts with an environment over K episodes.
  - Markov Decision Process (MDP): S - state set, A- action set, Reward function, Transition dynamics.
  - Episodic RL (H rounds per episodes).
  - Function approximation in Linear MDP ([Jin, Yang, Wang, Jordan. '19]): There exist a known feature mapping $\phi(x, a) \in \mathbb{R}^d$. Rewards and dynamics are linear in $\phi(\cdot, \cdot)$.

Reinforcement Learning

Background

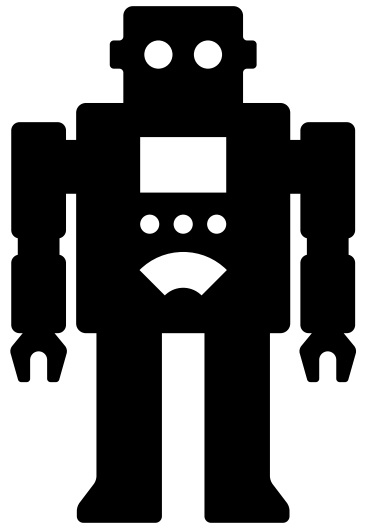
- **Setup**: An agent interacts with an environment over K episodes.
  - Markov Decision Process (MDP): S - state set, A- action set, Reward function, Transition dynamics.
  - Episodic RL (H rounds per episodes).
  - Function approximation in Linear MDP ([Jin, Yang, Wang, Jordan. '19]): There exist a known feature mapping $\phi(x, a) \in \mathbb{R}^d$. Rewards and dynamics are linear in $\phi(\cdot, \cdot)$.

At the beginning of each episode, the agent chooses policy $\pi : S \to A$

Reinforcement Learning
Background

- **Setup**: An agent interacts with an environment over K episodes.
  - Markov Decision Process (MDP): S - state set, A- action set, Reward function, Transition dynamics.
  - Episodic RL (H rounds per episodes).
  - Function approximation in Linear MDP ([Jin, Yang, Wang, Jordan. '19]): There exist a known feature mapping $\phi(x, a) \in \mathbb{R}^d$. Rewards and dynamics are linear in $\phi(\cdot, \cdot)$.

At the beginning of each episode, the agent chooses policy $\pi : S \to A$



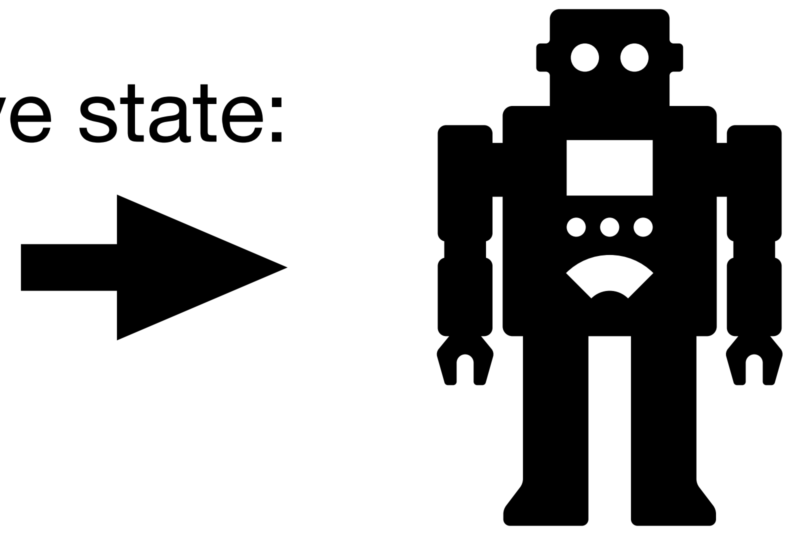
Reinforcement Learning
Background

- **Setup**: An agent interacts with an environment over K episodes.
  - Markov Decision Process (MDP): S - state set, A- action set, Reward function, Transition dynamics.
  - Episodic RL (H rounds per episodes).
  - Function approximation in Linear MDP ([Jin, Yang, Wang, Jordan. '19]): There exist a known feature mapping $\phi(x, a) \in \mathbb{R}^d$. Rewards and dynamics are linear in $\phi(\cdot, \cdot)$.

