Sequential Decision Making

Lecture 10 : Exploration in RL

Emilie Kaufmann

M2 Data Science, 2022/2023

From bandit to RL

Solve a multi-armed bandit problem
$=$ maximize rewards in a MDP with one state

The bandit world

- several principles for exploration/exploitation
- efficient algorithms (UCB, Thompson Sampling)
- with regret guarantees

RL algorithms so far

- ϵ-greedy exploration
- algorithms with (sometimes) convergence guarantees that are not very efficient

vs. (more) efficient algorithms with little theoretical understanding

Question : can we be inspired by bandit algorithms to

- propose new RL algorithms
- ... with theoretical guarantees ?

Outline

1. Regret minimization in Reinforcement Learning

2. Bandit tools for Regret Minimization in RL
   - Optimism for Reinforcement Learning
   - Thompson Sampling for Reinforcement Learning

3. Scalable heuristics inspired by those principles

Regret minimization

For simplicity, we will define regret for **episodic MDPs**, in which

$$V^\pi(s) = V_1^\pi(s) = \mathbb{E}^\pi\left[\sum_{h=1}^{H} r(s_t, a_t) \,\middle|\, s_1 = s\right].$$

For each episode $t \in \{1, \dots, T\}$, an episodic RL algorithm

- starts in some initial state $s_1^t \sim \rho$
- selects a policy π^t (based on observations from past episodes)
- uses this policy to generate an episode of length H :

$$s_1^t, a_1^t, r_1^t, s_2^t, \dots, s_H^t, a_H^t, r_H^t$$

where $a_h^t = \pi_h^t(s_h^t)$ and $(r_h^t, s_{h+1}^t) = \text{step}(s_h^t, a_h^t)$

Definition

The (pseudo)-regret of an episodic RL algorithm $\pi = (\pi^t)_{t\in\mathbb{N}}$ in T episodes is

$$\mathcal{R}_T(\pi) = \sum_{t=1}^{T}\left[V^\star(s_1^t) - V^{\pi^t}(s_1^t)\right].$$

Reminder : Minimizing regret in bandits

Small regret requires to introduce the right amount of exploration, which can be done with

- ϵ-greedy

explore uniformly with probability ϵ, otherwise trust the estimated model

- Upper Confidence Bounds algorithms

act as if the optimistic model were the true model

- Thompson Sampling

act as if a model sampled from the posterior distribution were the true model

What is wrong with ε-greedy in RL ?

Example : Q-Learning with ε-greedy

➜ ε-greedy exploration

$$a_t = \begin{cases} \mathrm{argmax}_{a \in \mathcal{A}} \hat{Q}_t(s_t, a) & \text{with probability } 1 - \varepsilon_t \\ \sim \mathcal{U}(\mathcal{A}) & \text{with probability } \epsilon_t \end{cases}$$

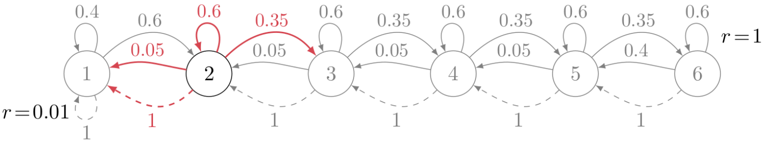
➜ Q-Learning update

$$\hat{Q}_t(s_t, a_t) = \hat{Q}_{t-1}(s_t, a_t) + \alpha_t \left(r_t + \gamma \max_b \hat{Q}_{t-1}(s_t, b) - \hat{Q}_{t-1}(s_t, a_t) \right)$$

⚠ $\hat{Q}_t(s, a)$ is *not* an unbiased estimate of $Q^\star(s, a)$...
(except in the bandit case)

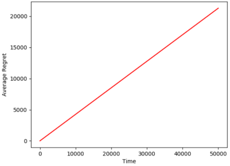
What is wrong with ε-greedy ?

The RiverSwim MDP :

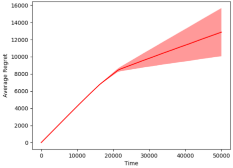


⚠ ε can be hard to tune...

