Echantillonnage adaptatif pour l'identification de la politique optimale dans les PDMs

Aymen Al Marjani[1]
en collaboration avec Alexandre Proutiere[2]

[1]UMPA, ENS Lyon

[2]KTH Royal Institute of Technology

CAp 2021
16 Juin 2021

Outline

Motivation

How many samples does it take to learn an optimal policy in RL ?

Infinite horizon MDPs

$\phi =< \mathcal{S}, \mathcal{A}, p_\phi, q_\phi, \gamma >$

Infinite horizon MDPs

$\phi =< \mathcal{S}, \mathcal{A}, p_\phi, q_\phi, \gamma >$

1. $\mathcal{S}, \mathcal{A}$**: Finite state and action spaces.**

Infinite horizon MDPs

$\phi =< \mathcal{S}, \mathcal{A}, p_\phi, q_\phi, \gamma >$

1. $\mathcal{S}, \mathcal{A}$: **Finite** state and action spaces.
2. After choosing action a at state s the agent:
   - receives reward $R(s,a) \sim q_\phi(.|s,a)$ and mean $r(s,a) \triangleq \mathbb{E}_{q(.|s,a)}[R(s,a)]$.
   - makes transition to $s' \sim p_\phi(.|s,a)$.

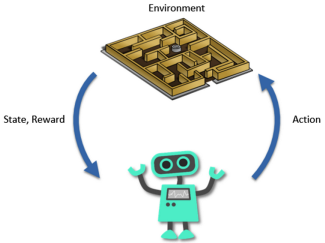


Figure: src:packtpub

Infinite horizon MDPs

$\phi =< \mathcal{S}, \mathcal{A}, p_\phi, q_\phi, \gamma >$

1. $\mathcal{S}, \mathcal{A}$: **Finite** state and action spaces.
2. After choosing action a at state s the agent:
    - receives reward $R(s, a) \sim q_\phi(.|s, a)$ and mean $r(s, a) \triangleq \mathbb{E}_{q(.|s,a)}[R(s, a)]$.
    - makes transition to $s' \sim p_\phi(.|s, a)$.
    - For simplicity, we assume q with support in $[0, 1]$.

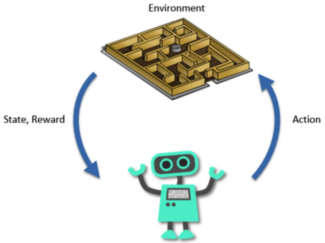


Figure: src:packtpub

Best Policy Identification

$\phi =< \mathcal{S}, \mathcal{A}, p_\phi, q_\phi, \gamma >$

- $\gamma \in [0, 1)$ is the discount factor.

Best Policy Identification

$\phi =< \mathcal{S}, \mathcal{A}, p_\phi, q_\phi, \gamma >$

- $\gamma \in [0, 1)$ is the discount factor.
- Identify a policy $\pi : \mathcal{S} \to \mathcal{A}$ maximizing the total discounted reward:

Best Policy Identification

$\phi = < \mathcal{S}, \mathcal{A}, p_\phi, q_\phi, \gamma >$

- $\gamma \in [0, 1)$ is the discount factor.
- Identify a policy $\pi : \mathcal{S} \to \mathcal{A}$ maximizing the total discounted reward:

$$\pi^\star_\phi \in \arg\max_{\pi} \ V^\pi_\phi(s) = \mathbb{E}_\phi \left[\sum_{t=0}^{\infty} \gamma^t R(s^\pi_t, \pi(s^\pi_t)) \middle| \ s_0 = s \right]$$

Best Policy Identification

$\phi =< \mathcal{S}, \mathcal{A}, p_\phi, q_\phi, \gamma >$

- $\gamma \in [0, 1)$ is the discount factor.
- Identify a policy $\pi : \mathcal{S} \to \mathcal{A}$ maximizing the total discounted reward:

$$\pi^\star_\phi \in \arg\max_\pi \; V^\pi_\phi(s) = \mathbb{E}_\phi\left[\sum_{t=0}^{\infty} \gamma^t R(s^\pi_t, \pi(s^\pi_t)) \middle| \; s_0 = s \right]$$

- **Assumption 1:** $\pi^\star \triangleq \pi^\star_\phi$ is unique.

Sampling schemes

- **Online model:** The agent can only follow trajectories: $(s_0, a_0, R_0, s_1, a_1 \ldots,)$ where $s_{t+1} \sim p_\phi(.|s_t, a_t)$.

Sampling schemes

- **Online model:** The agent can only follow trajectories: $(s_0, a_0, R_0, s_1, a_1 \ldots,)$ where $s_{t+1} \sim p_\phi(.|s_t, a_t)$.

- **Generative model:** At round t, the agent can sample *any* pair (s_t, a_t). She then observes $(R_t, s_t') \sim q_\phi(.|s_t, a_t) \otimes p_\phi(.|s_t, a_t)$. Next, she can choose *any* other pair (s_{t+1}, a_{t+1}) *independently of her previous state*.

Sampling schemes

- **Online model:** The agent can only follow trajectories: $(s_0, a_0, R_0, s_1, a_1 \dots,)$ where $s_{t+1} \sim p_\phi(.|s_t, a_t)$.

- **Generative model:** At round t, the agent can sample *any* pair (s_t, a_t). She then observes $(R_t, s_t') \sim q_\phi(.|s_t, a_t) \otimes p_\phi(.|s_t, a_t)$. Next, she can choose *any* other pair (s_{t+1}, a_{t+1}) *independently of her previous state*.

In this talk, we focus on the Generative model.

δ-PC algorithm

- **Sampling rule:** How to select next pair to sample depending on past observations: (s_{t+1}, a_{t+1}) is $\mathcal{F}_t \triangleq \sigma\big((s_j, a_j, R_j, s'_j)_{1 \leq j \leq t}\big)$ measurable.