At the beginning of each episode, the agent chooses policy $\pi : S \to A$

Observe state:

x_h



Choose action:

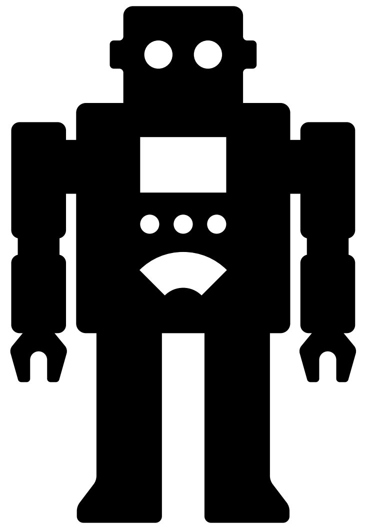
$a_h \leftarrow \pi_h(x_h)$

Reinforcement Learning
Background

- **Setup**: An agent interacts with an environment over K episodes.
  - Markov Decision Process (MDP): S - state set, A- action set, Reward function, Transition dynamics.
  - Episodic RL (H rounds per episodes).
  - Function approximation in Linear MDP ([Jin, Yang, Wang, Jordan. '19]): There exist a known feature mapping $\phi(x, a) \in \mathbb{R}^d$. Rewards and dynamics are linear in $\phi(\cdot, \cdot)$.

At the beginning of each episode, the agent chooses policy $\pi : S \to A$

Observe state: x_h



Choose action: $a_h \leftarrow \pi_h(x_h)$

Observe reward: $r_h \leftarrow \langle \theta_h, \phi(x_h, a_h) \rangle$

Unknown measure.

x_{h+1} Is sampled from $\mathbb{P}(\cdot \mid x_h, a_h) = \langle \phi(x_h, a_h), \mu(\cdot) \rangle$

Reinforcement Learning

Background

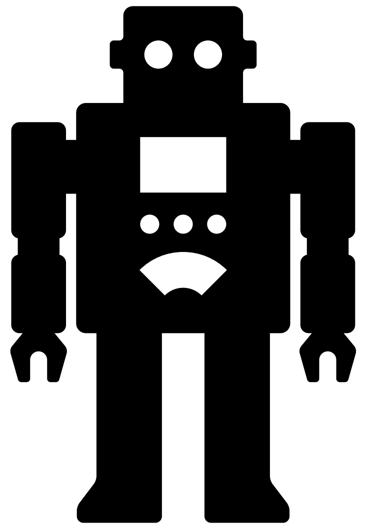
Realizability [Jin, Yang, Wang, Jordan. '19]:
There exist $w^\star$ s.t. the optimal policy has the form:

$$\pi_h^\star(x) = \arg\max_{a \in A} \langle w^\star, \phi(x,a) \rangle$$

- **Setup**: An agent interacts with an environment over K episodes.
    - Markov Decision Process (MDP): S - state set, A- action set, Reward function, Transition dynamics.
    - Episodic RL (H rounds per episodes).
    - Function approximation in Linear MDP ([Jin, Yang, Wang, Jordan. '19]): There exist a known feature mapping $\phi(x,a) \in \mathbb{R}^d$. Rewards and dynamics are linear in $\phi(\cdot\,,\cdot)$.

At the beginning of each episode, the agent chooses policy $\pi : S \to A$

Observe state: x_h



Choose action: $a_h \leftarrow \pi_h(x_h)$

Observe reward: $r_h \leftarrow \langle \theta_h, \phi(x_h, a_h) \rangle$

Unknown measure.

x_{h+1} Is sampled from $\mathbb{P}(\cdot \mid x_h, a_h) = \langle \phi(x_h, a_h), \mu(\cdot) \rangle$

Reinforcement Learning With Privacy

Motivation

- Private data: Sequence of state and rewards.
- Example:

$x_0 =$ (Fever=high, Cough=Yes, Covid=Positive) $\quad r_0 \leftarrow R(x_0, a_0)$

