Guarantees in Reinforcement Learning

Shipra Agrawal

Reinforcement learning

Sequential decision making

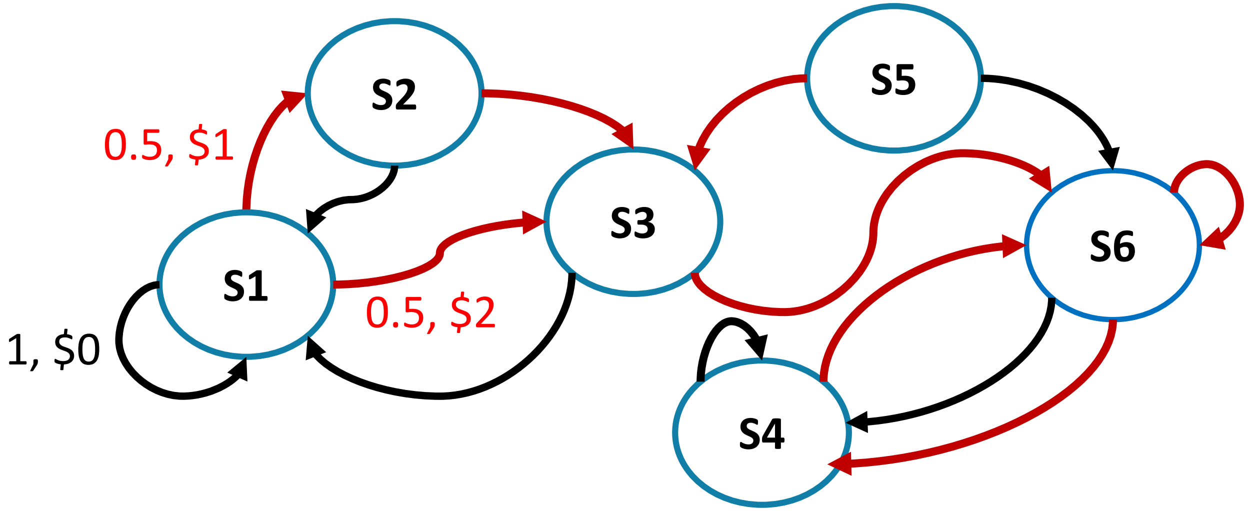
- Rounds $t = 1, \dots, T$
  - Observe state take an action
  - Observe response: reward and new state
- Response to an action depends on the state of the system
- Learn how to make decisions using the response
  - Trial and error method
- Applications: autonomous vehicle control, robot navigation, personalized medical treatments, inventory management, intelligent game playing and problem solving….

The reinforcement learning problem

System dynamics given by an MDP (S, A, P, r, s_0)

Rounds $t = 1, \ldots, T$. In round t,

- observe state s_t, take action a_t,
- observe reward $r_t \in [0,1]$, $\mathrm{E}[r_t] = \boldsymbol{r_{s_t,a_t}}$
- observe the transition to next state s_{t+1} with probability $\mathbf{P}_{\boldsymbol{s_t,a_t,}}(\boldsymbol{s_{t+1}})$