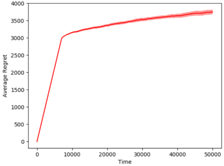
What is wrong with ε-greedy ?



$$\epsilon_t = 0.5$$



$$\epsilon_t = \frac{\epsilon_0}{(N(s_t) - 1000)^{2/3}}$$



$$\epsilon_t = \begin{cases} 1 & \text{if } t < 7000 \\ \frac{\epsilon_0}{\sqrt{N(s_t)}} & \text{otherwise} \end{cases}$$

credit : Alessandro Lazaric

⚠ ε-greedy performs undirected exploration

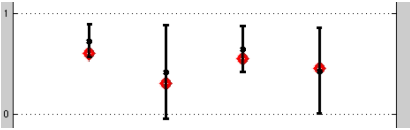
- alternative : model-based methods in which exploration is targeted towards *uncertain regions* of the state/action space

Outline

1. Regret minimization in Reinforcement Learning
2. Bandit tools for Regret Minimization in RL
   - Optimism for Reinforcement Learning
   - Thompson Sampling for Reinforcement Learning
3. Scalable heuristics inspired by those principles

Towards an optimistic learning algorithm

- **Reminder** : Optimistic Bandit model



set of possible bandit models $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3, \mu_4)$:

$$\mathcal{M}_t = \mathcal{I}_1(t) \times \mathcal{I}_2(t) \times \mathcal{I}_3(t) \times \mathcal{I}_4(t)$$

An optimistic bandit model is

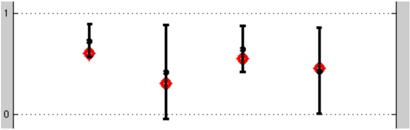
$$\boldsymbol{\mu}_t^+ \in \underset{\boldsymbol{\mu} \in \mathcal{M}_t}{\mathrm{argmax}} \ \mu^\star$$

➜ the best arm in $\boldsymbol{\mu}_t^+$ is $A_t = \underset{a \in \mathcal{A}}{\text{argmax}} \ \mathrm{UCB}_a(t)$

(arm selected by UCB)

Towards an optimistic learning algorithm

- **Reminder** : Optimistic Bandit model



set of possible bandit models $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3, \mu_4)$:

$$\mathcal{M}_t = \mathcal{I}_1(t) \times \mathcal{I}_2(t) \times \mathcal{I}_3(t) \times \mathcal{I}_4(t)$$

An optimistic bandit model is

$$\boldsymbol{\mu}_t^+ \in \underset{\boldsymbol{\mu} \in \mathcal{M}_t}{\mathrm{argmax}} \max_a \mu_a$$

➜ the best arm in $\boldsymbol{\mu}_t^+$ is $A_t = \underset{a \in \mathcal{A}}{\text{argmax}} \ \mathrm{UCB}_a(t)$
(arm selected by UCB)

Towards an optimistic learning algorithm

- **Extension** : Optimistic Markov Decision Process

set of possible MDPs $\boldsymbol{M} = \langle \mathcal{S}, \mathcal{A}, r, p \rangle$:

$$\mathcal{M}_t = \{\langle \mathcal{S}, \mathcal{A}, r, p \rangle : r, p \in \mathcal{B}_t^r \times \mathcal{B}_t^p\}$$

An optimistic Markov Decision Process is

$$\boldsymbol{M}_t^+ \in \underset{\boldsymbol{M} \in \mathcal{M}_t}{\mathrm{argmax}}\, V_{\boldsymbol{M}}^\star(s_1)$$

→ an optimal policy in $\boldsymbol{M}_t^+$ is such that

$$\pi_t^+ \in \underset{\pi}{\mathrm{argmax}} \max_{\boldsymbol{M} \in \mathcal{M}_t} V_{\boldsymbol{M}}^\pi(s_1)$$