$x_1 =$ (Fever=high, Cough=No, Covid=Positive) $\quad r_1 \leftarrow R(x_1, a_1)$

$x_2 =$ (Fever=no, Cough=No, Covid=Positive) $\quad r_2 \leftarrow R(x_2, a_2)$

$x_3 =$ (Fever=no, Cough=No, Covid=Negative) $\quad r_3 \leftarrow R(x_3, a_3)$

Reinforcement Learning With Privacy

Motivation

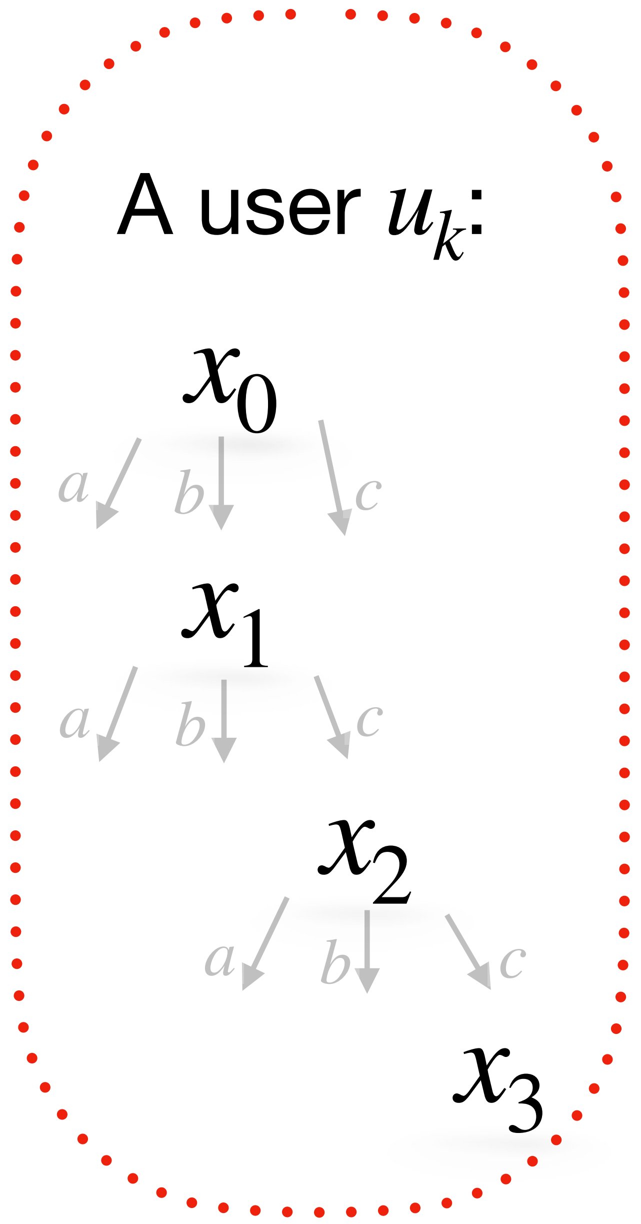
- Private data: Sequence of state and rewards.
- Example:

