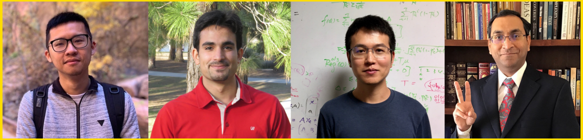
Learning Infinite-horizon Average-reward MDPs with Linear Function Approximation



Chen-Yu Wei, Mehdi Jafarnia-Jahromi, Haipeng Luo, Rahul Jain

University of Southern California

Overview

- We study online RL in **infinite-horizon average-reward** MDPs with linear function approximation.

Overview

- We study online RL in **infinite-horizon average-reward** MDPs with linear function approximation.

- Existing algorithms and regret bounds:
  - **Politex** (Abbasi-Yadkori et al.'19a): $T^{3/4}$
  - **EE-Politex** (Abbasi-Yadkori et al.'19b): $T^{4/5}$
  - **AAPI** (Hao et al.'20): $T^{2/3}$

Overview

- We study online RL in **infinite-horizon average-reward** MDPs with linear function approximation.

- Existing algorithms and regret bounds:
  - **Politex** (Abbasi-Yadkori et al.'19a): $T^{3/4}$
  - **EE-Politex** (Abbasi-Yadkori et al.'19b): $T^{4/5}$
  - **AAPI** (Hao et al.'20): $T^{2/3}$
  - Lower bound: $\sqrt{T}$

Overview

- We study online RL in **infinite-horizon average-reward** MDPs with linear function approximation.
- Existing algorithms and regret bounds:
  - **Politex** (Abbasi-Yadkori et al.'19a): $T^{3/4}$
  - **EE-Politex** (Abbasi-Yadkori et al.'19b): $T^{4/5}$
  - **AAPI** (Hao et al.'20): $T^{2/3}$
  - Lower bound: $\sqrt{T}$
- Existing algorithms make strong **uniformly mixing (UM)** and **uniformly excited feature (UEF)** assumptions

Overview

- We study online RL in **infinite-horizon average-reward** MDPs with linear function approximation.
- Existing algorithms and regret bounds:
  - **Politex** (Abbasi-Yadkori et al.'19a): $T^{3/4}$
  - **EE-Politex** (Abbasi-Yadkori et al.'19b): $T^{4/5}$
  - **AAPI** (Hao et al.'20): $T^{2/3}$
  - Lower bound: $\sqrt{T}$
- Existing algorithms make strong **uniformly mixing (UM)** and **uniformly excited feature (UEF)** assumptions
- Two contributions:
  - $\sqrt{T}$ regret bound under the same assumptions as Politex/AAPI
  - First attempt to relax the UM and UEF assumptions

Motivation

Recently there is significant progress in online RL with function approximation:
LSVI-UCB (Jin et al.'20), ELEANOR (Zanette et al.'20), $\mathcal{F}$-LSVI (Wang et al.'20)...

Motivation

Recently there is significant progress in online RL with function approximation: LSVI-UCB (Jin et al.'20), ELEANOR (Zanette et al.'20), $\mathcal{F}$-LSVI (Wang et al.'20)...

However, most of them study the **finite-horizon (episodic)** setting, and we lack comparable results for **infinite-horizon discounted** / **average-reward** settings.

Motivation

Recently there is significant progress in online RL with function approximation: LSVI-UCB (Jin et al.'20), ELEANOR (Zanette et al.'20), $\mathcal{F}$-LSVI (Wang et al.'20)...

However, most of them study the **finite-horizon (episodic)** setting, and we lack comparable results for **infinite-horizon discounted** / **average-reward** settings.

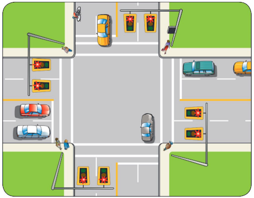
Infinite-horizon formulation is particularly relevant when the task is *continual* or *non-stopping*:

Motivation

Recently there is significant progress in online RL with function approximation: LSVI-UCB (Jin et al.'20), ELEANOR (Zanette et al.'20), $\mathcal{F}$-LSVI (Wang et al.'20)...

However, most of them study the **finite-horizon (episodic)** setting, and we lack comparable results for **infinite-horizon discounted** / **average-reward** settings.

Infinite-horizon formulation is particularly relevant when the task is *continual* or *non-stopping*:



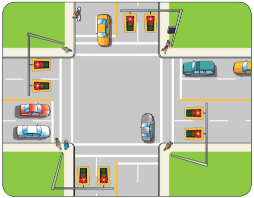
traffic control

Motivation

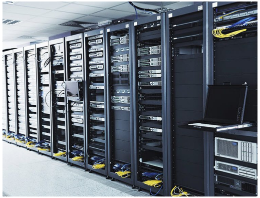
Recently there is significant progress in online RL with function approximation: LSVI-UCB (Jin et al.'20), ELEANOR (Zanette et al.'20), $\mathcal{F}$-LSVI (Wang et al.'20)...

However, most of them study the **finite-horizon (episodic)** setting, and we lack comparable results for **infinite-horizon discounted** / **average-reward** settings.

Infinite-horizon formulation is particularly relevant when the task is *continual* or *non-stopping*:



traffic control



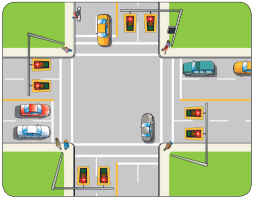
datacenter optimization

Motivation

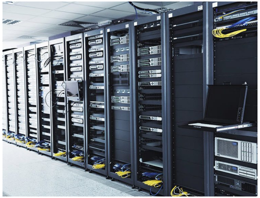
Recently there is significant progress in online RL with function approximation: LSVI-UCB (Jin et al.'20), ELEANOR (Zanette et al.'20), $\mathcal{F}$-LSVI (Wang et al.'20)...

However, most of them study the **finite-horizon (episodic)** setting, and we lack comparable results for **infinite-horizon discounted** / **average-reward** settings.

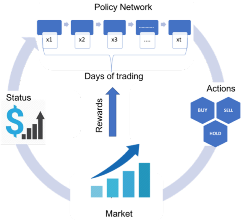
Infinite-horizon formulation is particularly relevant when the task is *continual* or *non-stopping*:



traffic control



datacenter optimization



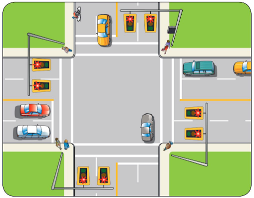
trading

Motivation

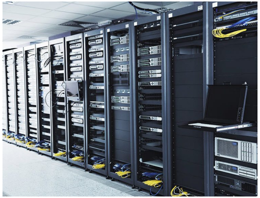
Recently there is significant progress in online RL with function approximation: LSVI-UCB (Jin et al.'20), ELEANOR (Zanette et al.'20), $\mathcal{F}$-LSVI (Wang et al.'20)...

However, most of them study the **finite-horizon (episodic)** setting, and we lack comparable results for **infinite-horizon discounted** / **average-reward** settings.