6 states,
2 actions

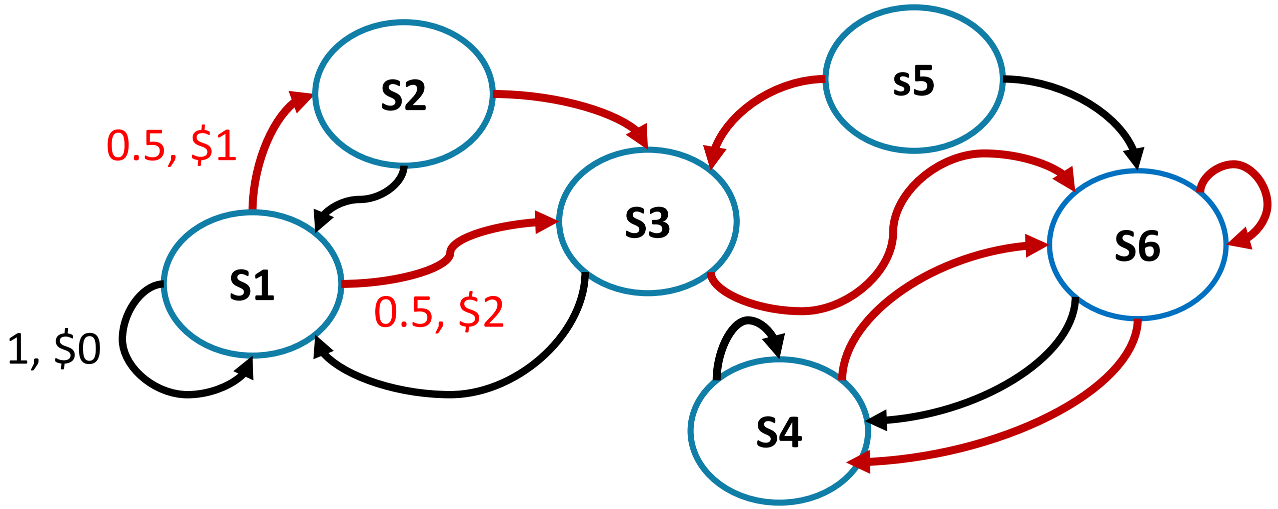
The reinforcement learning problem

Solution concept: optimal policy

- Which action to take in which state
- $\pi_t : S \rightarrow A$, Action $\pi_t(s)$ in state s

Goal: maximize total reward over a time horizon T

- **Unknown** reward distributions R and **unknown** transition function P
- Learn the MDP **from observations** while maximizing reward



6 states,
2 actions

Q-learning (tabular)

Initialize $Q(s, a), \forall s, a$

For $t = 1,2, \ldots ,$

- Take action a_t Greedy $= \max_a Q(s_t, a)$
- Observe reward r_t, next state s_{t+1}
- Update

$$Q^{t+1}(s_t, a_t) \leftarrow (1 - \alpha_t) Q^t(s_t, a_t) + \alpha_t \left(r_t + \gamma \max_{a'} Q^t(s_{t+1}, a')\right)$$

Q-learning (tabular)

Initialize $Q^0(s,a), \forall s, a$

For $t = 1,2, \dots,$

- Take action $a_t = \max_a Q(s_t, a)$
- Observe reward r_t, next state s_{t+1}
- Update

$$\boxed{Q^{t+1}(s_t, a_t) \leftarrow Q^t(s_t, a_t) + \alpha_t \delta_t}$$

where

$$\delta_t = r_t + \gamma \max_{a'} Q^t(s_{t+1}, a') - Q^t(s_t, a_t)$$

Is gradient of $\left(r_t + \gamma \max_{a'} Q^t(s_{t+1}, a') - Q\right)^2$ with respect to Q at $Q^t(s_t, a_t)$

Q-learning function approximation

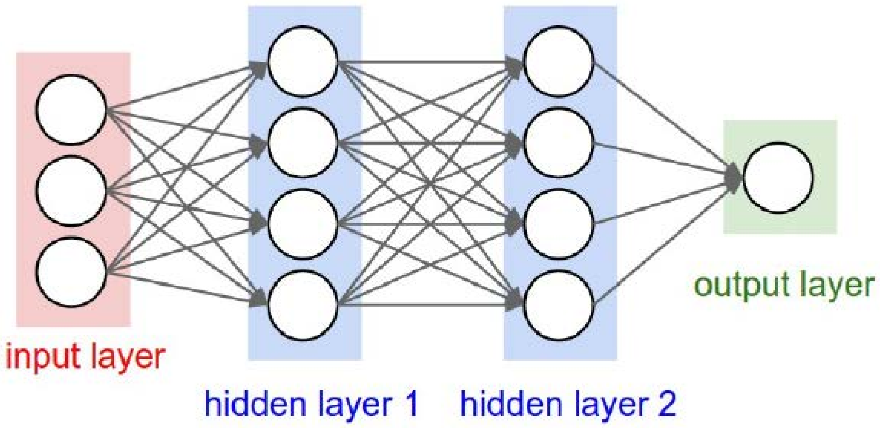
Use parametric function $Q_\theta(s, a)$