Challenges

1. How to construct the set $\mathcal{M}_t$ of possible MDPs ?
2. How to numerically compute π_t^+ ?

Towards an optimistic learning algorithm

- **Extension** : Optimistic Markov Decision Process

$$\text{set of possible MDPs } \boldsymbol{M} = \langle \mathcal{S}, \mathcal{A}, r, p \rangle :$$

$$\mathcal{M}_t = \{\langle \mathcal{S}, \mathcal{A}, r, p \rangle : r, p \in \mathcal{B}_t^r \times \mathcal{B}_t^p\}$$

An optimistic Markov Decision Process is

$$\boldsymbol{M}_t^+ \in \underset{\boldsymbol{M} \in \mathcal{M}_t}{\operatorname{argmax}} \max_{\pi} V_{\boldsymbol{M}}^{\pi}(s_1)$$

➜ an optimal policy in $\boldsymbol{M}_t^+$ is such that

$$\pi_t^+ \in \underset{\pi}{\operatorname{argmax}} \max_{\boldsymbol{M} \in \mathcal{M}_t} V_{\boldsymbol{M}}^{\pi}(s_1)$$

Challenges

1. How to construct the set $\mathcal{M}_t$ of possible MDPs ?
2. How to numerically compute π_t^+ ?

Step 1 : Constructing $\mathcal{M}_t$

$$\mathcal{M}_t = \left\{ \langle \mathcal{S}, \mathcal{A}, r, p \rangle : \forall (s,a) \in \mathcal{S} \times \mathcal{A}, r(s,a) \in \mathcal{B}_t^r(s,a), p(\cdot|s,a) \in \mathcal{B}_t^p(s,a) \right\}$$

Idea : build individual confidence regions

- on the average reward $r(s,a)$: $\mathcal{B}_t^r(s,a) \subseteq \mathbb{R}$
- on the transition probability vector $p(\cdot|s,a)$: $\mathcal{B}_t^p(s,a) \subseteq \Delta(\mathcal{S})$

that rely on the empirical estimates

$$\hat{r}_t(s,a) = \frac{1}{n_t(s,a)} \sum_{i=1}^{n_t(s,a)} r[i] \quad \text{and} \quad \hat{p}_t(s'|s,a) = \frac{n_t(s,a,s')}{n_t(s,a)}$$

$n_t(s,a)$: number of visits of (s,a) until episode t
$n_t(s,a,s')$: number of times s' was the next state when the transition (s,a) was performed until episode t

Goal : $\mathbb{P}_{\boldsymbol{M}}(\boldsymbol{M} \in \mathcal{M}_t)$ is close to 1

Step 1 : Constructing $\mathcal{M}_t$

$$\mathcal{M}_t = \Big\{ \langle \mathcal{S}, \mathcal{A}, r, p \rangle : \forall (s,a) \in \mathcal{S} \times \mathcal{A}, r(s,a) \in \mathcal{B}_t^r(s,a), p(\cdot|s,a) \in \mathcal{B}_t^p(s,a) \Big\}$$

Idea : build individual confidence regions

- on the average reward $r(s,a)$: $\mathcal{B}_t^r(s,a) \subseteq \mathbb{R}$

Assuming bounded rewards,

$$\mathcal{B}_t^r(s,a) = \left[\hat{r}_t(s,a) - \sqrt{\frac{\ln(4(n_t(s,a))^2/\delta)}{2n_t(s,a)}} ; \hat{r}_t(s,a) + \sqrt{\frac{\ln(4(n_t(s,a))^2/\delta)}{2n_t(s,a)}} \right]$$

satisfies

$$\mathbb{P}\Big(\exists t \in \mathbb{N} : r(s,a) \notin \mathcal{B}_t^r(s,a) \Big) \leq \delta.$$

(Hoeffding inequality + union bound)

Step 1 : Constructing $\mathcal{M}_t$

$$\mathcal{M}_t = \left\{ \langle \mathcal{S}, \mathcal{A}, r, p \rangle : \forall (s,a) \in \mathcal{S} \times \mathcal{A}, r(s,a) \in \mathcal{B}_t^r(s,a), p(\cdot|s,a) \in \mathcal{B}_t^p(s,a) \right\}$$