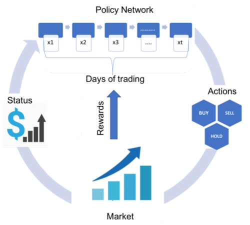
Infinite-horizon formulation is particularly relevant when the task is *continual* or *non-stopping*:



traffic control



datacenter optimization

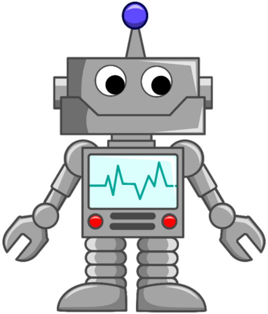


trading

self-driving

Problem Setting

Markov Decision Processes

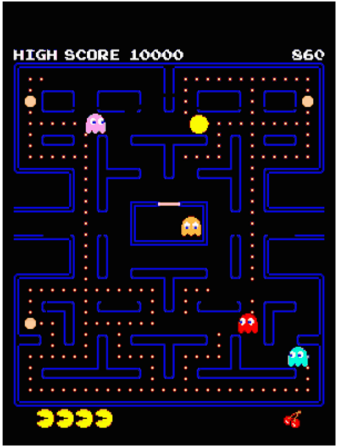


state $s_t \in \mathcal{S}$

action $a_t \in \mathcal{A}$

reward $r(s_t, a_t) \in [-1,1]$

next state $s_{t+1} \sim p(\cdot \mid s_t, a_t)$



We assume that $\mathcal{A}$ is finite, but $\mathcal{S}$ can be infinite.

Average-reward Setting and Regret

$$J^{\pi}(s) \triangleq \liminf_{n\to\infty} \frac{1}{n} \mathbb{E}\left[\sum_{t=1}^{n} r(s_t, a_t) \,\middle|\, a_t \sim \pi(\cdot \mid s_t, s_\tau, a_\tau, \tau < t), \quad s_1 = s\right]$$

$$J^{*}(s) \triangleq \sup_{\pi} J^{\pi}(s)$$

Average-reward Setting and Regret

$$J^{\pi}(s) \triangleq \liminf_{n\to\infty} \frac{1}{n} \mathbb{E}\left[\sum_{t=1}^{n} r(s_t, a_t) \;\middle|\; a_t \sim \pi(\cdot \mid s_t, s_\tau, a_\tau, \tau < t), \quad s_1 = s\right]$$

$$J^{*}(s) \triangleq \sup_{\pi} J^{\pi}(s)$$

Goal: behave as good as $J^*(s_1)$ without knowing p, r

Average-reward Setting and Regret

$$J^{\pi}(s) \triangleq \liminf_{n\to\infty} \frac{1}{n} \mathbb{E}\left[\sum_{t=1}^{n} r(s_t, a_t) \,\middle|\, a_t \sim \pi(\cdot \mid s_t, s_\tau, a_\tau, \tau < t), \quad s_1 = s\right]$$

$$J^{*}(s) \triangleq \sup_{\pi} J^{\pi}(s)$$

Goal: behave as good as $J^*(s_1)$ without knowing p, r

Two facts that make the learning problem (too) difficult for online RL:

- The optimal policy can be history-dependent (when $|\mathcal{S}| = \infty$)
- $J^*(s)$ depends on s

Assumption

The **Bellman Optimality Equation** holds:

$$q^*(s,a) = r(s,a) - J^* + \mathbb{E}_{s' \sim p(\cdot|s,a)}\left[v^*(s')\right], \qquad v^*(s) = \max_a q^*(s,a)$$

with some uniformly bounded $v^*(\cdot)$, $q^*(\cdot,\cdot)$, and J^*.
q^* and v^* are called *(optimal) bias functions*.

Assumption

The **Bellman Optimality Equation** holds:

$$q^*(s,a) = r(s,a) - J^* + \mathbb{E}_{s' \sim p(\cdot|s,a)}\left[v^*(s')\right], \qquad v^*(s) = \max_a q^*(s,a)$$

with some uniformly bounded $v^*(\cdot)$, $q^*(\cdot,\cdot)$, and J^*.
q^* and v^* are called *(optimal) bias functions*.

The assumption implies $J^*(s) = J^*$ and a stationary optimal policy $\pi^* : \mathcal{S} \to \mathcal{A}$.
E.g., weakly communicating MDP **(tabular case)**

Assumption

The **Bellman Optimality Equation** holds:

$$q^*(s,a) = r(s,a) - J^* + \mathbb{E}_{s' \sim p(\cdot|s,a)}\left[v^*(s')\right], \qquad v^*(s) = \max_a q^*(s,a)$$

with some uniformly bounded $v^*(\cdot)$, $q^*(\cdot,\cdot)$, and J^*.
q^* and v^* are called *(optimal) bias functions*.

The assumption implies $J^*(s) = J^*$ and a stationary optimal policy $\pi^* : \mathcal{S} \to \mathcal{A}$.
E.g., weakly communicating MDP **(tabular case)**

$$\boxed{\mathrm{Reg}_T \triangleq TJ^* - \sum_{t=1}^{T} r(s_t, a_t)}$$

Assumption

The **Bellman Optimality Equation** holds:

$$q^*(s,a) = r(s,a) - J^* + \mathbb{E}_{s'\sim p(\cdot|s,a)}\left[v^*(s')\right], \qquad v^*(s) = \max_a q^*(s,a)$$

with some uniformly bounded $v^*(\cdot)$, $q^*(\cdot,\cdot)$, and J^*.
q^* and v^* are called *(optimal) bias functions*.

The assumption implies $J^*(s) = J^*$ and a stationary optimal policy $\pi^* : \mathcal{S} \to \mathcal{A}$.
E.g., weakly communicating MDP **(tabular case)**

$$\boxed{\text{Reg}_T \triangleq TJ^* - \sum_{t=1}^{T} r(s_t, a_t)}$$

In the **tabular case**: $\Theta\left(\sqrt{\text{sp}(v^*)SAT}\right)$, where $\text{sp}(v^*) \triangleq \sup_{s,s'} |v^*(s) - v^*(s')|$
(Zhang&Ji'19, Jaksch et al.'10)

Linear Function Approximation

When the state or action space is very large, learning them independently could be inefficient.

Linear Function Approximation Schemes:

Linear Function Approximation

When the state or action space is very large, learning them independently could be inefficient.

Linear Function Approximation Schemes: given $\Phi(s, a) \in \mathbb{R}^d \;\; \forall s, a$

Examples

Linear Function Approximation

When the state or action space is very large, learning them independently could be inefficient.

Linear Function Approximation Schemes: given $\Phi(s, a) \in \mathbb{R}^d \ \forall s, a$

Examples

1. $q^*(s, a) = \Phi(s, a)^\top w^*$ **(Linear q^*)**

Linear Function Approximation

When the state or action space is very large, learning them independently could be inefficient.

Linear Function Approximation Schemes: given $\Phi(s, a) \in \mathbb{R}^d \ \forall s, a$

Examples

1. $q^*(s, a) = \Phi(s, a)^\top w^*$ **(Linear q^*)**
2. $q^\pi(s, a) = \Phi(s, a)^\top w^\pi \ \forall \pi$ **(Linear $q^\pi \ \forall \pi$)**

Linear Function Approximation

When the state or action space is very large, learning them independently could be inefficient.