Private Data of user u_k

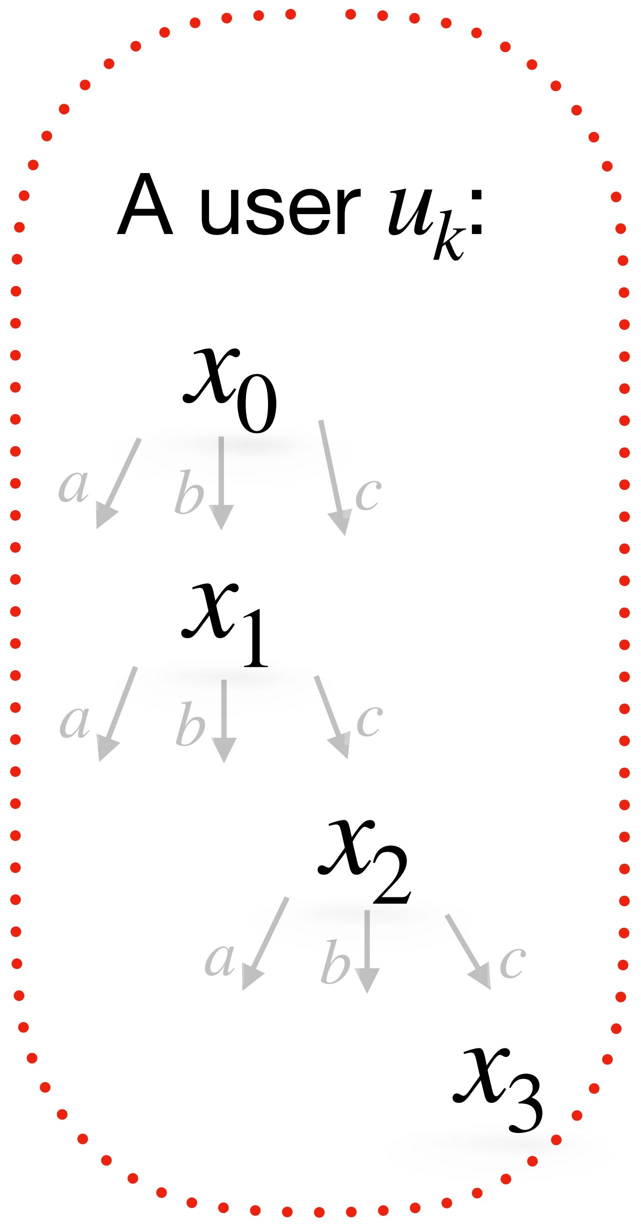
x_0 = (Fever=high, Cough=Yes, Covid=Positive)	$r_0 \leftarrow R(x_0, a_0)$
x_1 = (Fever=high, Cough=No, Covid=Positive)	$r_1 \leftarrow R(x_1, a_1)$
x_2 = (Fever=no, Cough=No, Covid=Positive)	$r_2 \leftarrow R(x_2, a_2)$
x_3 = (Fever=no, Cough=No, Covid=Negative)	$r_3 \leftarrow R(x_3, a_3)$

Joint Differential Privacy (JDP)

- Notation:
  - A user u_k Is represented by a tree. Each path encodes a sequence of states.
  - A randomized algorithm $\mathscr{M}$ takes as input a user sequence $U = (u_1, \ldots, u_K)$.
  - Outputs $a_1, \ldots a_K \leftarrow \mathscr{M}(U)$
  - $a_1, \ldots a_{k-1}, a_{k+1}, \ldots, a_K \leftarrow \mathscr{M}_{-k}(U)$ (Exclude action k)



Joint Differential Privacy (JDP)