- Linear function: feature vector for every s, a: $\mathrm{f}_{s,a} = [f_1, f_2, \ldots, f_d]$

$$Q_\theta(s, a) = f_{s,a}^T \theta$$

- Deep neural network

s, a



$Q_\theta(s, a)$

θ

Q-learning (function approximation)

Initialize ~~$Q^{\theta}(s,a), \forall s, a$~~ θ^0

For $t = 1, 2, \ldots,$

- Take action $a_t = \max_a Q_{\theta^t}(s_t, a)$
- Observe reward r_t, next state s_{t+1}
- Update

$$\boxed{\theta^{t+1} \leftarrow \theta^t + \alpha_t \delta_t}$$

where

$$\delta_t = \left(r_t + \gamma \max_{a'} Q_{\theta^{t+1}}(s_{t+1}, a') - Q_{\theta^t}(s_t, a_t)\right) \nabla_{\theta^t} Q_{\theta^t}(s_t, a_t)$$

Is gradient of $\left(r_t + \gamma \max_{a'} Q^t(s_{t+1}, a') - Q_\theta(s_t, a_t)\right)^2$ with respect to θ at θ^t

Guarantee (tabular)

Theorem 1 (Watkins and Dayan [1992]). *Given bounded rewards* $|r_t| \leq R$, *learning rates* $0 \leq \alpha_t < 1$, *and*

$$\sum_{i=1}^{\infty} \alpha_{n^i(s,a)} = \infty, \sum_{i=1}^{\infty} (\alpha_{n^i(s,a)})^2 < \infty,$$

then $\hat{Q}^t(s,a) \to Q(s,a)$ *as* $t \to \infty$ *for all* s, a *with probability* 1. *Here,* $n^i(s,a)$ *is the index of the* i^{th} *time the action* a *is tried in state* s, *and* $\hat{Q}^t(s,a)$ *is the estimate* $\hat{Q}$ *in round* t.

Reinforcement learning guarantees

- PAC analysis
  - Bound the sample complexity for finding a near-optimal policy.
- Regret analysis
  - Difference in reward obtained by algorithm compared to a benchmark policy, over the steps of the execution of the algorithm.

- Focus on number of samples vs. reward

Optimistic Q-learning (PAC-type bounds)

[Evan-dar and Mansour 2002]

Initialize $Q^0(s,a) = \frac{1}{\prod_t^T (1-\alpha_t)} V_{max}, \forall s, a$

For $t = 1,2, \dots,$

- Take action $a_t = \max_a Q(s_t, a)$
- Observe reward r_t, next state s_{t+1}
- Update

$$Q^{t+1}(s_t, a_t) \leftarrow (1-\alpha_t) Q^t(s_t, a_t) + \alpha_t \left(r_t + \gamma \max_{a'} Q^t(s_{t+1}, a')\right)$$

- Optimism: Guarantees $Q^t(s,a) \geq Q^*(s,a)$, $\forall t \leq T$
- **Theorem**: By setting T large enough (inverse function of ϵ, δ), which ever state action is played infinitely many times is near optimal: has $Q^*(s,a) - V^*(s) \geq \epsilon$

Delayed Q-learning [Strehl et al. 2006]

- Modified version of optimistic Q-learning
- Batch update of Q-values after sufficient number of plays

Theorem 1 *Let M be any MDP and let ϵ and δ be two positive real numbers. If Delayed Q-learning is executed on MDP M, then it will follow an ϵ-optimal policy on all but $O\left(\frac{SA}{(1-\gamma)^8\epsilon^4}\ln\frac{1}{\delta}\ln\frac{1}{\epsilon(1-\gamma)}\ln\frac{SA}{\delta\epsilon(1-\gamma)}\right)$ timesteps, with probability at least $1-\delta$.*

Regret minimization

Goal:

- Minimize regret in time T

$$\mathrm{Reg}(M, T) = T\,\rho^* - \sum_{t=1}^{T} r(s_t, a_t)$$

- ρ^* is infinite horizon average reward (gain), achieved by the best stationary policy π^*

(We care about **performance** during the execution of the algorithm)

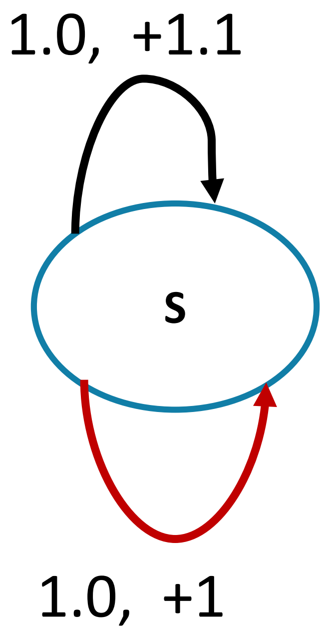
How to learn the response model and **state transition model**, while minimizing "regret"?