δ-PC algorithm

- **Sampling rule:** How to select next pair to sample depending on past observations: (s_{t+1}, a_{t+1}) is $\mathcal{F}_t \triangleq \sigma\big((s_j, a_j, R_j, s_j')_{1\le j\le t}\big)$ measurable.
- **Stopping rule:** The algorithm stops sampling after collecting τ samples and returns $\widehat{\pi}^\star_\tau$. τ is a stopping time w.r.t. the filtration $(\mathcal{F}_t)_{t\ge 1}$.

δ-PC algorithm

- **Sampling rule:** How to select next pair to sample depending on past observations: (s_{t+1}, a_{t+1}) is $\mathcal{F}_t \triangleq \sigma\big((s_j, a_j, R_j, s_j')_{1\leq j\leq t}\big)$ measurable.
- **Stopping rule:** The algorithm stops sampling after collecting τ samples and returns $\widehat{\pi}^\star_\tau$. τ is a stopping time w.r.t. the filtration $(\mathcal{F}_t)_{t\geq 1}$.
- δ-**PC algorithm:** $\mathbb{P}_\phi(\widehat{\pi}^\star_\tau \neq \pi^\star) \leq \delta$.

δ-PC algorithm

- **Sampling rule:** How to select next pair to sample depending on past observations: (s_{t+1}, a_{t+1}) is $\mathcal{F}_t \triangleq \sigma\big((s_j, a_j, R_j, s'_j)_{1\le j\le t}\big)$ measurable.
- **Stopping rule:** The algorithm stops sampling after collecting τ samples and returns $\widehat{\pi}^\star_\tau$. τ is a stopping time w.r.t. the filtration $(\mathcal{F}_t)_{t\ge 1}$.
- δ-**PC algorithm:** $\mathbb{P}_\phi(\widehat{\pi}^\star_\tau \neq \pi^\star) \le \delta$.
- Identify $\pi^\star$ as fast as possible!

δ-PC algorithm

- **Sampling rule:** How to select next pair to sample depending on past observations: (s_{t+1}, a_{t+1}) is $\mathcal{F}_t \triangleq \sigma\big((s_j, a_j, R_j, s'_j)_{1\leq j\leq t}\big)$ measurable.
- **Stopping rule:** The algorithm stops sampling after collecting τ samples and returns $\widehat{\pi}^\star_\tau$. τ is a stopping time w.r.t. the filtration $(\mathcal{F}_t)_{t\geq 1}$.
- δ-**PC algorithm:** $\mathbb{P}_\phi(\widehat{\pi}^\star_\tau \neq \pi^\star) \leq \delta$.
- Identify $\pi^\star$ as fast as possible!

$\implies$ Algorithm with minimal *sample complexity* $\mathbb{E}[\tau_\delta]$

Learning: be specific!

Two kinds of guarantees:

Learning: be specific!

Two kinds of guarantees:

- **Minimax** over a set of MDPs Φ:

$$\inf_{\mathbb{A}:\delta\text{-PC}} \sup_{\phi \in \Phi} \mathbb{E}_{\phi,\mathbb{A}}[\tau_\delta]$$

Learning: be specific!

Two kinds of guarantees:

- **Minimax** over a set of MDPs Φ:

$$\inf_{\mathbb{A}:\delta\text{-PC}} \sup_{\phi\in\Phi} \mathbb{E}_{\phi,\mathbb{A}}[\tau_\delta]$$

- Minimax lower bounds often come from pathological examples. Real world scenarios are not that hard (unless in adversarial settings).

Learning: be specific!

Two kinds of guarantees:

- **Minimax** over a set of MDPs Φ:

$$\inf_{\mathbb{A}:\delta\text{-PC}} \sup_{\phi \in \Phi} \mathbb{E}_{\phi,\mathbb{A}}[\tau_\delta]$$

- Minimax lower bounds often come from pathological examples. Real world scenarios are not that hard (unless in adversarial settings).
- Algorithms that sample state-actions uniformly at random are sufficient to be minimax optimal !

Learning: be specific!

Two kinds of guarantees:

- **Minimax** over a set of MDPs Φ:

$$\inf_{\mathbb{A}:\delta\text{-PC}} \sup_{\phi\in\Phi} \mathbb{E}_{\phi,\mathbb{A}}[\tau_\delta]$$

- **Instance-specific:** For a given ϕ:

$$\inf_{\mathbb{A}:\delta\text{-PC}} \mathbb{E}_{\phi,\mathbb{A}}[\tau_\delta]$$

Learning: be specific!

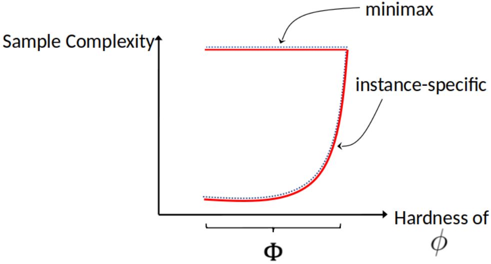
Two kinds of guarantees:

- **Minimax** over a set of MDPs Φ:

$$\inf_{\mathbb{A}:\delta\text{-PC}} \sup_{\phi\in\Phi} \mathbb{E}_{\phi,\mathbb{A}}[\tau_\delta]$$

- **Instance-specific:** For a given ϕ:

$$\inf_{\mathbb{A}:\delta\text{-PC}} \mathbb{E}_{\phi,\mathbb{A}}[\tau_\delta]$$



Learning: be specific!

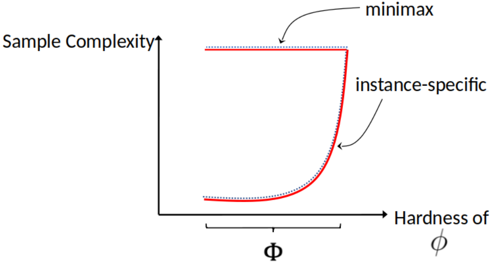
Two kinds of guarantees:

- **Minimax** over a set of MDPs Φ:

$$\inf_{\mathbb{A}:\delta\text{-PC}} \sup_{\phi\in\Phi} \mathbb{E}_{\phi,\mathbb{A}}[\tau_\delta]$$

- **Instance-specific:** For a given ϕ:

$$\inf_{\mathbb{A}:\delta\text{-PC}} \mathbb{E}_{\phi,\mathbb{A}}[\tau_\delta]$$



- We seek algorithms that can adapt to the hardness of the instance.