Linear Function Approximation Schemes: given $\Phi(s, a) \in \mathbb{R}^d \ \ \forall s, a$

Examples

1. $q^*(s, a) = \Phi(s, a)^\top w^*$ **(Linear q^*)**
2. $q^\pi(s, a) = \Phi(s, a)^\top w^\pi \ \forall \pi$ **(Linear $q^\pi \ \forall \pi$)**
3. $p(s'|s, a) = \Phi(s, a)^\top \Psi(s'), \qquad r(s, a) = \Phi(s, a)^\top \Theta$ **(Linear MDP)**

Linear Function Approximation

When the state or action space is very large, learning them independently could be inefficient.

Linear Function Approximation Schemes: given $\Phi(s,a) \in \mathbb{R}^d \;\; \forall s, a$

Examples

1. $q^*(s,a) = \Phi(s,a)^\top w^*$ **(Linear q^*)**
2. $q^\pi(s,a) = \Phi(s,a)^\top w^\pi \; \forall \pi$ **(Linear $q^\pi \; \forall \pi$)**
3. $p(s'|s,a) = \Phi(s,a)^\top \Psi(s'), \qquad r(s,a) = \Phi(s,a)^\top \Theta$ **(Linear MDP)**

(Linear MDP) $\subset$ **(Linear $q^\pi \; \forall \pi$)** $\subset$ **(Linear q^*)**

Other Assumptions Made in Previous Works

- **(Uniformly Mixing)** [Politex, EE-Politex, AAPI] For any policy π, any state distributions ν,

$$\|\mathbb{P}^{\pi}\nu - \mu^{\pi}\|_{\text{TV}} \le e^{-1/t_{\text{mix}}}\|\nu - \mu^{\pi}\|_{\text{TV}}$$

where μ^{π} is the unique *stationary state distirbution* under π.

Other Assumptions Made in Previous Works

- **(Uniformly Mixing)** [Politex, EE-Politex, AAPI] For any policy π, any state distributions ν,

$$\|\mathbb{P}^{\pi}\nu - \mu^{\pi}\|_{\text{TV}} \leq e^{-1/t_{\text{mix}}}\|\nu - \mu^{\pi}\|_{\text{TV}}$$

  where μ^{π} is the unique *stationary state distirbution* under π.

- **(Uniformly Excited Features)** [Politex, AAPI] For any policy π,

$$\lambda_{\min}\left(\mathbb{E}_{s\sim\mu^{\pi},a\sim\pi(\cdot|s)}\left[\Phi(s,a)\Phi(s,a)^{\top}\right]\right) \geq \sigma, \tag{1}$$

Other Assumptions Made in Previous Works

- **(Uniformly Mixing)** [Politex, EE-Politex, AAPI] For any policy π, any state distributions ν,

$$\|\mathbb{P}^{\pi}\nu - \mu^{\pi}\|_{\text{TV}} \leq e^{-1/t_{\text{mix}}}\|\nu - \mu^{\pi}\|_{\text{TV}}$$

  where μ^{π} is the unique *stationary state distirbution* under π.

- **(Uniformly Excited Features)** [Politex, AAPI] For any policy π,

$$\lambda_{\min}\left(\mathbb{E}_{s\sim\mu^{\pi},a\sim\pi(\cdot|s)}\left[\Phi(s,a)\Phi(s,a)^{\top}\right]\right) \geq \sigma, \tag{1}$$

  [EE-Politex] assumes that (1) holds for some known policy π_e

Comparison with Previous Works

Algorithm	Regret	Assumptions	
		Explorability	Structure
Politex (Abbasi-Yadkori et al.)	$O(T^{\frac{3}{4}})$	UM + UEF	Linear q^π $\forall\pi$
AAPI (Hao et al.)	$O(T^{\frac{2}{3}})$		

UM: Uniformly Mixing   UEF: Uniformly Excited Features   BOE: Bellman Optimality Eqn.

Comparison with Previous Works

<table>
<tr><th rowspan="2">Algorithm</th><th rowspan="2">Regret</th><th colspan="2">Assumptions</th></tr>
<tr><th>Explorability</th><th>Structure</th></tr>
<tr><td>Politex (Abbasi-Yadkori et al.)</td><td>$O(T^{\frac{3}{4}})$</td><td rowspan="2">UM + UEF</td><td rowspan="4">Linear q^{π} $\forall \pi$</td></tr>
<tr><td>AAPI (Hao et al.)</td><td>$O(T^{\frac{2}{3}})$</td></tr>
<tr><td>EE-Politex (Abbasi-Yadkori et al.)</td><td>$O(T^{\frac{4}{5}})$</td><td>UM + π_e</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td></td><td></td></tr>
</table>

UM: Uniformly Mixing UEF: Uniformly Excited Features BOE: Bellman Optimality Eqn.

Comparison with Previous Works

Algorithm	Regret	Assumptions	
		Explorability	Structure
Politex (Abbasi-Yadkori et al.)	$O(T^{\frac{3}{4}})$	UM + UEF	Linear q^π $\forall\pi$
AAPI (Hao et al.)	$O(T^{\frac{2}{3}})$		
EE-Politex (Abbasi-Yadkori et al.)	$O(T^{\frac{4}{5}})$	UM + π_e	
MDP-EXP2 (Part II)	$O(\sqrt{T})$	UM + UEF	

UM: Uniformly Mixing UEF: Uniformly Excited Features BOE: Bellman Optimality Eqn.

Comparison with Previous Works

Algorithm	Regret	Assumptions	
		Explorability	Structure
Politex (Abbasi-Yadkori et al.)	$O(T^{\frac{3}{4}})$	UM + UEF	Linear q^π $\forall\pi$
AAPI (Hao et al.)	$O(T^{\frac{2}{3}})$		
EE-Politex (Abbasi-Yadkori et al.)	$O(T^{\frac{4}{5}})$	UM + π_e	
MDP-EXP2 (Part II)	$O(\sqrt{T})$	UM + UEF	
FOPO (Part I)	$O(\sqrt{T})$	BOE	Linear MDP

UM: Uniformly Mixing   UEF: Uniformly Excited Features   BOE: Bellman Optimality Eqn.

Comparison with Previous Works

Algorithm	Regret	Assumptions	
		Explorability	Structure
Politex (Abbasi-Yadkori et al.)	$O(T^{\frac{3}{4}})$	UM + UEF	Linear q^π $\forall\pi$
AAPI (Hao et al.)	$O(T^{\frac{2}{3}})$		
EE-Politex (Abbasi-Yadkori et al.)	$O(T^{\frac{4}{5}})$	UM + π_e	
MDP-EXP2 (Part II)	$O(\sqrt{T})$	UM + UEF	
FOPO (Part I)	$O(\sqrt{T})$	BOE	Linear MDP
Optimistic-LSVI (Part I)	$O(T^{\frac{3}{4}})$		

UM: Uniformly Mixing UEF: Uniformly Excited Features BOE: Bellman Optimality Eqn.