Idea : build individual confidence regions

- on the average reward $r(s,a)$: $\mathcal{B}_t^r(s,a) \subseteq \mathbb{R}$

Assuming bounded rewards,

$$\mathcal{B}_t^r(s,a) = \left[\hat{r}_t(s,a) - \beta_t^r(s,a) ; \hat{r}_t(s,a) + \beta_t^r(s,a) \right]$$

satisfies

$$\mathbb{P}\Big(\exists t \in \mathbb{N} : r(s,a) \notin \mathcal{B}_t^r(s,a) \Big) \leq \delta.$$

(Hoeffding inequality + union bound)

Step 1 : Constructing $\mathcal{M}_t$

$$\mathcal{M}_t = \left\{ \langle \mathcal{S}, \mathcal{A}, r, p \rangle : \forall (s,a) \in \mathcal{S} \times \mathcal{A}, r(s,a) \in \mathcal{B}_t^r(s,a), p(\cdot|s,a) \in \mathcal{B}_t^p(s,a) \right\}$$

Idea : build individual confidence regions

- on the transition probability vector $p(\cdot|s,a)$: $\mathcal{B}_t^p(s,a) \subseteq \Delta(\mathcal{S})$

$$\mathcal{B}_t^p(s,a) = \left\{ p(\cdot|s,a) \in \Delta(\mathcal{S}) : \|p(\cdot|s,a) - \hat{p}_t(\cdot|s,a)\|_1 \leq C\sqrt{\frac{S \ln(n_t(s,a)/\delta)}{n_t(s,a)}} \right\}$$

satisfies

$$\mathbb{P}\Big(\exists t \in \mathbb{N} : p(\cdot|s,a) \notin \mathcal{B}_t^p(s,a)\Big) \leq \delta.$$

(Freedman inequality + union bound)
[Jaksch et al., 2010]

Step 1 : Constructing $\mathcal{M}_t$

$$\mathcal{M}_t = \left\{ \langle \mathcal{S}, \mathcal{A}, r, p \rangle : \forall (s,a) \in \mathcal{S} \times \mathcal{A}, r(s,a) \in \mathcal{B}_t^r(s,a), p(\cdot|s,a) \in \mathcal{B}_t^p(s,a) \right\}$$

Idea : build individual confidence regions

- on the transition probability vector $p(\cdot|s,a)$: $\mathcal{B}_t^p(s,a) \subseteq \Delta(\mathcal{S})$

$$\mathcal{B}_t^p(s,a) = \left\{ p(\cdot|s,a) \in \Delta(\mathcal{S}) : \left\| p(\cdot|s,a) - \hat{p}_t(\cdot|s,a) \right\|_1 \leq \beta_t^p(s,a) \right\}$$

satisfies

$$\mathbb{P}\Big(\exists t \in \mathbb{N} : p(\cdot|s,a) \notin \mathcal{B}_t^p(s,a) \Big) \leq \delta.$$

(Freedman inequality + union bound)
[Jaksch et al., 2010]

Step 1 : Constructing $\mathcal{M}_t$

$$\mathcal{M}_t = \left\{ \langle \mathcal{S}, \mathcal{A}, r, p \rangle : \forall (s,a) \in \mathcal{S} \times \mathcal{A}, r(s,a) \in \mathcal{B}_t^r(s,a), p(\cdot|s,a) \in \mathcal{B}_t^p(s,a) \right\}$$

$$\begin{aligned}
\mathcal{B}_t^r(s,a) &= \left[\hat{r}_t(s,a) - \beta_t^r(s,a) ; \hat{r}_t(s,a) + \beta_t^r(s,a) \right] \\
\mathcal{B}_t^p(s,a) &= \left\{ p(\cdot|s,a) \in \Delta(\mathcal{S}) : \| p(\cdot|s,a) - \hat{p}_t(\cdot|s,a) \|_1 \leq \beta_t^p(s,a) \right\}
\end{aligned}$$

with exploration bonuses :

$$\begin{aligned}
\beta_t^r(s,a) &\propto \sqrt{\frac{\ln(n_t(s,a)/\delta)}{n_t(s,a)}} \\
\beta_t^p(s,a) &\propto \sqrt{\frac{S \ln(n_t(s,a)/\delta)}{n_t(s,a)}}
\end{aligned}$$