Information-Theoretical lower bound

Information-Theoretical lower bound

Define:

- The set of alternative MDPs $\mathsf{Alt}(\phi) = \{\psi : \ \pi^\star \text{ is not optimal in } \psi\}$.

Information-Theoretical lower bound

Define:

- The set of alternative MDPs $\mathrm{Alt}(\phi) = \{\psi : \ \pi^\star \text{ is not optimal in } \psi\}$.
- Σ the simplex of $\mathbb{R}^{SA}$.

Information-Theoretical lower bound

Define:

- The set of alternative MDPs $\mathrm{Alt}(\phi) = \{\psi : \ \pi^\star \text{ is not optimal in } \psi\}$.
- Σ the simplex of $\mathbb{R}^{SA}$.
- $\mathrm{KL}_{\phi|\psi}(s,a) = \mathrm{KL}(q_\phi(s,a),\ q_\psi(s,a)) + \mathrm{KL}(p_\phi(s,a),\ p_\psi(s,a))$

Information-Theoretical lower bound

Define:

- The set of alternative MDPs $\mathsf{Alt}(\phi) = \{\psi : \ \pi^\star \text{ is not optimal in } \psi\}$.
- Σ the simplex of $\mathbb{R}^{SA}$.
- $\mathrm{KL}_{\phi|\psi}(s,a) = \mathsf{KL}(q_\phi(s,a),\ q_\psi(s,a)) + \mathsf{KL}(p_\phi(s,a),\ p_\psi(s,a))$

Proposition 1

The sample complexity of any δ-PC algorithm satisfies: for any ϕ with a unique optimal policy,

$$\mathbb{E}_\phi[\tau_\delta] \geq T^\star(\phi) \log(1/2.4\delta),$$

$$\text{where } \ T^\star(\phi)^{-1} = \sup_{\omega \in \Sigma} \inf_{\psi \in \mathsf{Alt}(\phi)} \sum_{s,a} \omega_{sa} \mathrm{KL}_{\phi|\psi}(s,a). \tag{1}$$

Solving the lower bound program?

Recall the value functions:

$$V_\phi^\pi(s) = \mathbb{E}_\phi\left[\sum_{t=0}^{\infty} \gamma^t R(s_t^\pi, \pi(s_t^\pi)) \,\middle|\, s_0 = s\right] = (I - \gamma P_\pi)^{-1} r_\pi$$

$$Q_\phi^\pi(s,a) = r(s,a) + \mathbb{E}_\phi\left[\sum_{t=1}^{\infty} \gamma^t R(s_t^\pi, \pi(s_t^\pi)) \,\middle|\, s_0 = s, a_0 = a\right]$$

Solving the lower bound program?

Recall the value functions:

$$V_\phi^\pi(s) = \mathbb{E}_\phi\left[\sum_{t=0}^{\infty} \gamma^t R(s_t^\pi, \pi(s_t^\pi)) \middle| s_0 = s\right] = (I - \gamma P_\pi)^{-1} r_\pi$$

$$Q_\phi^\pi(s,a) = r(s,a) + \mathbb{E}_\phi\left[\sum_{t=1}^{\infty} \gamma^t R(s_t^\pi, \pi(s_t^\pi)) \middle| s_0 = s, a_0 = a\right]$$

- By definition: $\text{Alt}(\phi) = \{\psi : \exists (s,\pi) \in \mathcal{S} \times \Pi,\ V_\psi^\pi(s) > V_\psi^{\pi^\star}(s)\}$.

Solving the lower bound program?

Recall the value functions:

$$V_\phi^\pi(s) = \mathbb{E}_\phi\left[\sum_{t=0}^{\infty} \gamma^t R(s_t^\pi, \pi(s_t^\pi)) \middle| \, s_0 = s\right] = (I - \gamma P_\pi)^{-1} r_\pi$$

$$Q_\phi^\pi(s,a) = r(s,a) + \mathbb{E}_\phi\left[\sum_{t=1}^{\infty} \gamma^t R(s_t^\pi, \pi(s_t^\pi)) \middle| \, s_0 = s, a_0 = a\right]$$

- By definition: $\mathrm{Alt}(\phi) = \{\psi : \exists (s,\pi) \in \mathcal{S} \times \Pi,\ V_\psi^\pi(s) > V_\psi^{\pi^\star}(s)\}$.
- Involves too many parameters of ψ:

$$\Big(r(x,\pi(x)), p(x,\pi(x)), r(x,\pi^\star(x)), p(x,\pi^\star(x))\Big)_{x \in \mathcal{S}}.$$

Solving the lower bound program?

Recall the value functions:

$$V_\phi^\pi(s) = \mathbb{E}_\phi\left[\sum_{t=0}^{\infty} \gamma^t R(s_t^\pi, \pi(s_t^\pi)) \middle| s_0 = s\right] = (I - \gamma P_\pi)^{-1} r_\pi$$

$$Q_\phi^\pi(s,a) = r(s,a) + \mathbb{E}_\phi\left[\sum_{t=1}^{\infty} \gamma^t R(s_t^\pi, \pi(s_t^\pi)) \middle| s_0 = s, a_0 = a\right]$$

- By definition: $\text{Alt}(\phi) = \{\psi : \exists (s,\pi) \in \mathcal{S} \times \Pi,\ V_\psi^\pi(s) > V_\psi^{\pi^\star}(s)\}$.
- Involves too many parameters of ψ:

$$\Big(r(x,\pi(x)), p(x,\pi(x)), r(x,\pi^\star(x)), p(x,\pi^\star(x))\Big)_{x\in\mathcal{S}}.$$

$\Longrightarrow$ We need further simplifcation.