Comparison with Previous Works

Algorithm	Regret	Assumptions	
		Explorability	Structure
Politex (Abbasi-Yadkori et al.)	$O(T^{\frac{3}{4}})$	UM + UEF	Linear q^π $\forall\pi$
AAPI (Hao et al.)	$O(T^{\frac{2}{3}})$		
EE-Politex (Abbasi-Yadkori et al.)	$O(T^{\frac{4}{5}})$	UM + π_e	
MDP-EXP2 (Part II)	$O(\sqrt{T})$	UM + UEF	
FOPO (Part I)	$O(\sqrt{T})$	BOE	Linear MDP
Optimistic-LSVI (Part I)	$O(T^{\frac{3}{4}})$		

UM: Uniformly Mixing   UEF: Uniformly Excited Features   BOE: Bellman Optimality Eqn.

Relations between the Assumptions:

- (UM + UEF) $\subset$ (UM + π_e) $\subset$ BOE
- (Linear MDP) $\subset$ (Linear q^π $\forall\pi$)

Comparison with Previous Works

Algorithm	Regret	Assumptions	
		Explorability	Structure
Politex (Abbasi-Yadkori et al.)	$O(T^{\frac{3}{4}})$	UM + UEF	Linear q^π $\forall\pi$
AAPI (Hao et al.)	$O(T^{\frac{2}{3}})$		
EE-Politex (Abbasi-Yadkori et al.)	$O(T^{\frac{4}{5}})$	UM + π_e	
MDP-EXP2 (Part II)	$O(\sqrt{T})$	UM + UEF	
FOPO (Part I)	$O(\sqrt{T})$	BOE	Linear MDP
Optimistic-LSVI (Part I)	$O(T^{\frac{3}{4}})$		

UM: Uniformly Mixing UEF: Uniformly Excited Features BOE: Bellman Optimality Eqn.

Contributions

- Improving Politex/AAPI's regret bound under the same setting (Part II)
- First attempt to relax the UM and UEF assumptions (Part I)

Part I: Linear MDP

Recap of the Assumptions

1. $p(s'|s,a) = \Phi(s,a)^\top \Psi(s'), \quad r(s,a) = \Phi(s,a)^\top \Theta$
2. Bellman optimality equation (BOE) holds:

$$q^*(s,a) = r(s,a) - J^* + \mathbb{E}_{s'\sim p(\cdot|s,a)}\left[v^*(s')\right], \qquad v^*(s) = \max_a q^*(s,a)$$

3. $\Phi(s,a)_1 = 1$ (W.L.O.G.)

The assumptions imply $\boxed{q^*(s,a) = \Phi(s,a)^\top w^*}$

Algorithm: Fixed-point OPtimization with Optimism (FOPO)

Algorithm: Fixed-point OPtimization with Optimism (FOPO)

FOPO is based on the “optimism under the face of uncertainty” principle:

Algorithm: Fixed-point OPtimization with Optimism (FOPO)

FOPO is based on the "optimism under the face of uncertainty" principle:

$\Lambda_t \triangleq I + \sum_{\tau=1}^{t-1} \Phi(s_\tau, a_\tau)\Phi(s_\tau, a_\tau)^\top$

Every time when $\det(\Lambda_t)$ doubles, solve

$$\max_{w_t, J, b} \; J$$

$$\text{s.t.} \quad w_t = \Lambda_t^{-1} \sum_{\tau=1}^{t-1} \Phi(s_\tau, a_\tau) \left(r(s_\tau, a_\tau) - J + \max_a \Phi(s_{\tau+1}, a)^\top w_t \right) + b$$

$$\|w_t\| \le \text{sp}(v^*)\sqrt{d}, \quad \|b\|_{\Lambda_t} \le \beta = \Theta\left(\text{sp}(v^*) d \log T\right)$$

Else: $w_t \leftarrow w_{t-1}$

Choose $a_t = \text{argmax}_a \Phi(s_t, a)^\top w_t$

Algorithm: Fixed-point OPtimization with Optimism (FOPO)

FOPO is based on the **global optimism** idea (Zanette et al'20, Neu&Pike-Burke'20).

Algorithm: Fixed-point OPtimization with Optimism (FOPO)

FOPO is based on the **global optimism** idea (Zanette et al'20, Neu&Pike-Burke'20).

Theorem

FOPO achieves $\text{Reg}_T = \widetilde{O}\left(\text{sp}(v^*)\sqrt{d^3T}\right)$.

Algorithm: Fixed-point OPtimization with Optimism (FOPO)

FOPO is based on the **global optimism** idea (Zanette et al'20, Neu&Pike-Burke'20).

Theorem

FOPO achieves $\text{Reg}_T = \widetilde{O}\left(\text{sp}(v^*)\sqrt{d^3 T}\right)$.

Issues of FOPO: not computationally tractable.

Algorithm: Fixed-point OPtimization with Optimism (FOPO)

FOPO is based on the **global optimism** idea (Zanette et al'20, Neu&Pike-Burke'20).

Theorem

FOPO achieves $\text{Reg}_T = \widetilde{O}\left(\text{sp}(v^*)\sqrt{d^3T}\right)$.

Issues of FOPO: not computationally tractable.

Remark. Achieving $O(\text{sp}(v^*)\sqrt{T})$ with a **computationally efficient** algorithm is already highly non-trivial in the tabular case: REGAL (Bartlett&Tewari'09), SCAL (Fruit et al'18), SCAL+ (Qian et al.'18)

Making FOPO Efficient

Attempt 1.

1. Avoid solving a fixed-point problem of w_t.

Making FOPO Efficient

Attempt 1.

1. Avoid solving a fixed-point problem of w_t.
2. Use local exploration bonus instead of global optimism (Neu&Pike-Burke'20).

Making FOPO Efficient

Attempt 1.

1. Avoid solving a fixed-point problem of w_t.
2. Use local exploration bonus instead of global optimism (Neu&Pike-Burke'20).
3. Reduce the average-reward problem to a discounted problem (Wei et al.'20).

Making FOPO Efficient

Attempt 1.

1. Avoid solving a fixed-point problem of w_t.
2. Use local exploration bonus instead of global optimism (Neu&Pike-Burke'20).
3. Reduce the average-reward problem to a discounted problem (Wei et al.'20).

Counterpart of LSVI-UCB (Jin et al'20) for the discounted setting:

$$w_t = \Lambda_t^{-1} \sum_{\tau=1}^{t-1} \Phi(s_\tau, a_\tau) \left(r(s_\tau, a_\tau) + \gamma V_{t-1}(s_{\tau+1}) \right),$$

$$V_{t-1}(\cdot) = \max_a \left(\Phi(\cdot, a)^\top w_{t-1} + \text{bonus}(\cdot, a) \right)$$

Making FOPO Efficient

Attempt 1.

1. Avoid solving a fixed-point problem of w_t.
2. Use local exploration bonus instead of global optimism (Neu&Pike-Burke'20).
3. Reduce the average-reward problem to a discounted problem (Wei et al.'20).