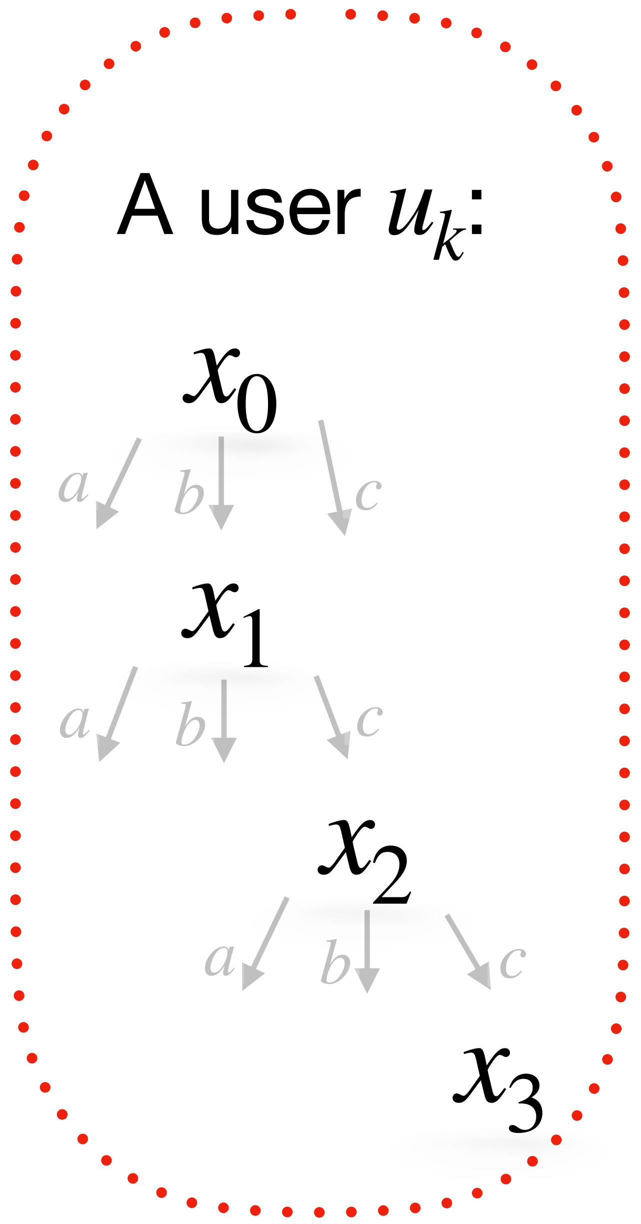


- Notation:
  - A user u_k Is represented by a tree. Each path encodes a sequence of states.
  - A randomized algorithm $\mathscr{M}$ takes as input a user sequence $U = (u_1, \ldots, u_K)$.
  - Outputs $a_1, \ldots a_K \leftarrow \mathscr{M}(U)$
  - $a_1, \ldots a_{k-1}, a_{k+1}, \ldots, a_K \leftarrow \mathscr{M}_{-k}(U)$ (Exclude action k)

A randomized algorithm $\mathscr{M}$ is **JDP** if:

- For all k and all k-neighboring $U = (u_1, \ldots, u_K)$, $\widehat{U} = (\hat{u}_1, \ldots, \hat{u}_K)$, s.t. $u_i = \hat{u}_i$ only if $i \neq k$.
- Then $\mathscr{M}_{-k}(U) \sim \mathscr{M}_{-k}(\widehat{U})$

Joint Differential Privacy (JDP)



- Notation:
  - A user u_k Is represented by a tree. Each path encodes a sequence of states.
  - A randomized algorithm $\mathscr{M}$ takes as input a user sequence $U = (u_1, \ldots, u_K)$.
  - Outputs $a_1, \ldots a_K \leftarrow \mathscr{M}(U)$
  - $a_1, \ldots a_{k-1}, a_{k+1}, \ldots, a_K \leftarrow \mathscr{M}_{-k}(U)$ (Exclude action k)

A randomized algorithm $\mathscr{M}$ is **JDP** if:

- For all k and all k-neighboring $U = (u_1, \ldots, u_K)$, $\widehat{U} = (\hat{u}_1, \ldots, \hat{u}_K)$, s.t. $u_i = \hat{u}_i$ only if $i \neq k$.
- Then $\mathscr{M}_{-k}(U) \sim \mathscr{M}_{-k}(\widehat{U})$

Intuition: Changing the data of a user in position $k \in [K]$, has a small effect on the outcome of past or future episodes.

Metrics

- Suppose there exists an optimal policy $\pi^\star$ and the algorithm plays policies $\{\pi_1, \ldots, \pi_K\}$.
- **Regret:**.

$$R(K) = \left(\text{Reward for always playing } \pi^\star\right) - \left(\text{Reward for playing } \pi_1, \ldots, \pi_K\right)$$

- **Switching Cost:** Number of times the algorithm updates the policy (Controls trade-off between non-private and private regret).

