Thompson Sampling for Learning Parameterized Markov Decision Processes

Aditya Gopalan & Shie Mannor

COLT, Paris, July 2015
Online Reinforcement Learning

\[ S_1 \quad S_2 \quad S_3 \]

\[ S_n \]
Online Reinforcement Learning

$s_1$  $s_2$  $s_3$  

$s_n$

$a_1$  

$a_2$
Online Reinforcement Learning

\[ p(s, a_1, s') \]

\[ p(s, a_2, s') \]
Online Reinforcement Learning

$r(s, a_1, s')$

$r(s, a_2, s')$
Online Reinforcement Learning

\[ S_1 \]

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Online Reinforcement Learning

$s_2$
Online Reinforcement Learning

$S_3$
Online Reinforcement Learning
Online Reinforcement Learning

$S_6$
Online Reinforcement Learning
Online Reinforcement Learning

\[ S_7 \]
Online Reinforcement Learning

$s_3$
Online Reinforcement Learning

\[ S_4 \]
Online Reinforcement Learning
Online Reinforcement Learning

$s_7$
Online Reinforcement Learning

$S_{11}$
Online Reinforcement Learning
Online Reinforcement Learning
Online Reinforcement Learning

Net reward

\[= 5 + 0 + (-1) + (-1) + 3 + 1 + 0 + 2 + 50 + (-100) + 5 + 20 + 50 = 34\]
Online Reinforcement Learning

- Play actions to maximize $\sum_{t=1}^{T} r(s_t, A_t, s_{t+1})$

equivalently, minimize regret

$$\sum_{t=1}^{T} r(s_t, a^*(s_t), s_{t+1}) - \sum_{t=1}^{T} r(s_t, A_t, s_{t+1})$$

- Interesting case: Parameterized MDPs: $p_{\theta^*}(\cdot)$ and $r_{\theta^*}(\cdot)$ where $\theta^* \in \Theta$
  - # states, # actions could be large but $\Theta$ "small"
  - Parameterization can help generalize!
Thompson Sampling [Thompson 1933]

Imagine ‘fictitious’ prior distribution over parameters $\Theta$
Thompson Sampling

Sample a parameter

\[ \mu \sim \text{Prior} \]
Thompson Sampling

Play greedily wrt $\mu$

(in our case: Play optimal policy for MDP via Value Iteration, Policy Iteration, Linear Programming, ... )
Thompson Sampling

Observe state transitions, rewards & Update prior to posterior (Bayes’ Theorem), and REPEAT

\[ \mathbb{P}[\cdot] \rightarrow \mathbb{P}[\cdot \mid \text{observations}] \]
[G.-Mannor’15] For ergodic MDP parameterizations, and under suitably “nice” priors on $\Theta$, with probability at least $(1 - \delta)$, TSMDP gives regret bounded by

$$B + C \log(T)$$

in $T$ rounds, where $B$ depends on $\delta$, $\Theta$ and the prior, $C$ depends only on $\Theta$, the true model $\theta^*$ and, more importantly, the “effective dimension” of $\Theta$. 
Main Result

[G.-Mannor’15] For ergodic MDP parameterizations, and under suitably “nice” priors on $\Theta$, with probability at least $(1 - \delta)$, TSMDP gives regret bounded by

$$B + C \log(T)$$

in $T$ rounds, where $B$ depends on $\delta$, $\Theta$ and the prior, $C$ depends only on $\Theta$, the true model $\theta^*$ and, more importantly, the “effective dimension” of $\Theta$.

• Implication: Provably rapid, problem-dependent learning when effective dimensionality of MDP is small
• Bayesian Regret [Osband-Russo-Van Roy 2013]

\[ \mathbb{E}_{\text{Bayes}} [R_T] = O \left( \sqrt{d_K d_E T} \right) \]

where \( d_K = \) Kolmogorov dimension of parameterization

\( d_E = \) Eluder dimension of parameterization

• But (a) Bayesian setting, and (b) \( \sqrt{T} \) regret growth
Techniques

• Fairly general technique relying on analyzing posterior concentration via **marginal KL divergences**

• Set up “game” involving play counts of suboptimal policies

• Each suboptimal policy “dies” when its stopping condition is met

• Value of game = Regret scaling \( C \)
A “Distance” Measure

- Marginal KL-Divergence in the parameter space:

\[ D_c(\theta^* || \theta) := \sum_{s_1 \in S} \pi^{(c)}_{s_1} \sum_{s_2 \in S} p_{\theta^*}(s_1, c(s_1), s_2) \log \frac{p_{\theta^*}(s_1, c(s_1), s_2)}{p_{\theta}(s_1, c(s_1), s_2)} \]

\[ = \sum_{s_1 \in S} \pi^{(c)}_{s_1} \text{KL} (p_{\theta^*}(s_1, c(s_1), \cdot) \| p_{\theta}(s_1, c(s_1), \cdot)) \]

for any deterministic policy \( c \)

- Encodes to what degree applying policy \( c \) can “resolve” parameter \( \theta \) from parameter \( \theta^* \)