Solving the lower bound program?

Lemma 2

The set of alternative MDPs can be decomposed as follows:

$$\mathrm{Alt}(\phi) = \bigcup_{(s,a): a \neq \pi^\star(s)} \left\{ \psi : \; Q_{\psi}^{\pi^\star}(s,a) > V_{\psi}^{\pi^\star}(s) \right\}. \tag{2}$$

Solving the lower bound program?

Lemma 2

The set of alternative MDPs can be decomposed as follows:

$$\mathrm{Alt}(\phi) = \bigcup_{(s,a): a \neq \pi^\star(s)} \left\{ \psi : \; Q_{\psi}^{\pi^\star}(s,a) > V_{\psi}^{\pi^\star}(s) \right\}. \qquad (2)$$

- In contrast with $Q_{\phi}^{\pi^\star}(s,a) < V_{\phi}^{\pi^\star}(s)$, for $a \neq \pi^\star(s)$.

Solving the lower bound program?

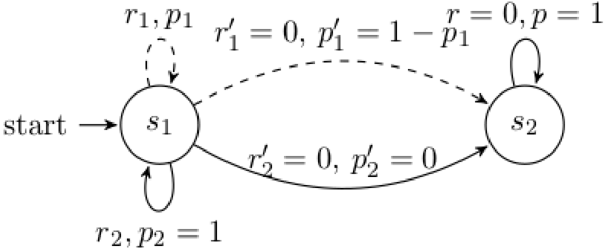
Lemma 2

The set of alternative MDPs can be decomposed as follows:

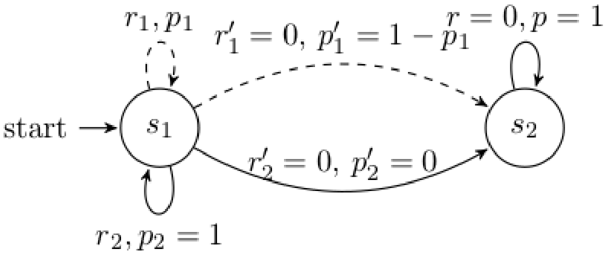
$$\mathrm{Alt}(\phi) = \bigcup_{(s,a): a \neq \pi^\star(s)} \left\{ \psi : \; Q_{\psi}^{\pi^\star}(s,a) > V_{\psi}^{\pi^\star}(s) \right\}. \quad (2)$$

- In contrast with $Q_{\phi}^{\pi^\star}(s,a) < V_{\phi}^{\pi^\star}(s)$, for $a \neq \pi^\star(s)$.
- Only involves $(r(s,a), p(s,a))$ and $\big(r(x,\pi^\star(x)), p(x,\pi^\star(x))\big)_{x\in\mathcal{S}}$ in ψ.

IT Lower bound: Hard to solve!

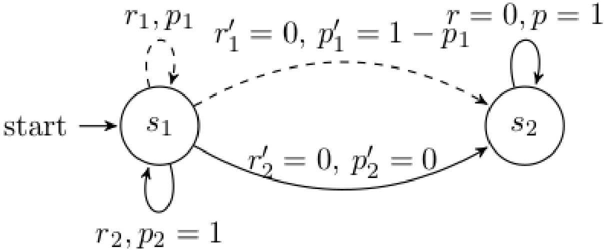


IT Lower bound: Hard to solve!



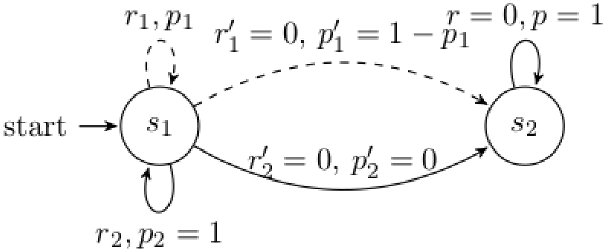
- $Q(s_1, a_i) = \frac{r_i}{1-\gamma p_i},\ \ i = 1, 2.$

IT Lower bound: Hard to solve!



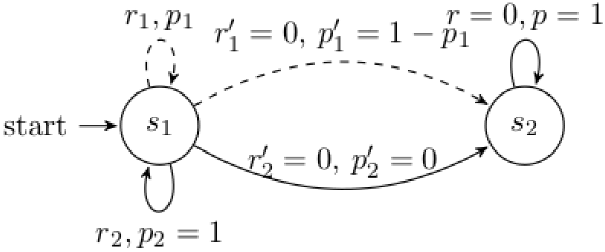
- $Q(s_1, a_i) = \frac{r_i}{1-\gamma p_i},\ \ i = 1, 2.$
- Can easily construct ψ and $\overline{\psi}$ such that:

IT Lower bound: Hard to solve!



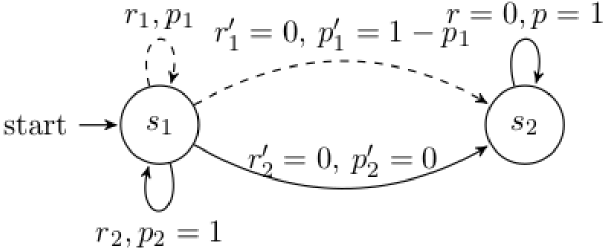
- $Q(s_1, a_i) = \frac{r_i}{1-\gamma p_i},\ \ i = 1, 2.$
- Can easily construct ψ and $\overline{\psi}$ such that:
  - Both ψ and $\overline{\psi}$ satisfy $\frac{r_1}{1-\gamma p_1} > \frac{r_2}{1-\gamma p_2}$.

IT Lower bound: Hard to solve!



