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

Aymen Al Marjani\textsuperscript{1}

en collaboration avec Alexandre Proutiere\textsuperscript{2}

\textsuperscript{1}UMPA, ENS Lyon

\textsuperscript{2}KTH Royal Institute of Technology

CAp 2021
16 Juin 2021
How many samples does it take to learn an optimal policy in RL?
Infinite horizon MDPs

\[ \phi = \langle S, A, p_\phi, q_\phi, \gamma \rangle \]
\( \phi = \langle S, A, p_\phi, q_\phi, \gamma \rangle \)

1. \( S, A \): Finite state and action spaces.
$\phi = \langle S, A, p_\phi, q_\phi, \gamma \rangle$

1. $S, A$: **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

\[ \phi = \langle S, A, p_\phi, q_\phi, \gamma \rangle \]

1. \( S, A \): **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

\[ \phi = \langle S, A, p_\phi, q_\phi, \gamma \rangle \]

- \( \gamma \in [0, 1) \) is the discount factor.
Best Policy Identification

\[ \phi = \langle S, A, p_\phi, q_\phi, \gamma \rangle \]

- \( \gamma \in [0, 1) \) is the discount factor.
- Identify a policy \( \pi : S \rightarrow A \) maximizing the total discounted reward:

Assumption 1: \( \pi^* \triangleq \pi^* \phi \) is unique.
Best Policy Identification

\( \phi \leq< S, A, p_\phi, q_\phi, \gamma > \)

- \( \gamma \in [0, 1) \) is the discount factor.
- Identify a policy \( \pi : S \rightarrow A \) maximizing the total discounted reward:

\[ \pi^* \in \arg \max \ V_\phi^\pi (s) = \mathbb{E}_\phi \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t^\pi, \pi(s_t^\pi)) \bigg| s_0 = s \right] \]
Best Policy Identification

\[ \phi = \langle S, A, p_\phi, q_\phi, \gamma \rangle \]

- \( \gamma \in [0, 1) \) is the discount factor.
- Identify a policy \( \pi : S \rightarrow A \) maximizing the total discounted reward:

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

- **Assumption 1:** \( \pi^* \triangleq \pi^*_\phi \) is unique.
Online model: The agent can only follow trajectories: 
\((s_0, a_0, R_0, s_1, a_1 \ldots, )\) where \(s_{t+1} \sim p_{\phi}(\cdot|s_t, a_t)\).
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: 
\[(s_0, a_0, R_0, s_1, a_1 \ldots, ) \] where \( s_{t+1} \sim p_{\phi}(\cdot | s_t, a_t) \).

- **Generative model:** At round \( t \), the agent can sample any pair \( (s_t, a_t) \). She then observes \( (R_t, s'_t) \sim q_{\phi}(\cdot | s_t, a_t) \otimes p_{\phi}(\cdot | s_t, a_t) \). 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**.
**Sampling rule:** How to select next pair to sample depending on past observations: \((s_{t+1}, a_{t+1})\) is \(\mathcal{F}_t \triangleq \sigma((s_j, a_j, R_j, s'_j)_{1 \leq j \leq t})\) measurable.
δ-PC algorithm

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

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

**Stopping rule:** The algorithm stops sampling after collecting \(\tau\) samples and returns \(\hat{\pi}^*_\tau\). \(\tau\) is a stopping time w.r.t. the filtration \((\mathcal{F}_t)_{t \geq 1}\).

**\(\delta\)-PC algorithm:** \(\mathbb{P}_\phi(\hat{\pi}^*_\tau \neq \pi^*) \leq \delta\).
**Sampling rule:** How to select next pair to sample depending on past observations: \((s_{t+1}, a_{t+1})\) is \(\mathcal{F}_t \triangleq \sigma \left( (s_j, a_j, R_j, s'_j)_{1 \leq j \leq t} \right)\) measurable.

**Stopping rule:** The algorithm stops sampling after collecting \(\tau\) samples and returns \(\hat{\pi}^\tau\). \(\tau\) is a stopping time w.r.t. the filtration \((\mathcal{F}_t)_{t \geq 1}\).