Step 2 : Optimistic Value Iteration

Goal : Approximate $\pi^+ \in \underset{\pi}{\mathrm{argmax}} \max_{\boldsymbol{M} \in \mathcal{M}} V^\pi_{\boldsymbol{M}}$ for a set of MDPs

$$\mathcal{M} = \left\{ \langle \mathcal{S}, \mathcal{A}, r, p \rangle : \forall (s,a) \in \mathcal{S} \times \mathcal{A}, r(s,a) \in \mathcal{B}^r(s,a), p(\cdot|s,a) \in \mathcal{B}^p(s,a) \right\}$$

Recall the optimal solution for a fixed MDP : $\pi^\star_h = \text{greedy}(Q^\star_h)$ where

$$Q^\star_h(s,a) = r(s,a) + \sum_{s'} p(s'|s,a) \max_b Q^\star_{h+1}(s',b)$$

➜ $\pi^+_h = \text{greedy}(Q^+_h)$ where

$$Q^+_h(s,a) = \max_{(r,p) \in \mathcal{M}} \left[r(s,a) + \sum_{s'} p(s'|s,a) \max_b Q^+_{h+1}(s',b) \right]$$

Step 2 : Optimistic Value Iteration

$$Q_h^+(s,a) = \max_{(r,p)\in\mathcal{B}^r(s,a)\times\mathcal{B}^p(s,a)} \left[r(s,a) + p(\cdot|s,a)^\top \underbrace{\left(\max_b Q_{h+1}^+(s',b)\right)_{s'\in\mathcal{S}}}_{V_{h+1}^+} \right]$$

$$\begin{aligned}
&= \max_{r\in\mathcal{B}^r(s,a)} r + \max_{p\in\mathcal{B}^p(s,a)} p^\top V_{h+1}^+ \\
&= \hat{r}_t(s,a) + \beta_t^r(s,a) + \max_{p\in\mathcal{B}^p(s,a)} p^\top V_{h+1}^+ \\
&= \hat{r}_t(s,a) + \beta_t^r(s,a) + \hat{p}_t(\cdot|s,a)^\top V_{h+1}^+ + \max_{p\in\mathcal{B}^p(s,a)} (p - \hat{p}_t(\cdot|s,a))^\top V_{h+1}^+ \\
&\leq \hat{r}_t(s,a) + \beta_t^r(s,a) + \hat{p}_t(\cdot|s,a)^\top V_{h+1}^+ + \max_{p\in\mathcal{B}^p(s,a)} \|p - \hat{p}_t(\cdot|s,a)\|_1 \|V_{h+1}^+\|_\infty \\
&= \hat{r}_t(s,a) + \beta_t^r(s,a) + \hat{p}_t(\cdot|s,a)^\top V_{h+1}^+ + \beta_t^p(s,a)(H-h)r_{\max} \\
&= \hat{r}_t(s,a) + \underbrace{\left[\beta_t^r(s,a) + \beta_t^p(s,a)(H-h)r_{\max}\right]}_{\text{exploration bonus}} + \hat{p}_t(\cdot|s,a)^\top V_{h+1}^+
\end{aligned}$$

Optimistic algorithm

A family of algorithms

An **optimistic algorithm** uses in episode $t+1$ the exporation policy $\pi_h^{t+1} = \text{greedy}\left(\overline{Q}_h\right)$ where $\overline{Q}_h(s,a)$ is an optimistic Q-value function

$$\begin{aligned}\overline{Q}_h(s,a) &= \hat{r}_t(s,a) + \beta_t(s,a) + \sum_{s' \in \mathcal{S}} \hat{p}_t(s'|s,a) \max_b \overline{V}_{h+1}(s') \\ \overline{V}_h(s) &= \min\left[H - h; \max_b Q_h(s,b)\right],\end{aligned}$$

where $\beta_t(s,a)$ is some exploration bonus.