Contributions

- Algorithm: **JDP** version of **O**ptimistic **L**east-**S**quares-**V**alue-**I**teration ([Jin, Yang, Wang, Jordan. '19]) and [Wang, Zhou, Gu. '21]:
  - Private OLS: $\widetilde{w_h^k} \leftarrow (\widetilde{\Lambda_h^k})^{-1} \cdot \widetilde{y_h^k}$
  - Optimism bonus: $\widetilde{\mathrm{B}_h^k}(x,a) = \widetilde{\beta}\sqrt{\phi(x,a)(\widetilde{\Lambda_h^k})^{-1}\phi(x,a)}$

Private Statistics for OLS

$\widetilde{\Lambda_h^k} = \Lambda_h^k + \widetilde{\lambda} I + \text{noise}_2$

$\widetilde{y_h^k} = y_h^k + \text{noise}_2$

Contributions

- Algorithm: **JDP** version of **O**ptimistic **L**east-**S**quares-**V**alue-**I**teration ([Jin, Yang, Wang, Jordan. '19]) and [Wang, Zhou, Gu. '21]:
  - Private OLS: $\widetilde{w_h^k} \leftarrow (\widetilde{\Lambda_h^k})^{-1} \cdot \widetilde{y_h^k}$
  - Optimism bonus: $\widetilde{\mathrm{B}_h^k}(x,a) = \widetilde{\beta}\sqrt{\phi(x,a)(\widetilde{\Lambda_h^k})^{-1}\phi(x,a)}$

Private Statistics for OLS

$\widetilde{\Lambda_h^k} = \Lambda_h^k + \widetilde{\lambda} I + \text{noise}_2$

$\widetilde{y_h^k} = y_h^k + \text{noise}_2$

(ϵ, δ)-**J**oint **D**ifferentially **P**rivate/**L**ow **S**witching **C**ost and achieves regret of

$$\widetilde{O}\left(\sqrt{d^3H^4K} + \underbrace{H^3\sqrt{\frac{d^{5/2}K}{\epsilon}}}_{\text{Privacy cost}}\right)$$

Contributions

- Algorithm: **JDP** version of **O**ptimistic **L**east-**S**quares-**V**alue-**I**teration ([Jin, Yang, Wang, Jordan. '19]) and [Wang, Zhou, Gu. '21]:
  - Private OLS: $\widetilde{w_h^k} \leftarrow (\widetilde{\Lambda_h^k})^{-1} \cdot \widetilde{y_h^k}$
  - Optimism bonus: $\widetilde{\mathrm{B}_h^k}(x, a) = \widetilde{\beta}\sqrt{\phi(x,a)(\widetilde{\Lambda_h^k})^{-1}\phi(x,a)}$

Private Statistics for OLS

$\widetilde{\Lambda_h^k} = \Lambda_h^k + \widetilde{\lambda} I + \text{noise}_2$

$\widetilde{y_h^k} = y_h^k + \text{noise}_2$

(ϵ, δ)-**J**oint **D**ifferentially **P**rivate/**L**ow **S**witching **C**ost and achieves regret of

$$\widetilde{O}\left(\sqrt{d^3H^4K} + \underbrace{H^3\sqrt{\frac{d^{5/2}K}{\epsilon}}}_{\text{Privacy cost}}\right)$$

Matches non-private

Results [Jin et al. '19]

Contributions

- Algorithm: **JDP** version of **O**ptimistic **L**east-**S**quares-**V**alue-**I**teration ([Jin, Yang, Wang, Jordan. '19]) and [Wang, Zhou, Gu. '21]:
  - Private OLS: $\widetilde{w_h^k} \leftarrow (\widetilde{\Lambda_h^k})^{-1} \cdot \widetilde{y_h^k}$
  - Optimism bonus: $\mathrm{B}_h^k(x,a) = \widetilde{\beta}\sqrt{\phi(x,a)(\widetilde{\Lambda_h^k})^{-1}\phi(x,a)}$

Private Statistics for OLS

$\widetilde{\Lambda_h^k} = \Lambda_h^k + \widetilde{\lambda} I + \text{noise}_2$

$\widetilde{y_h^k} = y_h^k + \text{noise}_2$

(ϵ, δ)-**J**oint **D**ifferentially **P**rivate/**L**ow **S**witching **C**ost and achieves regret of

$$\widetilde{O}\left(\sqrt{d^3H^4K} + \underbrace{H^3\sqrt{\frac{d^{5/2}K}{\epsilon}}}_{\text{Privacy cost}}\right)$$

Matches non-private Results [Jin et al. '19]

Improvement over:

$$\widetilde{O}\left(\sqrt{d^3H^4K} + \frac{d^{8/5}H^{11/5}K^{3/5}}{\epsilon^{2/5}}\right)$$

- **Challenge**: Privately track statistics with low privacy cost.