**\(\delta\)-PC algorithm:** \(\mathbb{P}_\phi(\hat{\pi}^\tau \neq \pi^*) \leq \delta\).

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

**Stopping rule:** The algorithm stops sampling after collecting \(\tau\) samples and returns \(\hat{\pi}_\tau^*\). \(\tau\) is a stopping time w.r.t. the filtration \((\mathcal{F}_t)_{t \geq 1}\).

**\(\delta\)-PC algorithm:** \(\mathbb{P}_\phi(\hat{\pi}_\tau^* \neq \pi^*) \leq \delta\).

**Identify \(\pi^*\) as fast as possible!**

\[\implies \text{Algorithm with minimal sample complexity } \mathbb{E}[\tau_\delta]\]
Learning: be specific!

Two kinds of guarantees:
Two kinds of guarantees:

- **Minimax** over a set of MDPs $\Phi$:

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

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!
Two kinds of guarantees:

- **Minimax** over a set of MDPs $\Phi$:

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

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

- **Minimax** over a set of MDPs $\Phi$:

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

- 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!
Two kinds of guarantees:

- **Minimax** over a set of MDPs $\Phi$:

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

- **Instance-specific**: For a given $\phi$:

\[
\inf_{A: \delta-PC} \mathbb{E}_{\phi, A}[\tau_\delta]
\]
Two kinds of guarantees:

- **Minimax** over a set of MDPs $\Phi$:
  \[
  \inf_{A: \delta\text{-PC}} \sup_{\phi \in \Phi} \mathbb{E}_{\phi,A}[\tau_\delta]
  \]

- **Instance-specific**: For a given $\phi$:
  \[
  \inf_{A: \delta\text{-PC}} \mathbb{E}_{\phi,A}[\tau_\delta]
  \]
Two kinds of guarantees:

- **Minimax** over a set of MDPs $\Phi$:
  \[
  \inf_{A:\delta\text{-PC}} \sup_{\phi \in \Phi} \mathbb{E}_{\phi,A}[\tau_\delta]
  \]

- **Instance-specific**: For a given $\phi$:
  \[
  \inf_{A:\delta\text{-PC}} \mathbb{E}_{\phi,A}[\tau_\delta]
  \]

- We seek algorithms that can adapt to the hardness of the instance.
Information-Theoretical lower bound

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

Let $\Sigma$ the simplex of $\mathbb{R}^{SA}$.

$$\text{KL}(\phi | \psi)(s, a) = \text{KL}(q_{\phi}(s, a), q_{\psi}(s, a)) + \text{KL}(p_{\phi}(s, a), p_{\psi}(s, a))$$

Proposition 1: The sample complexity of any $\delta$-PC algorithm satisfies: for any $\phi$ with a unique optimal policy, $E_{\phi}[\tau_{\delta}] \geq T^{\star}(\phi) \log \left(\frac{1}{2.4 \delta}\right)$, where $T^{\star}(\phi) - 1 = \sup_{\omega \in \Sigma} \inf_{\psi \in \text{Alt}(\phi)} \sum_{s, a \omega} q_{\omega}(s, a)$.
Define:

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

Proposition 1

The sample complexity of any $\delta$-PC algorithm satisfies: for any $\phi$ with a unique optimal policy,

$$E_{\phi}[\tau_{\delta}] \geq T^* (\phi) \log (1/2.4\delta),$$

where $T^* (\phi) = \sup_{\omega \in \Sigma} \inf_{\psi \in \text{Alt}(\phi)} \sum_{s, a} \omega_{sa} KL(\phi | \psi(s, a))$. (1)
Define:

- The set of alternative MDPs $\text{Alt}(\phi) = \{\psi : \pi^* \text{ is not optimal in } \psi\}$.
- $\Sigma$ the simplex of $\mathbb{R}^{SA}$.
Information-Theoretical lower bound

Define:

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

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

**Proposition 1**

The sample complexity of any $\delta$-PC algorithm satisfies: for any $\phi$ with a unique optimal policy,