Counterpart of LSVI-UCB (Jin et al'20) for the discounted setting:

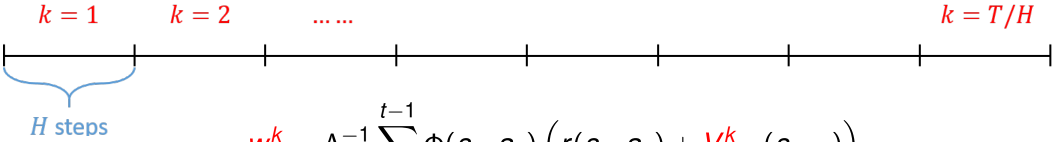
$$w_t = \Lambda_t^{-1} \sum_{\tau=1}^{t-1} \Phi(s_\tau, a_\tau)\left(r(s_\tau, a_\tau) + \gamma V_{t-1}(s_{\tau+1})\right),$$

$$V_{t-1}(\cdot) = \max_a \left(\Phi(\cdot, a)^\top w_{t-1} + \text{bonus}(\cdot, a)\right)$$

Unfortunately, we are unable to show sub-linear regret for this algorithm.

Making FOPO Efficient: Optimistic LSVI

Idea. Reduction to the episodic setting (Jin et al.'20)

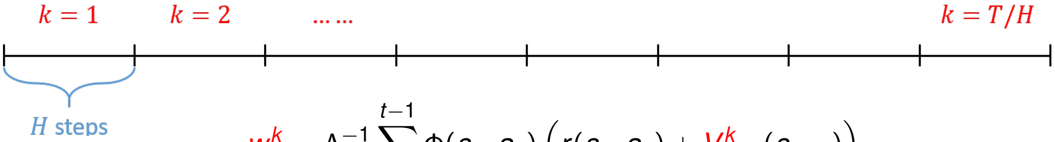


$$w_h^k = \Lambda_t^{-1} \sum_{\tau=1}^{t-1} \Phi(s_\tau, a_\tau) \left(r(s_\tau, a_\tau) + V_{h+1}^k(s_{\tau+1}) \right),$$

$$V_{h+1}^k(\cdot) = \max_a \left(\Phi(\cdot, a)^\top w_{h+1}^k + \text{bonus}(\cdot, a) \right)$$

Making FOPO Efficient: Optimistic LSVI

Idea. Reduction to the episodic setting (Jin et al.'20)



$$w_h^k = \Lambda_t^{-1} \sum_{\tau=1}^{t-1} \Phi(s_\tau, a_\tau) \left(r(s_\tau, a_\tau) + V_{h+1}^k(s_{\tau+1}) \right),$$

$$V_{h+1}^k(\cdot) = \max_a \left(\Phi(\cdot, a)^\top w_{h+1}^k + \text{bonus}(\cdot, a) \right)$$

Theorem

By reduction to the episodic setting, we get $\widetilde{O}\left(\sqrt{\text{sp}(v^*)}(dT)^{\frac{3}{4}}\right)$ regret efficiently.

Summary for Part I (Linear MDPs)

1. A computationally **intractable** algorithm with $O\left(\mathrm{sp}(v^*)\sqrt{d^3T}\right)$ regret.
2. A computationally **efficient** algorithm with $O\left(\sqrt{\mathrm{sp}(v^*)}(dT)^{\frac{3}{4}}\right)$ regret (by reducing the problem to the episodic setting)

Summary for Part I (Linear MDPs)

1. A computationally **intractable** algorithm with $O\left(\text{sp}(v^*)\sqrt{d^3 T}\right)$ regret.
2. A computationally **efficient** algorithm with $O\left(\sqrt{\text{sp}(v^*)}(dT)^{\frac{3}{4}}\right)$ regret (by reducing the problem to the episodic setting)

Open Problems

1. An $O(\sqrt{T})$ computationally tractable algorithm for linear MDPs.
2. Sample complexity bound for online RL + linear MDPs + **infinite-horizon discounted setting**.

Part II: MDP-EXP2

Recap of the Assumptions

1. **Uniformly Mixing**: for any policy π, any state distribution ν,

$$\|\mathbb{P}^{\pi}\nu - \mu^{\pi}\|_{\text{TV}} \leq e^{-1/t_{\text{mix}}}\|\nu - \mu^{\pi}\|_{\text{TV}}$$

2. **Uniformly Excited Features**: for any π

$$\lambda_{\min}\left(\mathbb{E}_{s\sim\mu^{\pi}, a\sim\pi(\cdot|s)}\left[\Phi(s,a)\Phi(s,a)^{\top}\right]\right) \geq \sigma$$

Recap of the Assumptions

1. **Uniformly Mixing**: for any policy π, any state distribution ν,

$$\|\mathbb{P}^{\pi}\nu - \mu^{\pi}\|_{\text{TV}} \leq e^{-1/t_{\text{mix}}}\|\nu - \mu^{\pi}\|_{\text{TV}}$$

2. **Uniformly Excited Features**: for any π

$$\lambda_{\min}\left(\mathbb{E}_{s\sim\mu^{\pi},a\sim\pi(\cdot|s)}\left[\Phi(s,a)\Phi(s,a)^{\top}\right]\right) \geq \sigma$$

Uniformly Mixing implies $J^{\pi}(s) = J^{\pi}$ and

$$q^{\pi}(s,a) = r(s,a) - J^{\pi} + \mathbb{E}_{s'\sim p(\cdot|s,a)}\left[v^{\pi}(s')\right]$$

Recap of the Assumptions

1. **Uniformly Mixing**: for any policy π, any state distribution ν,

$$\|\mathbb{P}^\pi \nu - \mu^\pi\|_{\text{TV}} \le e^{-1/t_{\text{mix}}} \|\nu - \mu^\pi\|_{\text{TV}}$$

2. **Uniformly Excited Features**: for any π

$$\lambda_{\min}\left(\mathbb{E}_{s\sim\mu^\pi, a\sim\pi(\cdot|s)}\left[\Phi(s,a)\Phi(s,a)^\top\right]\right) \ge \sigma$$

Uniformly Mixing implies $J^\pi(s) = J^\pi$ and

$$q^\pi(s,a) = r(s,a) - J^\pi + \mathbb{E}_{s'\sim p(\cdot|s,a)}\left[v^\pi(s')\right]$$

3. **Linear q^π**: for any π, there exists $w^\pi \in \mathbb{R}^d$,

$$q^\pi(s,a) = \Phi(s,a)^\top w^\pi$$

Recap of the Assumptions

1. **Uniformly Mixing**: for any policy π, any state distribution ν,

$$\|\mathbb{P}^{\pi}\nu - \mu^{\pi}\|_{\mathrm{TV}} \leq e^{-1/t_{\mathrm{mix}}}\|\nu - \mu^{\pi}\|_{\mathrm{TV}}$$

2. **Uniformly Excited Features**: for any π

$$\lambda_{\min}\left(\mathbb{E}_{s\sim\mu^{\pi}, a\sim\pi(\cdot|s)}\left[\Phi(s,a)\Phi(s,a)^{\top}\right]\right) \geq \sigma$$

Uniformly Mixing implies $J^{\pi}(s) = J^{\pi}$ and

$$q^{\pi}(s,a) = r(s,a) - J^{\pi} + \mathbb{E}_{s'\sim p(\cdot|s,a)}\left[v^{\pi}(s')\right]$$

3. **Linear q^{π}**: for any π, there exists $w^{\pi} \in \mathbb{R}^{d}$,

$$q^{\pi}(s,a) = \Phi(s,a)^{\top} w^{\pi}$$

4. $\Phi_1(s,a) = 1$ (W.L.O.G.)