From the previous calculation, one can propose

$$\beta_t(s,a) = \beta_t^r(s,a) + C\beta_t^p(s,a) \simeq \sqrt{\frac{\ln(n_t(s,a))}{n_t(s,a)}} + C\sqrt{\frac{S\ln(n_t(s,a))}{n_t(s,a)}}$$

➜ $\beta_t(s,a)$ scales in $1/\sqrt{n_t(s,a)}$ where $n_t(s,a)$ is the number of previous visits to (s,a).

Optimistic algorithm

A family of algorithms

An **optimistic algorithm** uses in episode $t+1$ the exporation policy $\pi_h^{t+1} = \text{greedy}\left(\overline{Q}_h\right)$ where $\overline{Q}_h(s,a)$ is an optimistic Q-value function

$$\begin{aligned} \overline{Q}_h(s,a) &= \hat{r}_t(s,a) + \beta_t(s,a) + \sum_{s' \in \mathcal{S}} \hat{p}_t(s'|s,a) \max_b \overline{V}_{h+1}(s') \\ \overline{V}_h(s) &= \min\left[H - h; \max_b Q_h(s,b)\right], \end{aligned}$$

where $\beta_t(s,a)$ is some exploration bonus.

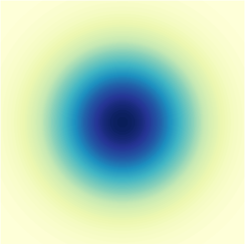
- An example of optimistic algorithm in the episodic setting : UCB-VI [Azar et al., 2017]
- Optimistic algorithms were first proposed in the more complex average-reward MDPs : UCRL [Jaksch et al., 2010]

UCB-VI achieves $R_T = \mathcal{O}(\sqrt{H^2SAT})$ w.h.p.

Outline

Posterior Sampling for RL

Bayesian assumption : $\boldsymbol{M}$ is drawn from some prior distribution ν_0.



$\nu_t \in \Delta(\mathcal{M})$: posterior distribution over the set of MDPs

Optimism	**Posterior Sampling**
Set of possible MDPs	Posterior distribution over MDPs
Compute the optimistic MDP	Sample from the posterior distribution

Posterior Sampling for Episodic RL

Algorithm 1: PSRL

Input : Prior distribution ν_0

for $t = 1, 2, \ldots$ **do**

  $s_1 \sim \rho$ \\\\ get the starting state of episode t

  Sample $\widetilde{M}_t \sim \nu_{t-1}$ \\\\ sample an MDP from the current posterior distribution

  Compute $\tilde{\pi}^t$ an optimal policy for $\widetilde{M}_t$

  for $h = 1, \ldots, H$ **do**

    $a_h = \tilde{\pi}^t_h(s_h)$ \\\\ choose next action according to $\tilde{\pi}^t$

    $r_h, s_{h+1} = \mathrm{step}(s_h, a_h)$

  end

  Compute ν_t based on ν_{t-1} and $\{(s_h, a_h, r_h, s_{h+1})\}_{h=1}^{H}$

end

[Strens, 2000, Osband et al., 2013]

Outline

Limitations of optimistic approaches

An important message from optimistic approaches :

➜ Do not only trust the estimated MDP $\hat{M}_t$, but take into account the uncertainty in the underlying estimate

$$\mathcal{B}_t^r(s,a) = \left[\hat{r}_t(s,a) - \beta_t^r(s,a); \hat{r}_t(s,a) + \beta_t^r(s,a)\right]$$

$$\mathcal{B}_t^p(s,a) = \left\{p(\cdot|s,a) \in \Delta(\mathcal{S}) : \|p(\cdot|s,a) - \hat{p}_t(\cdot|s,a)\|_1 \leq \beta_t^p(s,a)\right\}$$

expressed by exploration bonuses scaling in $\sqrt{\frac{1}{n_t(s,a)}}$ where $n_t(s,a)$ is the count (=number of visits) of (s,a).