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

where $T^*(\phi)^{-1} = \sup_{\omega \in \Sigma} \inf_{\psi \in \text{Alt}(\phi)} \sum_{s, a} \omega_{sa} \text{KL}_{\phi|\psi}(s, a)$.  \quad (1)
Recall the value functions:

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

\[ Q_\phi^\pi(s, a) = r(s, a) + \mathbb{E}_\phi \left[ \sum_{t=1}^{\infty} \gamma^t R(s^\pi_t, \pi(s^\pi_t)) \right| s_0 = s, a_0 = a \]
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: \( \text{Alt}(\phi) = \{ \psi : \exists (s, \pi) \in S \times \Pi, \ V_{\psi}^\pi(s) > V_{\psi}^{\pi^*}(s) \} \).
Solving the lower bound program?

Recall the value functions:

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

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

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

\[ \left( r(x, \pi(x)), p(x, \pi(x)), r(x, \pi^*(x)), p(x, \pi^*(x)) \right)_{x \in S} \]
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: \( \text{Alt}(\phi) = \{ \psi : \exists (s, \pi) \in S \times \Pi, \ V_\psi^\pi(s) > V_\phi^\pi(s) \} \).
- Involves too many parameters of \( \psi \):

\[ \left( r(x, \pi(x)), p(x, \pi(x)), r(x, \pi^*(x)), p(x, \pi^*(x)) \right)_{x \in S} \]

\[ \implies \text{We need further simplification.} \]
Solving the lower bound program?

Lemma 2

The set of alternative MDPs can be decomposed as follows:

\[ \text{Alt}(\phi) = \bigcup_{(s,a): a \neq \pi^*(s)} \{ \psi : Q_{\psi}^\pi(s, a) > V_{\psi}^\pi(s) \} \]  \hspace{1cm} (2)
Solving the lower bound program?

Lemma 2

The set of alternative MDPs can be decomposed as follows:

\[
\text{Alt}(\phi) = \bigcup_{(s,a): a \neq \pi^*(s)} \{\psi : Q_{\psi}^\pi(s, a) > V_{\psi}^\pi(s)\}.
\] (2)

- In contrast with \( Q_{\phi}^\pi(s, a) < V_{\phi}^\pi(s) \), for \( a \neq \pi^*(s) \).
Lemma 2

The set of alternative MDPs can be decomposed as follows:

\[
\text{Alt}(\phi) = \bigcup_{(s,a): a \neq \pi^*(s)} \{ \psi : Q_{\psi}^{\pi^*}(s, a) > V_{\psi}^{\pi^*}(s) \}.
\]  

(2)

- In contrast with \( Q_{\phi}^{\pi^*}(s, a) < V_{\phi}^{\pi^*}(s) \), for \( a \neq \pi^*(s) \).
- Only involves \((r(s, a), p(s, a))\) and \((r(x, \pi^*(x)), p(x, \pi^*(x)))\)\( \forall x \in S \) in \( \psi \).
IT Lower bound: Hard to solve!

\[ \text{Alt}(\phi) \text{ and } \text{Alt}(s_1 a_1(\phi)) \text{ are not convex.} \]

\[ \text{The sub-problem } \min_{\psi \in \text{Alt}(\phi)} \sum_{s,a} \omega \cdot s, a \cdot KL(\phi|\psi(s,a)) \text{ is non-convex.} \]
IT Lower bound: Hard to solve!

\[ Q(s_1, a_i) = \frac{r_i}{1 - \gamma p_i}, \quad i = 1, 2. \]
IT Lower bound: Hard to solve!

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

- Can easily construct \( \psi \) and \( \overline{\psi} \) such that:

\begin{align*}
\text{Alt}(\phi) \quad & \text{and} \quad \text{Alt}_{s_1 a_1}(\phi) \quad \text{are not convex.} \\
\implies \quad & \text{The sub-problem} \quad \inf_{\psi \in \text{Alt}(\phi)} \sum_{s, a} w_{sa} \text{KL}(\phi || \psi(s, a)) \quad \text{is non-convex.}
\end{align*}
IT Lower bound: Hard to solve!