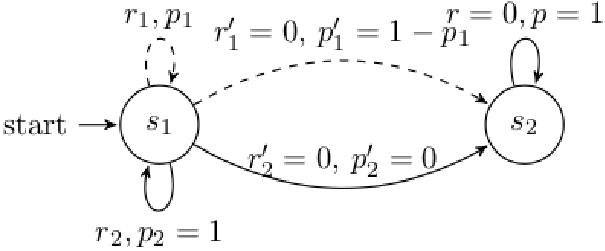
- $Q(s_1, a_i) = \frac{r_i}{1-\gamma p_i},\ i = 1, 2.$
- Can easily construct ψ and $\overline{\psi}$ such that:
  - Both ψ and $\overline{\psi}$ satisfy $\frac{r_1}{1-\gamma p_1} > \frac{r_2}{1-\gamma p_2}$.
  - $\phi = \frac{\psi+\overline{\psi}}{2}$ satisfies $\frac{r_1}{1-\gamma p_1} < \frac{r_2}{1-\gamma p_2}$.

IT Lower bound: Hard to solve!



- Alt(ϕ) and $\mathrm{Alt}_{s_1 a_1}(\phi)$ are not convex.

IT Lower bound: Hard to solve!



- $\text{Alt}(\phi)$ and $\text{Alt}_{s_1 a_1}(\phi)$ are not convex.
- $\Longrightarrow$ The sub-problem $\inf\limits_{\psi \in \text{Alt}(\phi)} \sum\limits_{s,a} \omega_{sa} \text{KL}_{\phi|\psi}(s,a)$ is non-convex.

IT Lower bound: MDP vs MAB

	MAB	**MDP**
Parameters	$\mu_1 > \ldots \geq \mu_K$	$\big(r(s,a), p(s,a)\big)_{s,a}$
Objective	Identify $a^\star = \arg\max_{a \in [K]} \ \mu_a$	Identify $\pi^\star = \arg\max_{\pi} \ (I - \gamma P_\pi)^{-1} r_\pi$
Alternative instances	$\bigcup_{a \neq 1} \{\lambda : \ \lambda_a > \lambda_1\}$ union of convex sets	$\bigcup_{(s,a \neq \pi^\star(s))} \{\psi : \ Q_\psi^{\pi^\star}(s,a) > V_\psi^{\pi^\star}(s)\}$ Not union of convex
IT lower bound	Tractable	Hard to solve

Upper bound: Idea

Define $T(\phi, \omega)^{-1} \triangleq \inf_{\psi \in \mathrm{Alt}(\phi)} \sum_{s,a} \omega_{sa} \mathrm{KL}_{\phi|\psi}(s, a)$.

Upper bound: Idea

Define $T(\phi, \omega)^{-1} \triangleq \inf_{\psi \in \mathrm{Alt}(\phi)} \sum_{s,a} \omega_{sa} \mathrm{KL}_{\phi|\psi}(s, a)$.

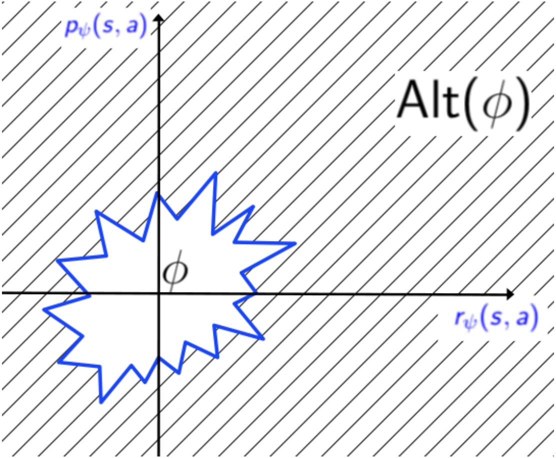


Figure: Alt(ϕ): Non-convex boundary

Upper bound: Idea

Define $T(\phi,\omega)^{-1} \triangleq \inf_{\psi \in \mathrm{Alt}(\phi)} \sum_{s,a} \omega_{sa} \mathrm{KL}_{\phi|\psi}(s,a)$.

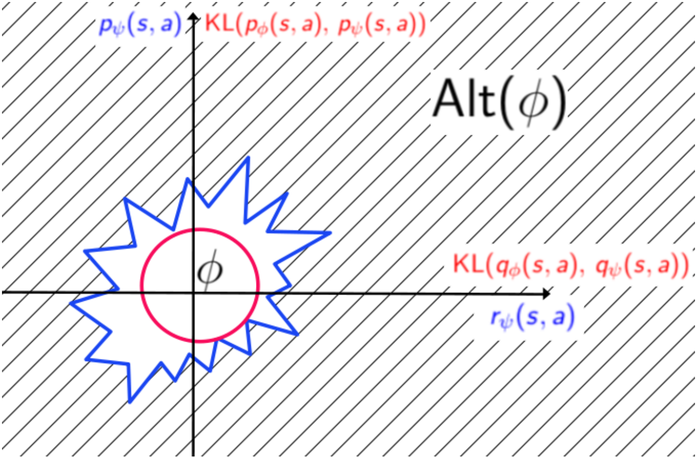


Figure: KL Ball

Upper bound: Instance-specific quantities

Define:

- The sub-optimality gap: $\Delta_{sa} = V^\star_\phi(s) - Q^\star_\phi(s,a)$.
- The minimum gap $\Delta_{\min} = \min_{s, a \neq \pi^\star(s)} \Delta_{sa}$.
- The variance of the value function $\mathrm{Var}_{(s,a)}[V^\star_\phi] = \mathbb{V}_{s' \sim p_\phi(.|s,a)}[V^\star_\phi(s)]$.
- The span of the value function $\mathrm{sp}[V^\star_\phi] = \max_s V^\star_\phi(s) - \min_s V^\star_\phi(s)$.