Detour: Adversarial Linear Bandit Algorithm – EXP2

(Dani et al'08, Bubeck et al'12)

(linear bandit $\approx$ single state MDP)

Detour: Adversarial Linear Bandit Algorithm – EXP2

(Dani et al'08, Bubeck et al'12)

(linear bandit $\approx$ single state MDP)
Given action set $\mathcal{A}$, and feature mappings $\{\Phi(a)\}_{a\in\mathcal{A}} \subset \mathbb{R}^d$

$\pi_1 = \frac{1}{|\mathcal{A}|}$
For $t = 1, \ldots, T$**:**

- Sample $a_t \sim \pi_t \in \Delta_{\mathcal{A}}$, and observe reward $\Phi(a_t)^\top w_t$ (w_t can be adversarially chosen)

Detour: Adversarial Linear Bandit Algorithm – EXP2

(Dani et al'08, Bubeck et al'12)

(linear bandit $\approx$ single state MDP)
Given action set $\mathcal{A}$, and feature mappings $\{\Phi(a)\}_{a\in\mathcal{A}} \subset \mathbb{R}^d$

$\pi_1 = \frac{1}{|\mathcal{A}|}$
For $t = 1, \ldots, T$**:**

- Sample $a_t \sim \pi_t \in \Delta_{\mathcal{A}}$, and observe reward $\Phi(a_t)^\top w_t$ (w_t can be adversarially chosen)
- Construct $\widehat{w}_t$ with $\mathbb{E}[\widehat{w}_t] = w_t$ (unbiased estimator)

Detour: Adversarial Linear Bandit Algorithm – EXP2

(Dani et al'08, Bubeck et al'12)

(linear bandit $\approx$ single state MDP)
Given action set $\mathcal{A}$, and feature mappings $\{\Phi(a)\}_{a\in\mathcal{A}} \subset \mathbb{R}^d$

$\pi_1 = \frac{1}{|\mathcal{A}|}$
For $t = 1, \ldots, T$:

- Sample $a_t \sim \pi_t \in \Delta_{\mathcal{A}}$, and observe reward $\Phi(a_t)^\top w_t$ (w_t can be adversarially chosen)
- Construct $\widehat{w}_t$ with $\mathbb{E}[\widehat{w}_t] = w_t$ (unbiased estimator)
- Update action distribution with exponential weight:

$$\pi_{t+1}(a) \propto \pi_t(a) \exp\left(\eta \Phi(a)^\top \widehat{w}_t\right)$$

Detour: Adversarial Linear Bandit Algorithm – EXP2

(Dani et al'08, Bubeck et al'12)

(linear bandit $\approx$ single state MDP)
Given action set $\mathcal{A}$, and feature mappings $\{\Phi(a)\}_{a\in\mathcal{A}} \subset \mathbb{R}^d$

$\pi_1 = \frac{1}{|\mathcal{A}|}$
For $t = 1, \ldots, T$**:**

- Sample $a_t \sim \pi_t \in \Delta_{\mathcal{A}}$, and observe reward $\Phi(a_t)^\top w_t$ (w_t can be adversarially chosen)
- Construct $\widehat{w}_t$ with $\mathbb{E}[\widehat{w}_t] = w_t$ (unbiased estimator)
- Update action distribution with exponential weight:

$$\pi_{t+1}(a) \propto \pi_t(a) \exp\left(\eta \Phi(a)^\top \widehat{w}_t\right)$$

$$\textbf{Regret} \triangleq \sum_t \sum_a (\pi^*(a) - \pi_t(a)) \Phi(a)^\top w_t$$

Reduction from MDP to Adversarial LB

Based on the "performance difference lemma",

$$\textbf{Regret} = \sum_k \sum_{s,a} \mu^{\pi^*}(s) \left(\pi^*(a|s) - \pi_k(a|s)\right) q^{\pi_k}(s,a)$$

Reduction from MDP to Adversarial LB

Based on the “performance difference lemma”,

$$\textbf{Regret} = \sum_k \sum_{s,a} \mu^{\pi^*}(s) \left(\pi^*(a|s) - \pi_k(a|s)\right) q^{\pi_k}(s,a)$$

$$= \sum_s \mu^{\pi^*}(s) \underbrace{\left(\sum_k \sum_a \left(\pi^*(a|s) - \pi_k(a|s)\right) \Phi(s,a)^\top w^{\pi_k} \right)}_{\text{the regret of the linear bandit problem on state } s}$$

Reduction from MDP to Adversarial LB

Based on the "performance difference lemma",

$$\textbf{Regret} = \sum_k \sum_{s,a} \mu^{\pi^*}(s) \left(\pi^*(a|s) - \pi_k(a|s)\right) q^{\pi_k}(s,a)$$

$$= \sum_s \mu^{\pi^*}(s) \underbrace{\left(\sum_k \sum_a \left(\pi^*(a|s) - \pi_k(a|s)\right) \Phi(s,a)^\top w^{\pi_k} \right)}_{\text{the regret of the linear bandit problem on state } s}$$

Unlike in LB, the learner does not observe $\Phi(s_t, a_t)^\top w^{\pi_k}$

Reduction from MDP to Adversarial LB

Based on the "performance difference lemma",

$$\textbf{Regret} = \sum_k \sum_{s,a} \mu^{\pi^*}(s)\left(\pi^*(a|s) - \pi_k(a|s)\right) q^{\pi_k}(s,a)$$

$$= \sum_s \mu^{\pi^*}(s) \underbrace{\left(\sum_k \sum_a \left(\pi^*(a|s) - \pi_k(a|s)\right) \Phi(s,a)^\top w^{\pi_k}\right)}_{\text{the regret of the linear bandit problem on state } s}$$

Unlike in LB, the learner does not observe $\Phi(s_t, a_t)^\top w^{\pi_k}$

$$= \sum_s \mu^{\pi^*}(s) \left(\sum_k \sum_a \left(\pi^*(a|s) - \pi_k(a|s)\right) \left(\Phi(s,a)^\top w^{\pi_k} + c\right)\right)$$

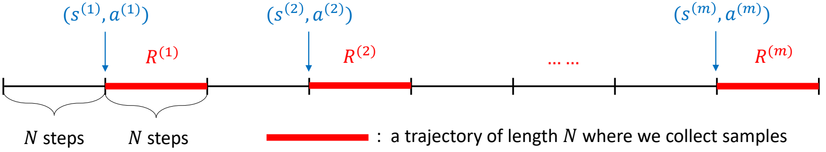
It suffices to construct a $\widehat{w}_k$ with $\mathbb{E}\left[\Phi(s,a)^\top \widehat{w}_k\right] = \Phi(s,a)^\top w^{\pi_k} + c$

Constructing (Nearly) Unbiased Estimators

Q: for a fixed π, how to construct $\widehat{w}$ with $\mathbb{E}\left[\Phi(s,a)^\top \widehat{w}\right] = \Phi(s,a)^\top w^\pi + c$?