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

Can easily construct $\psi$ and $\bar{\psi}$ such that:
- Both $\psi$ and $\bar{\psi}$ satisfy $\frac{r_1}{1 - \gamma p_1} > \frac{r_2}{1 - \gamma p_2}$. 
IT Lower bound: Hard to solve!

\[ Q(s_i, a_i) = \frac{r_i}{1 - \gamma p_i}, \quad i = 1, 2. \]

Can easily construct \( \psi \) and \( \overline{\psi} \) such that:

- Both \( \psi \) 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!

- $\text{Alt}(\phi)$ and $\text{Alt}_{s_1a_1}(\phi)$ are not convex.
IT Lower bound: Hard to solve!

- Alt(\(\phi\)) and Alt_{s_1a_1}(\(\phi\)) are not convex.
- \(\implies\) The sub-problem \(\inf_{\psi \in \text{Alt}(\phi)} \sum_{s,a} \omega_{sa} \text{KL}_{\phi|\psi}(s, a)\) is non-convex.
### IT Lower bound: MDP vs MAB

<table>
<thead>
<tr>
<th></th>
<th>MAB</th>
<th>MDP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td>$\mu_1 &gt; \ldots \geq \mu_K$</td>
<td>$(r(s, a), p(s, a))_{s,a}$</td>
</tr>
<tr>
<td><strong>Objective</strong></td>
<td>Identify $a^* = \arg \max_{a \in [K]} \mu_a$</td>
<td>Identify $\pi^* = \arg \max_{\pi} (I - \gamma P_{\pi})^{-1} r_\pi$</td>
</tr>
<tr>
<td><strong>Alternative</strong></td>
<td>$\bigcup { \lambda : \lambda_a &gt; \lambda_1 }_{a \neq 1}$</td>
<td>$\bigcup { \psi : Q_{\pi^<em>}^\pi(s, a) &gt; V_{\pi^</em>}^\psi(s) }_{(s, a \neq \pi^*(s)}$</td>
</tr>
<tr>
<td><strong>IT lower bound</strong></td>
<td>Tractable</td>
<td>Hard to solve</td>
</tr>
<tr>
<td></td>
<td>union of convex sets</td>
<td>Not union of convex</td>
</tr>
</tbody>
</table>
Define $T(\phi, \omega)^{-1} \triangleq \inf_{\psi \in \text{Alt}(\phi)} \sum_{s,a} \omega_{sa} \text{KL}_{\phi | \psi}(s, a)$. 
Upper bound: Idea

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

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

**Figure: KL Ball**
Define:

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

**Theorem 1 (Upper bound of minimal sample complexity)**

For all vectors $\omega$ in the simplex:

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

where

$$T_1(s, a; \phi) = \frac{2}{\Delta_{sa}^2},$$

$$T_2(s, a; \phi) = \max \left( \frac{16\text{Var}(s, a)[V_{\phi}^*]}{\Delta_{sa}^2}, \frac{6sp[V_{\phi}^*]^{4/3}}{\Delta_{sa}^{4/3}} \right),$$

$$T_3(\phi) = \frac{2}{[\Delta_{\min}(\phi)(1 - \gamma)]^2},$$

$$T_4(\phi) \leq \frac{27}{\Delta_{\min}(\phi)^2(1 - \gamma)^3} = O \left( \frac{\text{Minimax lower bound}}{SA} \right)$$
KLB-TS: Sampling rule

The optimal weights minimizing the upper-bound program:

\[
ω(φ) = \arg \inf_{ω ∈ Σ} \max_{a \neq \pi}(s, a) : T_1(s, a; φ) + T_2(s, a; φ) + T_3(φ) + T_4(φ) = \omega_{sa} + T_3(φ) + T_4(φ)
\]

are easy to compute!