Upper bound of the characteristic time

Theorem 1 (Upper bound of minimal sample complexity)

For all vectors ω in the simplex:

$$T(\phi,\omega) \leq U(\phi,\omega) \triangleq \max_{s,a\neq\pi^\star(s)} \frac{T_1(s,a;\phi)+T_2(s,a;\phi)}{\omega_{sa}} + \frac{T_3(\phi)+T_4(\phi)}{\min_s \omega_{s,\pi^\star(s)}},$$

$$\text{where} \begin{cases} T_1(s,a;\phi) = \dfrac{2}{\Delta_{sa}^2}, \\ T_2(s,a;\phi) = \max\left(\dfrac{16\mathrm{Var}_{(s,a)}[V_\phi^\star]}{\Delta_{sa}^2}, \dfrac{6\mathrm{sp}[V_\phi^\star]^{4/3}}{\Delta_{sa}^{4/3}}\right), \\ T_3(\phi) = \dfrac{2}{[\Delta_{\min}(\phi)(1-\gamma)]^2}, \\ T_4(\phi) \leq \dfrac{27}{\Delta_{\min}(\phi)^2(1-\gamma)^3} = \mathcal{O}\left(\dfrac{\text{Minimax lower bound}}{SA}\right) \end{cases}$$

KLB-TS: Sampling rule

KLB-TS: Sampling rule

- The optimal weights minimizing the upper-bound program:

$$\overline{\omega}(\phi) = \underset{\omega \in \Sigma}{\arg\inf} \max_{(s,a): a \neq \pi^\star(s)} \frac{T_1(s,a;\phi) + T_2(s,a;\phi)}{\omega_{sa}} + \frac{T_3(\phi) + T_4(\phi)}{\min_s \omega_{s,\pi^\star(s)}}$$

are easy to compute !

KLB-TS: Sampling rule

- The optimal weights minimizing the upper-bound program:

$$\overline{\omega}(\phi) = \arg\inf_{\omega \in \Sigma} \max_{(s,a): a \neq \pi^\star(s)} \frac{T_1(s,a;\phi) + T_2(s,a;\phi)}{\omega_{sa}} + \frac{T_3(\phi) + T_4(\phi)}{\min_s \omega_{s,\pi^\star(s)}}$$

  are easy to compute !

- $\overline{\omega}_{sa} \propto \frac{1+\mathrm{Var}_{p_\phi(s,a)}[V_\phi^\star]}{\Delta_{s,a}^2}$.

KLB-TS: Sampling rule

- The optimal weights minimizing the upper-bound program:

$$\overline{\omega}(\phi) = \arg\inf_{\omega \in \Sigma} \max_{(s,a): a \neq \pi^\star(s)} \frac{T_1(s,a;\phi) + T_2(s,a;\phi)}{\omega_{sa}} + \frac{T_3(\phi) + T_4(\phi)}{\min_s \omega_{s,\pi^\star(s)}}$$

  are easy to compute !

- $\overline{\omega}_{sa} \propto \frac{1+\mathrm{Var}_{p_\phi(s,a)}[V_\phi^\star]}{\Delta_{s,a}^2}$.

- $\overline{\omega}_{s,\pi^\star(s)} \propto \frac{1+\mathrm{Var}_{\max}^\star[V_\phi^\star]}{\Delta_{\min}^2(1-\gamma)^2}$.

KLB-TS: Sampling rule

- The optimal weights minimizing the upper-bound program:

$$\overline{\omega}(\phi) = \arg\inf_{\omega \in \Sigma} \max_{(s,a):a \neq \pi^\star(s)} \frac{T_1(s,a;\phi) + T_2(s,a;\phi)}{\omega_{sa}} + \frac{T_3(\phi) + T_4(\phi)}{\min_s \omega_{s,\pi^\star(s)}}$$

  are easy to compute !

- Use C-Tracking [Garivier and Kaufmann, 2016]:

KLB-TS: Sampling rule

- The optimal weights minimizing the upper-bound program:

$$\overline{\omega}(\phi) = \arg\inf_{\omega \in \Sigma} \max_{(s,a): a \neq \pi^\star(s)} \frac{T_1(s,a;\phi) + T_2(s,a;\phi)}{\omega_{sa}} + \frac{T_3(\phi) + T_4(\phi)}{\min_s \omega_{s,\pi^\star(s)}}$$

  are easy to compute !

- Use C-Tracking [Garivier and Kaufmann, 2016]:
  - Project $\overline{\omega}(\widehat{\phi}_t)$ on $\{\omega \in \Sigma : \forall (s,a),\ \omega_{sa} \geq \frac{1}{\sqrt{t}}\}$ to get $\tilde{\omega}(\widehat{\phi}_t)$.

KLB-TS: Sampling rule

- The optimal weights minimizing the upper-bound program:

$$\overline{\omega}(\phi) = \underset{\omega \in \Sigma}{\arg\inf} \max_{(s,a): a \neq \pi^\star(s)} \frac{T_1(s,a;\phi) + T_2(s,a;\phi)}{\omega_{sa}} + \frac{T_3(\phi) + T_4(\phi)}{\min_s \omega_{s,\pi^\star(s)}}$$

  are easy to compute !

- Use C-Tracking [Garivier and Kaufmann, 2016]:
  - Project $\overline{\omega}(\widehat{\phi}_t)$ on $\{\omega \in \Sigma : \forall (s,a),\ \omega_{sa} \geq \frac{1}{\sqrt{t}}\}$ to get $\tilde{\omega}(\widehat{\phi}_t)$.
  - $(s_{t+1}, a_{t+1}) \in \underset{(s,a) \in \mathcal{S} \times \mathcal{A}}{\arg\max} \sum_{s=1}^{t} \tilde{\omega}_{sa}(\widehat{\phi}_s) - N_{sa}(t)$.

KLB-TS: Sampling rule

- The optimal weights minimizing the upper-bound program:

$$\overline{\omega}(\phi) = \arg\inf_{\omega \in \Sigma} \max_{(s,a): a \neq \pi^\star(s)} \frac{T_1(s,a;\phi) + T_2(s,a;\phi)}{\omega_{sa}} + \frac{T_3(\phi) + T_4(\phi)}{\min_s \omega_{s,\pi^\star(s)}}$$

  are easy to compute !