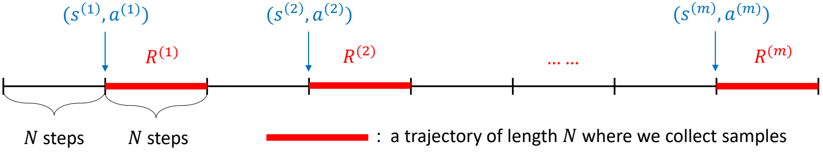
Constructing (Nearly) Unbiased Estimators

Q: for a fixed π, how to construct $\widehat{w}$ with $\mathbb{E}\left[\Phi(s,a)^\top \widehat{w}\right] = \Phi(s,a)^\top w^\pi + c$?



Constructing (Nearly) Unbiased Estimators

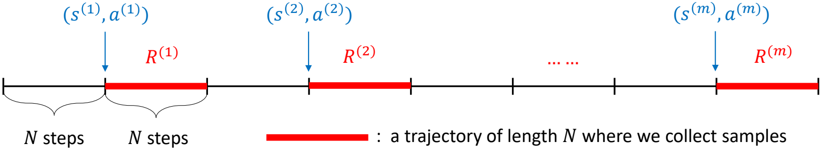
Q: for a fixed π, how to construct $\widehat{w}$ with $\mathbb{E}\left[\Phi(s,a)^\top \widehat{w}\right] = \Phi(s,a)^\top w^\pi + c$?



1. Execute π for $2mN$ steps, where $m = \Theta\left(\frac{1}{\sigma}\right)$ and $N = \Theta\left(t_{\text{mix}}\right)$

Constructing (Nearly) Unbiased Estimators

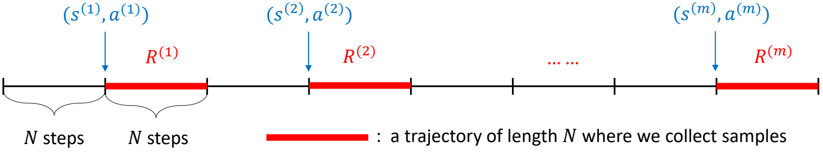
Q: for a fixed π, how to construct $\widehat{w}$ with $\mathbb{E}\left[\Phi(s,a)^\top \widehat{w}\right] = \Phi(s,a)^\top w^\pi + c$?



1. Execute π for $2mN$ steps, where $m = \Theta\left(\frac{1}{\sigma}\right)$ and $N = \Theta\left(t_{\text{mix}}\right)$
2. Collect m trajectories each of length N

Constructing (Nearly) Unbiased Estimators

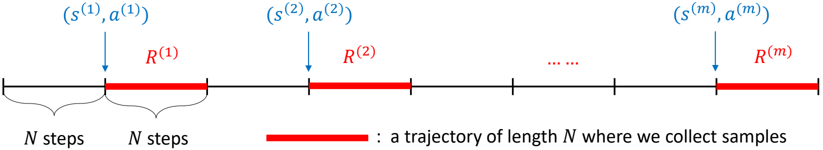
Q: for a fixed π, how to construct $\widehat{w}$ with $\mathbb{E}\left[\Phi(s,a)^\top \widehat{w}\right] = \Phi(s,a)^\top w^\pi + c$?



1. Execute π for $2mN$ steps, where $m = \Theta\left(\frac{1}{\sigma}\right)$ and $N = \Theta\left(t_{\text{mix}}\right)$
2. Collect m trajectories each of length N
3. $$\widehat{w} = \Lambda^{-1}\left(\sum_{i=1}^{m} \Phi(s^{(i)}, a^{(i)}) R^{(i)}\right), \quad \Lambda \triangleq \sum_{i=1}^{m}\sum_{a} \pi_k(a|s^{(i)})\Phi(s^{(i)}, a)\Phi(s^{(i)}, a)^\top$$

Constructing (Nearly) Unbiased Estimators

Q: for a fixed π, how to construct $\widehat{w}$ with $\mathbb{E}\left[\Phi(s,a)^\top \widehat{w}\right] = \Phi(s,a)^\top w^\pi + c$?



1. Execute π for $2mN$ steps, where $m = \Theta\left(\frac{1}{\sigma}\right)$ and $N = \Theta\left(t_{\text{mix}}\right)$
2. Collect m trajectories each of length N
3. $$\widehat{w} = \Lambda^{-1}\left(\sum_{i=1}^{m} \Phi(s^{(i)}, a^{(i)}) R^{(i)}\right), \quad \Lambda \triangleq \sum_{i=1}^{m}\sum_{a} \pi_k(a|s^{(i)})\Phi(s^{(i)}, a)\Phi(s^{(i)}, a)^\top$$

Nearly unbiased estimator: $\mathbb{E}\left[\Phi(s,a)^\top \widehat{w}\right] \approx \Phi(s,a)^\top w^\pi + NJ^\pi$

MDP-EXP2

For epoch $k = 1, \ldots, K$:

1. Execute π_k for $\Theta\left(\frac{t_{\text{mix}}}{\sigma}\right)$ steps and construct $\widehat{w}_k$ as described previously
2. Update the policy:

$$\pi_{k+1}(a|s) \propto \pi_k(a|s) \exp\left(\eta \Phi(s,a)^\top \widehat{w}_k\right)$$

MDP-EXP2

For epoch $k = 1, \ldots, K$:

1. Execute π_k for $\Theta\left(\frac{t_{\text{mix}}}{\sigma}\right)$ steps and construct $\widehat{w}_k$ as described previously
2. Update the policy:

$$\pi_{k+1}(a|s) \propto \pi_k(a|s) \exp\left(\eta \Phi(s,a)^\top \widehat{w}_k\right)$$

Theorem

EXP-MDP2 achieves $\mathbb{E}[\text{Reg}_T] = \widetilde{O}\left(\frac{1}{\sigma}\sqrt{t_{\text{mix}}^3 T}\right)$.

Comparison with Previous Analysis

Politex and **AAPI** are also based on the exponential weight update algorithm, but only get $O(T^{3/4})$ or $O(T^{2/3})$ regret.

Comparison with Previous Analysis

Politex and **AAPI** are also based on the exponential weight update algorithm, but only get $O(T^{3/4})$ or $O(T^{2/3})$ regret.

- **Politex** and **AAPI** use LSPE to construct $\widehat{w}_k$, and argue that it is ϵ-accurate (i.e. $\left\|\widehat{w}_k - w^{\pi_k}\right\| \le \epsilon$) after collecting $O\left(\frac{1}{\epsilon^2}\right)$ samples.

(O hides some constants related to t_{mix} and σ)

Comparison with Previous Analysis

Politex and **AAPI** are also based on the exponential weight update algorithm, but only get $O(T^{3/4})$ or $O(T^{2/3})$ regret.

- **Politex** and **AAPI** use LSPE to construct $\widehat{w}_k$, and argue that it is ϵ-accurate (i.e. $\left\|\widehat{w}_k - w^{\pi_k}\right\| \leq \epsilon$) after collecting $O\left(\frac{1}{\epsilon^2}\right)$ samples.
- In **MDP-EXP2**, we use EXP2 to construct $\widehat{w}_k$, and argue that it is unbiased with constant variance after collecting $O(1)$ samples.