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

\[
\omega(\hat{φ}_t) \text{ on } \{ω ∈ Σ : \forall (s, a) , ω_{sa} ≥ 1/√t\}
\]

\[
(\hat{s}_t+1, \hat{a}_{t+1}) \in \arg \max_{(s, a) ∈ S × A} \sum_{t=1}^\infty \tilde{ω}_{sa}(\hat{φ}_s) - N_{sa}(t).
\]

Ensures that

\[
P_{φ}(∀(s, a) ∈ S × A, \lim_{t→∞} N_{sa}(t) = ω_{sa}(φ) = 1).
\]
The optimal weights minimizing the upper-bound program:

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

are easy to compute!
The optimal weights minimizing the upper-bound program:

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

are easy to compute!

\[
\omega_{sa} \propto \frac{1 + \text{Var}_{p_{\phi}(s, a)}[V^*]}{\Delta_{s, a}^2}.
\]
KLB-TS: Sampling rule

- The optimal weights minimizing the upper-bound program:

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

are easy to compute!

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

- \(\overline{\omega}_{s, \pi^*(s)} \propto \frac{1 + \text{Var}_{\text{max}}^*[V_{\phi}]}{\Delta_{\text{min}}^2(1 - \gamma)^2\} \).
The optimal weights minimizing the upper-bound program:

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

are easy to compute!

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

$$\text{Project} \overline{w}(\hat{\phi}_t) \text{ on } \{\omega \in \Sigma : \forall (s,a), \omega_{sa} \geq 1/\sqrt{t}\} \text{ to get } \tilde{\omega}(\hat{\phi}_t).$$

$$\text{(st+1, at+1)} \in \arg \max_{(s,a) \in S \times A} \sum_t \tilde{\omega}_{sa}(\hat{\phi}_s) - N_{sa}(t).$$

Ensures that

$$P_{\phi}(\forall (s,a) \in S \times A, \lim_{t \to \infty} N_{sa}(t) = \overline{w}_{s,a}(\phi)).$$

Aymen Al Marjani
Adaptive Sampling for BPI
16 Juin 2021
18 / 29
The optimal weights minimizing the upper-bound program:

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

are easy to compute!

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

- Project $\bar{\omega}(\hat{\phi}_t)$ on $\{\omega \in \Sigma : \forall (s, a), \omega_{sa} \geq \frac{1}{\sqrt{t}}\}$ to get $\tilde{\omega}(\hat{\phi}_t)$. 

Aymen Al Marjani
Adaptive Sampling for BPI
16 Juin 2021 18 / 29
KLB-TS: Sampling rule

- The optimal weights minimizing the upper-bound program:

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

are easy to compute!

- Use C-Tracking [Garivier and Kaufmann, 2016]:
  - Project \(\bar{\omega}(\hat{\phi}_t)\) on \(\{\omega \in \Sigma : \forall (s, a), \omega_{sa} \geq \frac{1}{\sqrt{t}}\}\) to get \(\tilde{\omega}(\hat{\phi}_t)\).
  - \((s_{t+1}, a_{t+1}) \in \arg\max_{(s,a) \in S \times A} \sum_{s=1}^{t} \tilde{\omega}_{sa}(\hat{\phi}_s) - N_{sa}(t)\).
KLB-TS: Sampling rule

- The optimal weights minimizing the upper-bound program:

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

are easy to compute!

- Use C-Tracking [Garivier and Kaufmann, 2016]:
  - Project \( \bar{\omega}(\hat{\phi}_t) \) on \( \{ \omega \in \Sigma : \forall (s, a), \ \omega_{sa} \geq \frac{1}{\sqrt{t}} \} \) to get \( \tilde{\omega}(\hat{\phi}_t) \).
  - \((s_{t+1}, a_{t+1}) \in \arg \max_{(s,a) \in S \times A} \sum_{s=1}^{t} \tilde{\omega}_sa(\hat{\phi}_s) - N_{sa}(t) \).
  - Ensures that \( \mathbb{P}_\phi \left( \forall (s, a) \in S \times A, \ \lim_{t \to \infty} \frac{N_{sa}(t)}{t} = \bar{\omega}_{s,a}(\phi) \right) = 1. \)
We ensure that $\phi$ falls within the KL-ball with probability $1 - \delta$, using PAC bounds on the KL divergence.
We ensure that \( \phi \) 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 $\tau_\delta$ 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:

$$U(\phi) \triangleq \inf_\omega U(\phi, \omega)$$

$$= \mathcal{O}\left( S \min \left( \frac{\text{Var}^\ast_{\text{max}}[V^\ast_\phi]}{\Delta^2_{\text{min}}(1-\gamma)^2}, \frac{1}{\Delta^2_{\text{min}}(1-\gamma)^3} \right) \right)$$

$$+ \sum_{s,a \neq \pi^\ast(s)} \frac{1 + \text{Var}_{(s,a)}[V^\ast_\phi]}{\Delta^2_{s,a}}$$