The need for exploration

- Single state MDP
  - Solution concept: optimal action
  - **Multi-armed bandit problem**
- Uncertainty in rewards
  - Random rewards with unknown mean μ_1, μ_2
- Exploit only: use the current best estimate (MLE/empirical mean) of unknown mean to pick arms
- Initial few trials can mislead into playing red action forever

1.1, 1, 0.2,

1, 1, 1, 1, 1, 1, 1, ……

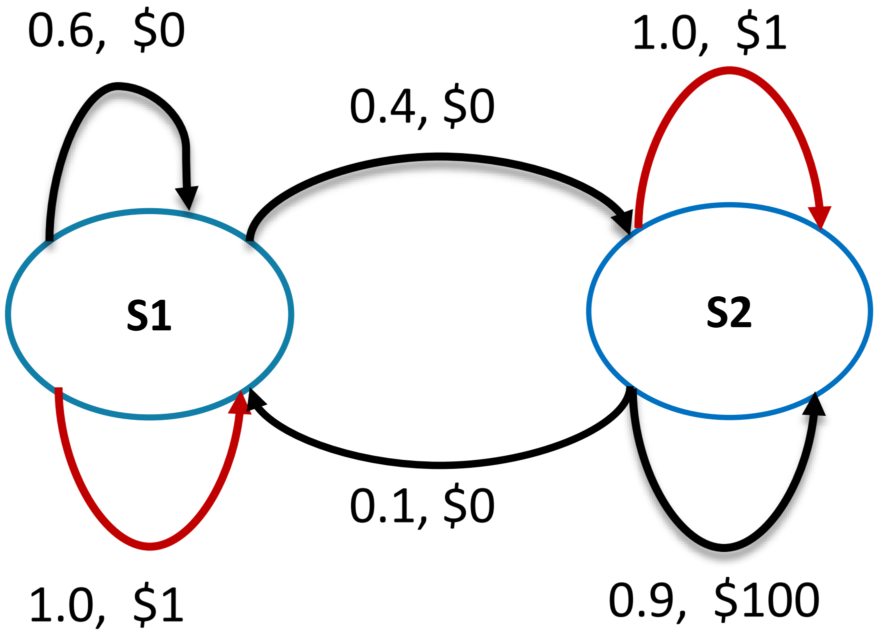


$P(s, a, s),\ r(s, a)$

Exploration-Exploitation tradeoff

- Exploitation: play the empirical mean reward maximizer
- Exploration: play less explored actions to ensure empirical estimates converge

The need for exploration



- Uncertainty in rewards, state transitions
- Unknown reward distribution, transition probabilities
- Exploration-exploitation:
  - Explore actions/states/policies, learn reward distributions and transition model
  - Exploit the (seemingly) best policy

Summary of recent work

Upper confidence bound based algorithms [Jaksch, Ortner, Auer, 2010] [Bartlett, Tewari, 2012]

- Worst-case regret bound $\tilde{O}(DS\sqrt{AT})$ for communicating MDP
- Lower bound $\Omega\left(\sqrt{DSAT}\right)$

Optimistic Posterior Sampling [A. Jia 2017]

- Worst-case regret bound $\tilde{O}(D\sqrt{SAT})$ for communicating MDP of diameter D
- Improvement by a factor of $\sqrt{S}$

Optimistic Value iteration [Azar, Osband, Munos, 2017]

- Worst-case reget bound $\tilde{O}(\sqrt{HSAT})$ in episodic setting

Posterior sampling known prior setting [Osband and Van Roy, 2016, 2017]

- Bayesian regret bound of $\tilde{O}(H\sqrt{SAT})$ in episodic setting, length H episodes