Scaling for large state action spaces ?

- each state action pair may be visited only very little...
- UCB-VI is quite inefficient in practice for large state-spaces (efficient, continuous variants is an active research direction)

A heuristic : count-based exploration

General principle

1. Estimate a “proxi” for the number of visits of a state $\tilde{n}_t(s)$
2. Add an exploration bonus directly to the collected rewards :

$$r_t^+ = r_t + c\sqrt{\frac{1}{\tilde{n}_t(s_t)}}$$

3. Run any DeepRL algorithm on

$$\mathcal{D} = \bigcup_t \left\{ (s_t, a_t, r_t^+, s_{t+1}) \right\}$$

A heuristic : count-based exploration

General principle

1. Estimate a "proxi" for the number of visits of a state $\tilde{n}_t(s)$
2. Add an exploration bonus directly to the collected rewards :

$$r_t^+ = r_t + c\sqrt{\frac{1}{\tilde{n}_t(s_t)}}$$

3. Run any DeepRL algorithm on

$$\mathcal{D} = \bigcup_t \left\{ (s_t, a_t, r_t^+, s_{t+1}) \right\}$$

Example of pseudo-counts :

- use a hash function, e.g. $\phi : \mathcal{S} \to \{-1, 1\}^k$
  $n(\phi(s_t)) \leftarrow n(\phi(s_t)) + 1$
  (possibly learn a good hash function)

[Tang et al., 2017]

Limitations of Posterior Sampling

An important message from posterior sampling :

➜ Adding some noise to the estimated MDP $\hat{M}_t$ is helpful !

$$\begin{aligned}\tilde{r}_t(s,a) &= \hat{r}_t(s,a) + \epsilon_t(s,a) \\ \tilde{p}_t(s'|s,a) &= \hat{p}_t(\cdot|s,a) + \epsilon'_t(s,a).\end{aligned}$$

Scaling for large state action spaces ?

- maintaining independent posterior over all state action rewards and transitions can be costly
- more sophisticated prior distributions encoding some structure and the associated posteriors can be hard to sample from

➜ use other type of (non-Bayesian) randomized exploration ?
Noisy Networks [Fortunato et al., 2017]
Bootstrap DQN [Osband et al., 2016] ...

Conclusion : Bandits for RL

Bandits tools are useful for Reinforcement Learning :

- UCRL, PSRL : bandit-based exploration for tabular MDPs
- ... that can motivate "deeper" heuristics

Bandit tools lead to big success in Monte-Carlo planning

- ... without proper sample complexity guarantees

➜ Unifying theory and practice is a big challenge in RL !

Azar, M. G., Osband, I., and Munos, R. (2017).
Minimax regret bounds for reinforcement learning.
In *Proceedings of the 34th International Conference on Machine Learning, (ICML)*.

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C., and Legg, S. (2017).
Noisy networks for exploration.
arXiv :1706.10295.

Jaksch, T., Ortner, R., and Auer, P. (2010).
Near-Optimal regret bounds for reinforcement learning.
Journal of Machine Learning Research, 11 :1563–1600.

Osband, I., Blundell, C., Pritzel, A., and Roy, B. V. (2016).
Deep exploration via bootstrapped DQN.
In *Advances in Neural Information Processing Systems (NIPS)*.

Osband, I., Van Roy, B., and Russo, D. (2013).
(More) Efficient Reinforcement Learning Via Posterior Sampling.
In *Advances in Neural Information Processing Systems*.

Strens, M. (2000).
A Bayesian Framework for Reinforcement Learning.
In *ICML*.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, X., Duan, Y., Schulman, J., Turck, F. D., and Abbeel, P. (2017).
#exploration : A study of count-based exploration for deep reinforcement learning.
In *Advances in Neural Information Processing Systems (NIPS)*.