- Use C-Tracking [Garivier and Kaufmann, 2016]:
  - Project $\overline{\omega}(\widehat{\phi}_t)$ on $\{\omega \in \Sigma : \forall (s,a),\ \omega_{sa} \geq \frac{1}{\sqrt{t}}\}$ to get $\tilde{\omega}(\widehat{\phi}_t)$.
  - $(s_{t+1}, a_{t+1}) \in \underset{(s,a) \in \mathcal{S} \times \mathcal{A}}{\arg\max} \sum_{s=1}^{t} \tilde{\omega}_{sa}(\widehat{\phi}_s) - N_{sa}(t)$.

- Ensures that $\mathbb{P}_\phi\left(\forall (s,a) \in \mathcal{S} \times \mathcal{A}, \quad \lim_{t \to \infty} \frac{N_{sa}(t)}{t} = \overline{\omega}_{s,a}(\phi)\right) = 1$.

KLB-TS: stopping rule

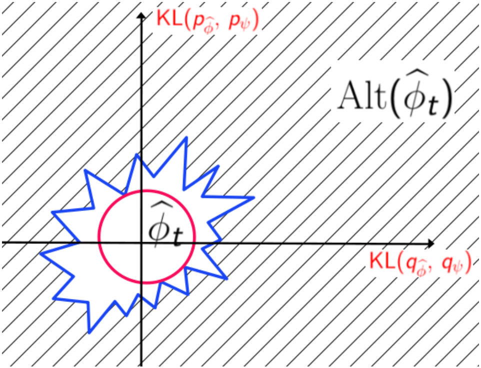


Figure: KL-Ball Stopping rule

KLB-TS: stopping rule

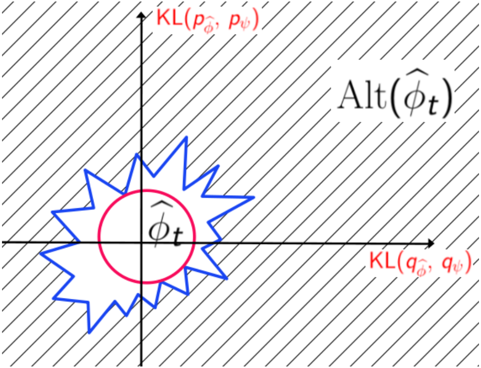


Figure: KL-Ball Stopping rule

- We ensure that ϕ falls within the KL-ball with probability $1-\delta$, using PAC bounds on the KL divergence..

Algorithm: Guarantees

Theorem 3

KLB-TS has a sample complexity τ_δ satisfying:
for all $\delta \in (0,1)$, $\mathbb{E}_\phi[\tau_\delta]$ is finite and $\limsup_{\delta \to 0} \frac{\mathbb{E}_\phi[\tau_\delta]}{\log(1/\delta)} \leq 4U(\phi)$, where:

$$
\begin{aligned}
U(\phi) &\triangleq \inf_\omega U(\phi,\omega) \\
&= \mathcal{O}\Bigg(S \min\left(\frac{\mathrm{Var}^\star_{\max}[V^\star_\phi]}{\Delta^2_{\min}(1-\gamma)^2}, \frac{1}{\Delta^2_{\min}(1-\gamma)^3} \right) \\
&\quad + \sum_{s,a\neq\pi^\star(s)} \frac{1+\mathrm{Var}_{(s,a)}[V^\star_\phi]}{\Delta^2_{s,a}} \Bigg)
\end{aligned}
$$

- $\mathrm{Var}^\star_{\max}[V^\star_\phi] = \max_s \mathrm{Var}_{(s,\pi^\star(s))}[V^\star_\phi]$.

Comparison with State of the Art: BESPOKE

1. Advantages:

Comparison with State of the Art: BESPOKE

1. Advantages:
   - Provides a clear stopping rule to find an ε-optimal policy.

Comparison with State of the Art: BESPOKE

1. Advantages:
   - Provides a clear stopping rule to find an ε-optimal policy.
   - First problem-specific bound, w.h.p:

$$\tau_\delta \leq \tilde{\mathcal{O}}\Bigg(\sum_{s \in \mathcal{S}} \min\left(\frac{1}{(1-\gamma)^3 \Delta_{\min}^2}, \frac{\mathrm{Var}_{(s,\pi^\star(s))}[R] + \gamma^2 \mathrm{Var}_{(s,\pi^\star(s))}[V_\phi^\star]}{\Delta_{\min}^2} \right) + \sum_{s, a \neq \pi^\star(s)} \frac{\mathrm{Var}[R(s,a)] + \gamma^2 \mathrm{Var}_{p(s,a)}[V_\phi^\star]}{\Delta_{sa}^2} + \frac{S^2 A}{(1-\gamma)^2} \Bigg).$$

Comparison with State of the Art: BESPOKE

1. Advantages:
   - Provides a clear stopping rule to find an ε-optimal policy.
   - First problem-specific bound, w.h.p:

$$\tau_\delta \leq \tilde{\mathcal{O}}\left(\sum_{s \in \mathcal{S}} \min\left(\frac{1}{(1-\gamma)^3 \Delta_{\min}^2}, \frac{\mathrm{Var}_{(s,\pi^\star(s))}[R] + \gamma^2 \mathrm{Var}_{(s,\pi^\star(s))}[V_\phi^\star]}{\Delta_{\min}^2} \right) + \sum_{s, a \neq \pi^\star(s)} \frac{\mathrm{Var}[R(s,a)] + \gamma^2 \mathrm{Var}_{p(s,a)}[V_\phi^\star]}{\Delta_{sa}^2} + \frac{S^2 A}{(1-\gamma)^2} \right).$$