Next…

- ➢ **UCRL: Upper confidence bound based algorithm for RL**
- Posterior sampling based algorithm for RL
  - Main result
  - Proof techniques

UCRL algorithm [Jacksch, Ortner, Auer 2002]

- Similar principles as UCB

This is a ***Model-based approach***

- Maintain an estimate of model $\hat{P}, \hat{R}$
- Occassionally solve the MDP $\left(\mathrm{S}, \mathrm{A}, \hat{P}, \hat{R}, s_1\right)$ to find a policy
- Run this policy for some time to get samples, and update model estimate

Compare to ``model-free" approach or direct learning approach like Q-learning

- Directly update Q-values or value function or policy using samples.

UCRL algorithm

- Proceed in epochs, an epoch ends when the number of visits of *some* state-action pair doubles.

In the beginning of every epoch k

- Use samples to compute an optimistic MDP (S, A, $\tilde{R}, \tilde{P}, s_1$)
  - MDP with value greater than true MDP (Upper bound!!)
- Solve the optimistic MDP to find optimal policy $\tilde{\pi}$

Execute $\tilde{\pi}$ in epoch k

- observe samples s_t, r_t, s_{t+1}

Go to next epoch If visits of *some* state-action pair doubles

- If $n_k(s,a) \geq 2\, n_{k-1}(s,a)$ for some s, a

UCRL algorithm (computing optimistic MDP)

In the beginning of every epoch k

- For every s, a, compute **empirical** model estimate
  - let $\mathrm{n_k}(s, a)$ be the number of times s, a was visited before this epoch,
  - let $\mathrm{n_k}(s, a, s')$ be the number of transition to s'
  - Set $\hat{R}(s, a)$ as average reward over these $\mathrm{n_k}(s, a)$ steps
  - Set $\hat{P}(s, a, s')$ as $\frac{\mathrm{n_k}(s,a,s')}{\mathrm{n_k}(s,a)}$
- Compute **optimistic** model estimate
  - Use Chernoff bounds to define confidence region around $\hat{R}, \hat{P}$
    - $|\hat{P}(s, a, s') - P(s, a, s')| \leq \frac{\log(t)}{\sqrt{n_k(s,a)}}$ with probability $1 - \frac{1}{t^2}$
    - True R, P lies in this region
  - Find the best combination , $\tilde{R}, \tilde{P}$ in this region
    - MDP (S, A, $\tilde{R}, \tilde{P}$, s_1) with maximum value
    - Will have value more than the true MDP

Main result

- Recall regret:

$$\text{Regret}(M, T) = T\,\rho^* - \sum_{t=1}^{T} r(s_t, a_t)$$

- <u>Theorem:</u> For any **communicating** MDP M with (unknown) diameter D, with high probability:

$$\text{Regret}(M, T) \leq \tilde{O}(DS\sqrt{AT})$$

- $\tilde{O}$() notation hides logarithmic factors in S,A,T beyond constants.

Communicating MDPs

Non-episodic setting, no restarts

- Can get stuck on a bad state for a long time

Communicating MDPs:

- There is always a way to get out of a bad state in finite time
- Definition: For every pair of states s, s', there exists a policy π such that using this policy starting from s, expected time to reach s' is finite and bounded by D, called **the diameter** of the MDP

Useful properties of communicating MDPs

- Optimal asymptotic average reward doesn't depend on the starting state.
- Asymptotic average reward (**Gain**) of policy π

$$\rho^{\pi}(s) = E\left[\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r\big(s_t, \pi(s_t)\big) \mid s_1 = s\right]$$

- There exists a single policy π^* such that

$$\max_{\pi} \rho^{\pi}(s) = \rho^{\pi^*}(s), \forall s \quad =: \ \rho^* \quad \textbf{(Optimal gain)}$$