(O hides some constants related to t_{mix} and σ)

Connection with Natural Policy Gradient

It is a folklore (Agarwal et al.'20, Bhandari&Russo'19) that the **Exponential Weight** algorithm has deep connection with **Natural Policy Gradient** (Kakade'02) over softmax policies, as well as TRPO, PPO (Neu et al'17).

Connection with Natural Policy Gradient

It is a folklore (Agarwal et al.'20, Bhandari&Russo'19) that the **Exponential Weight** algorithm has deep connection with **Natural Policy Gradient** (Kakade'02) over softmax policies, as well as TRPO, PPO (Neu et al'17).

$$\pi_k(a|s) \propto \exp(\Theta_k^\top \Phi(s, a))$$

Connection with Natural Policy Gradient

It is a folklore (Agarwal et al.'20, Bhandari&Russo'19) that the **Exponential Weight** algorithm has deep connection with **Natural Policy Gradient** (Kakade'02) over softmax policies, as well as TRPO, PPO (Neu et al'17).

$$\pi_k(a|s) \propto \exp(\Theta_k^\top \Phi(s,a))$$

MDP-EXP2:

$$\Theta_{k+1} \leftarrow \Theta_k + \eta \left(\sum_{i=1}^{m} \mathbb{E}\left[\Phi(s^{(i)}, a)\Phi(s^{(i)}, a)^\top \right] \right)^{-1} \left(\sum_{i=1}^{m} \Phi(s^{(i)}, a^{(i)}) R^{(i)} \right)$$

Connection with Natural Policy Gradient

It is a folklore (Agarwal et al.'20, Bhandari&Russo'19) that the **Exponential Weight** algorithm has deep connection with **Natural Policy Gradient** (Kakade'02) over softmax policies, as well as TRPO, PPO (Neu et al'17).

$$\pi_k(a|s) \propto \exp(\Theta_k^\top \Phi(s,a))$$

MDP-EXP2:

$$\Theta_{k+1} \leftarrow \Theta_k + \eta \left(\sum_{i=1}^{m} \mathbb{E}\left[\Phi(s^{(i)}, a)\Phi(s^{(i)}, a)^\top \right] \right)^{-1} \left(\sum_{i=1}^{m} \Phi(s^{(i)}, a^{(i)}) R^{(i)} \right)$$

NPG:

$$\Theta_{k+1} \leftarrow \Theta_k + \eta \underbrace{\left(\mathbb{E}\left[\left(\nabla_\Theta \log \pi_k(a|s) \right) \left(\nabla_\Theta \log \pi_k(a|s) \right)^\top \right] \right)}_{\text{Fisher information matrix}}^{-1} \underbrace{\left(\sum_{i=1}^{m} \nabla_\Theta \log \pi_k(a^{(i)}|s^{(i)}) R^{(i)} \right)}_{\text{REINFORCE gradient estimator}}$$

Open Problems

1. Is the same regret bound achievable if the learner does not know t_{mix} and σ?
2. How to relax those explorability assumptions? (adding bonus?)

Open Problems

1. Is the same regret bound achievable if the learner does not know t_{mix} and σ?
2. How to relax those explorability assumptions? (adding bonus?)

Open Problems from Part I:

1. An $O(\sqrt{T})$ computationally tractable algorithm for linear MDPs.
2. Sample complexity bound for online RL + linear MDPs + **infinite-horizon discounted setting**.

Summary of the Results

Algorithm	Regret	Assumptions	
		Explorability	Structure
Politex (Abbasi-Yadkori et al.'19a)	$O(T^{\frac{3}{4}})$	UM + UEF	Linear q^π $\forall\pi$
AAPI (Hao et al.'20)	$O(T^{\frac{2}{3}})$		
EE-Politex (Abbasi-Yadkori et al.'19b)	$O(T^{\frac{4}{5}})$	UM + π_e	
MDP-EXP2	$O(\sqrt{T})$	UM + UEF	
FOPO	$O(\sqrt{T})$	BOE	Linear MDP
Optimistic-LSVI	$O(T^{\frac{3}{4}})$		

UM: Uniformly Mixing UEF: Uniformly Excited Features BOE: Bellman Optimality Eqn.

Contributions

- Improving Politex/AAPI's regret bound under the same setting
- First attempt to relax the UM and UEF assumptions

References

- (Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, Gellert Weisz'19a) POLITEX: Regret Bounds for Policy Iteration using Expert Prediction
- (Yasin Abbasi-Yadkori, Nevena Lazic, Csaba Szepesvari, Gellert Weisz'19b) Exploration-Enhanced POLITEX
- (Alekh Agarwal, Sham M. Kakade, Jason D. Lee, Gaurav Mahajan'19) On the Theory of Policy Gradient Methods: Optimality, Approximation, and Distribution Shift
- (Peter Bartlett, Ambuj Tewari'09) REGAL: A Regularization based Algorithm for Reinforcement Learning in Weakly Communicating MDPs
- (Jalaj Bhandari, Daniel Russo'19) Global Optimality Guarantees For Policy Gradient Methods
- (Sébastien Bubeck, Nicolò Cesa-Bianchi, Sham M. Kakade'12) Towards minimax policies for online linear optimization with bandit feedback
- (Varsha Dani, Thomas P. Hayes, Sham M. Kakade'08) The Price of Bandit Information for Online Optimization

References

- (Ronan Fruit, Matteo Pirotta, Alessandro Lazaric, Ronald Ortner'18) Efficient Bias-Span-Constrained Exploration-Exploitation in Reinforcement Learning
- (Botao Hao, Nevena Lazic, Yasin Abbasi-Yadkori, Pooria Joulani, Csaba Szepesvari'20) Provably Efficient Adaptive Approximate Policy Iteration
- (Thomas Jaksch, Ronald Ortner, Peter Auer'10) Near-optimal Regret Bounds for Reinforcement Learning
- (Chi Jin, Zhuoran Yang, Zhaoran Wang, Michael I. Jordan'20) Provably Efficient Reinforcement Learning with Linear Function Approximation
- (Sham M. Kakade'02) A Natural Policy Gradient
- (Gergely Neu, Ciara Pike-Burke'20) A Unifying View of Optimism in Episodic Reinforcement Learning
- (Gergely Neu, Anders Jonsson, Vicenç Gómez'17) A Unified View of Entropy-regularized Markov decision processes

References

- (Jian Qian, Ronan Fruit, Matteo Pirotta, Alessandro Lazaric'19) Exploration Bonus for Regret Minimization in Discrete and Continuous Average Markov Decision Processes
- (Ruosong Wang, Simon S. Du, Lin F. Yang, Ruslan Salakhutdinov'20) On Reward-Free Reinforcement Learning with Linear Function Approximation
- (Chen-Yu Wei, Mehdi Jafarnia-Jahromi, Haipeng Luo, Hiteshi Sharma, Rahul Jain'20) Model-free Reinforcement Learning in Infinite-horizon Average-reward Markov Decision Processes
- (Zihan Zhang, Xiangyang Ji'19) Regret Minimization for Reinforcement Learning by Evaluating the Optimal Bias Function