- $\text{Var}^\ast_{\text{max}}[V^\ast_\phi] = \max_s \text{Var}_{(s,\pi^\ast(s))}[V^\ast_\phi]$. 

Aymen Al Marjani
Comparison with State of the Art: BESPOKE

Advantages:

1. Provides a clear stopping rule to find an $\epsilon$-optimal policy.
2. First problem-specific bound, w.h.p:
   \[ \tau \leq \tilde{O} \left( \sum_{s \in S} \min \left( \frac{1}{1 - \gamma} \Delta^2, \text{Var}(s, \pi^\star(s)) \right) R + \gamma^2 \text{Var}(s, \pi^\star(s)) \phi \right) \Delta^2 \]
Comparison with State of the Art: BESPOKE

1 Advantages:
   - Provides a clear stopping rule to find an \( \varepsilon \)-optimal policy.
Advantages:

- Provides a clear stopping rule to find an $\varepsilon$-optimal policy.

- First problem-specific bound, w.h.p:

$$\tau_\delta \leq \tilde{O}\left(\sum_{s \in S} \min\left(\frac{1}{(1 - \gamma)^3 \Delta_{\min}^2}, \frac{\text{Var}(s, \pi^*(s))[R] + \gamma^2 \text{Var}(s, \pi^*(s))[V^*]}{\Delta_{\min}^2}\right) + \sum_{s, a \neq \pi^*(s)} \frac{\text{Var}[R(s, a)] + \gamma^2 \text{Var}_p(s, a)[V^*]}{\Delta_{sa}^2} + \frac{S^2 A}{(1 - \gamma)^2}\right).$$
Comparison with State of the Art: BESPOKE

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

\[
\tau_\delta \leq \widetilde{O} \left( \sum_{s \in S} \min \left( \frac{1}{(1-\gamma)^3 \Delta_{\min}^2}, \frac{\text{Var}(s, \pi^*(s))[R] + \gamma^2 \text{Var}(s, \pi^*(s))[V^*_\phi]}{\Delta_{\min}^2} \right) 
   + \sum_{s, a \neq \pi^*(s)} \frac{\text{Var}[R(s, a)] + \gamma^2 \text{Var}_p(s, a)[V^*_\phi]}{\Delta_{sa}^2} + \frac{S^2 A}{(1-\gamma)^2} \right).
\]

2 Drawbacks:
   - Solves a convex problem at every step.
Comparison with State of the Art: BESPOKE

**Advantages:**
- Provides a clear stopping rule to find an $\varepsilon$-optimal policy.
- First problem-specific bound, w.h.p:

$$\tau_\delta \leq \tilde{O} \left( \sum_{s \in S} \min \left( \frac{1}{(1 - \gamma)^3 \Delta_{\min}^2}, \frac{\text{Var}(s, \pi^*(s))[R] + \gamma^2 \text{Var}(s, \pi^*(s))[V^*]}{\Delta_{\min}^2} \right) \right)$$

$$+ \sum_{s, a \neq \pi^*(s)} \frac{\text{Var}[R(s, a)] + \gamma^2 \text{Var}_p(s, a)[V^*]}{\Delta_{sa}^2} + \frac{S^2 A}{(1 - \gamma)^2}. $$

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

Figure: Asymptotic bound: $S=A=2$, $\gamma = 0.5$. 
Figure: KLB-TS vs. BESPOKE. $S=A=2$, $\gamma = 0.5$. 
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 $\varepsilon$-optimal policy identification.
Merci !