- **Challenge**: Privately track statistics with low privacy cost.
- **Strategy**: Minimize policy updates (I.e., low switching cost).

- **Challenge**: Privately track statistics with low privacy cost.
- **Strategy**: Minimize policy updates (I.e., low switching cost).
- **Prior Work:** ([Luyo, Garcelon, Lazaric, and Pirotta. '21]) proposed an algorithm with switching cost $\widetilde{O}(K^{2/5})$ and regret $\widetilde{O}(K^{3/5})$.

- **Challenge**: Privately track statistics with low privacy cost.
- **Strategy**: Minimize policy updates (I.e., low switching cost).
- **Prior Work:** ([Luyo, Garcelon, Lazaric, and Pirotta. ’21]) proposed an algorithm with switching cost $\widetilde{O}(K^{2/5})$ and regret $\widetilde{O}(K^{3/5})$.
  - Updates policy using a fix schedule (Ex. every B episode.)

- **Challenge**: Privately track statistics with low privacy cost.
- **Strategy**: Minimize policy updates (I.e., low switching cost).
- **Prior Work:** ([Luyo, Garcelon, Lazaric, and Pirotta. '21]) proposed an algorithm with switching cost $\widetilde{O}(K^{2/5})$ and regret $\widetilde{O}(K^{3/5})$.
  - Updates policy using a fix schedule (Ex. every B episode.)
- **Ours**: Do adaptive policy updates (based on data) [Wang, Zhou, Gu. '21].

- **Challenge**: Privately track statistics with low privacy cost.
- **Strategy**: Minimize policy updates (I.e., low switching cost).
- **Prior Work:** ([Luyo, Garcelon, Lazaric, and Pirotta. ’21]) proposed an algorithm with switching cost $\widetilde{O}(K^{2/5})$ and regret $\widetilde{O}(K^{3/5})$.
    - Updates policy using a fix schedule (Ex. every B episode.)
- **Ours**: Do adaptive policy updates (based on data) [Wang, Zhou, Gu. ’21].
    - Switching cost is $\widetilde{O}(\log(K))$ and regret is $\widetilde{O}(\sqrt{K})$

- **Challenge**: Privately track statistics with low privacy cost.
- **Strategy**: Minimize policy updates (I.e., low switching cost).
- **Prior Work:** ([Luyo, Garcelon, Lazaric, and Pirotta. '21]) proposed an algorithm with switching cost $\widetilde{O}(K^{2/5})$ and regret $\widetilde{O}(K^{3/5})$.
  - Updates policy using a fix schedule (Ex. every B episode.)
- **Ours**: Do adaptive policy updates (based on data) [Wang, Zhou, Gu. '21].
  - Switching cost is $\widetilde{O}(\log(K))$ and regret is $\widetilde{O}(\sqrt{K})$
  - Privacy analysis: Adaptive composition.

- **Challenge**: Privately track statistics with low privacy cost.
- **Strategy**: Minimize policy updates (I.e., low switching cost).
- **Prior Work:** ([Luyo, Garcelon, Lazaric, and Pirotta. '21]) proposed an algorithm with switching cost $\widetilde{O}(K^{2/5})$ and regret $\widetilde{O}(K^{3/5})$.
  - Updates policy using a fix schedule (Ex. every B episode.)
- **Ours**: Do adaptive policy updates (based on data) [Wang, Zhou, Gu. '21].
  - Switching cost is $\widetilde{O}(\log(K))$ and regret is $\widetilde{O}(\sqrt{K})$
  - Privacy analysis: Adaptive composition.

Dung Daniel Ngo
ngo00054@umn.edu
dtngo.com

Giuseppe Vietri
vietr002@umn.edu
www.giuseppevietri.com

Zhiwei Steven Wu
zstevenwu@cmu.edu
www.zstevenwu.com