2. Drawbacks:
   - Solves a convex problem at every step.

Comparison with State of the Art: BESPOKE

1. Advantages:
   - Provides a clear stopping rule to find an ε-optimal policy.
   - First problem-specific bound, w.h.p:

$$\tau_\delta \leq \tilde{\mathcal{O}}\Bigg(\sum_{s\in\mathcal{S}} \min\left(\frac{1}{(1-\gamma)^3\Delta_{\min}^2}, \frac{\mathrm{Var}_{(s,\pi^\star(s))}[R] + \gamma^2 \mathrm{Var}_{(s,\pi^\star(s))}[V_\phi^\star]}{\Delta_{\min}^2} \right) + \sum_{s,a\neq\pi^\star(s)} \frac{\mathrm{Var}[R(s,a)] + \gamma^2\mathrm{Var}_{p(s,a)}[V_\phi^\star]}{\Delta_{sa}^2} + \frac{S^2 A}{(1-\gamma)^2} \Bigg).$$

2. Drawbacks:
   - Solves a convex problem at every step.
   - Large burn-in phase: $\Omega\big(\frac{S^2 A \log(1/\delta)}{(1-\gamma)^2}\big)$.

Experiments

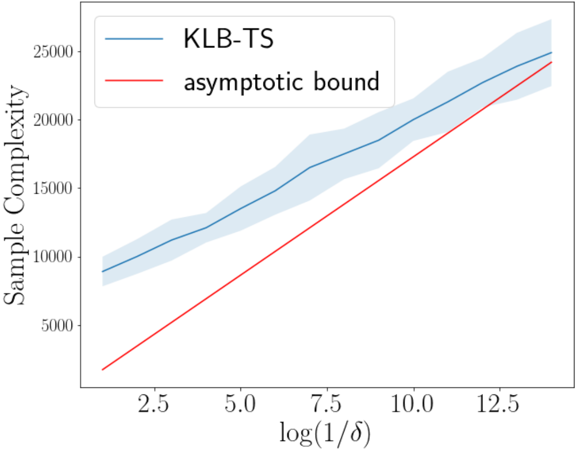


Figure: Asymptotic bound: S=A=2, $\gamma = 0.5$.

Experiments

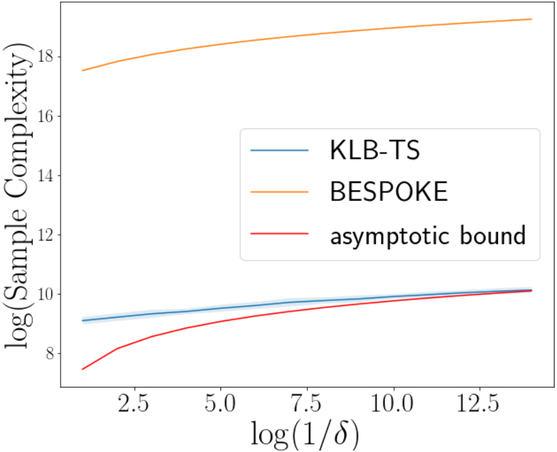


Figure: KLB-TS vs. BESPOKE. S=A=2, $\gamma = 0.5$.

Experiments

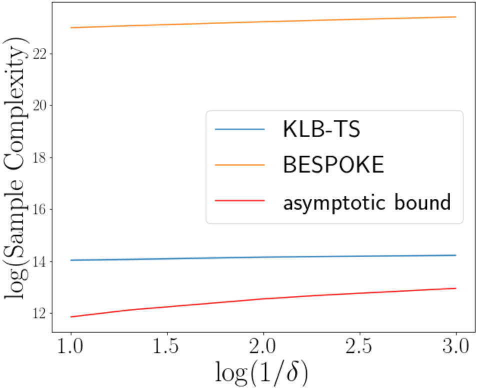


Figure: KLB-TS vs. BESPOKE. $S = 5, A = 10, \gamma = 0.7$.

- Most of BESPOKE's sample complexity comes from the burn-in phase $\Omega\left(\frac{S^2 A \log(1/\delta)}{(1-\gamma)^2}\right)$.

Conclusion

1. Algorithms designed using *problem-specific* bounds can achieve better sample complexity than minimax ones.
2. Contrary to MAB, IT lower bound is hard to solve for MDPs.
3. We can derive problem-specific surrogates which :
   - Are *explicit*, depending on functionals of the MDP.
   - Have a corresponding allocation that is easy to compute.
4. Can be used to devise (Asymptocically) Matching algorithm.
5. First step towards understanding problem-specific ε-optimal policy identification.

Merci !

References I

Agarwal, A., Kakade, S., and Yang, L. F. (2020). Model-based reinforcement learning with a generative model is minimax optimal. volume 125 of *Proceedings of Machine Learning Research*, pages 67–83. PMLR.

Azar, M. G., Munos, R., and Kappen, H. J. (2013). Minimax pac bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning*, 91(3):325–349.

Garivier, A. and Kaufmann, E. (2016). Optimal best arm identification with fixed confidence. In Feldman, V., Rakhlin, A., and Shamir, O., editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 998–1027, Columbia University, New York, New York, USA. PMLR.

References II

Kearns, M. and Singh, S. (1999). Finite-sample convergence rates for q-learning and indirect algorithms. *Advances in Neural Information Processing*, 11.

Sidford, A., Wang, M., Wu, X., Yang, L., and Ye, Y. (2018a). Near-optimal time and sample complexities for solving markov decision processes with a generative model. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 5186–5196. Curran Associates, Inc.

Sidford, A., Wang, M., Wu, X., and Ye, Y. (2018b). Variance reduced value iteration and faster algorithms for solving markov decision processes. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '18, page 770–787, USA. Society for Industrial and Applied Mathematics.

References III

Zanette, A., Kochenderfer, M. J., and Brunskill, E. (2019). Almost horizon-free structure-aware best policy identification with a generative model. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 5625–5634. Curran Associates, Inc.