Main result

- Recall regret:

$$\text{Regret}(M,T) = T\,\rho^* - \sum_{t=1}^{T} r(s_t, a_t)$$

- Theorem: For any **communicating** MDP M with (unknown) diameter D, with high probability:

$$\text{Regret}(M,T) \leq \tilde{O}(DS\sqrt{AT})$$

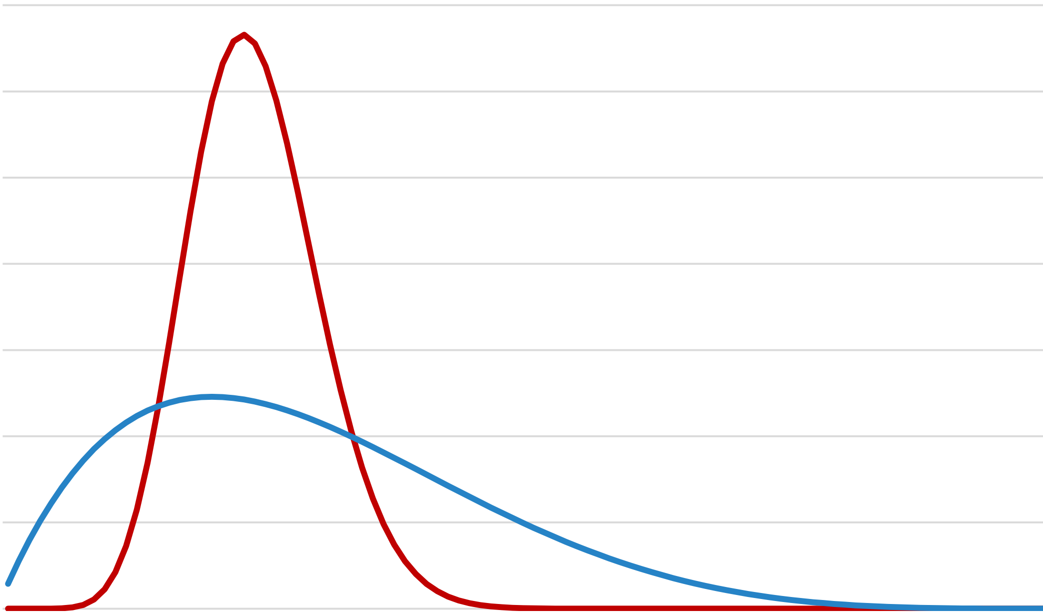
- $\tilde{O}$() notation hides logarithmic factors in S,A,T beyond constants.

Next...

- Our setting, regret definition
- **Posterior sampling algorithm** for MDPs
  - Main result
  - Proof techniques

Posterior Sampling: main idea [Thompson 1933]

- Maintain Bayesian posteriors for unknown parameters
- With more trials posteriors concentrate on the true parameters
  - Mode captures MLE: enables exploitation
- Less trials means more uncertainty in estimates
  - Spread/variance captures uncertainty: enables exploration
- A sample from the posterior is used as an estimate for unknown parameters to make decisions



Posterior Sampling : Bayesian posteriors

- Assume for simplicity: Known reward distribution
- Needs to learn the unknown transition probability vector $P_{s,a} = (P_{s,a}(1), \dots, P_{s,a}(S))$ for all s, a
- In any state $s_t = s, a_t = a$, observes new state s_{t+1}
  - outcome of a Multivariate Bernoulli trial with probability vector $P_{s,a}$

Posterior Sampling with Dirichlet priors

- Given prior $\text{Dirichlet}(\alpha_1, \alpha_2, \ldots, \alpha_S)$ on $P_{s,a}$
- After a Multinoulli trial with outcome (new state) i, Bayesian posterior on $P_{s,a}$

$$\text{Dirichlet}(\alpha_1, \alpha_2, \ldots, \alpha_i + 1, \ldots, \alpha_S)$$

- After $n_{s,a} = \alpha_1 + \cdots + \alpha_S$ observations for a state-action pair s, a
  - Posterior mean vector is empirical mean

$$\hat{P}_{s,a}(i) = \frac{\alpha_i}{\alpha_1 + \cdots + \alpha_S} = \frac{\alpha_i}{n_{s,a}}$$

  - variance bounded by $\frac{1}{n_{s,a}}$
- With more trials of s, a, the posterior mean concentrates around true mean

Posterior Sampling for RL (Thompson Sampling)

Learning

- Maintain a Dirichlet posterior for $P_{s,a}$ for every s, a
  - After round t, on observing outcome s_{t+1}, update for state s_t and action a_t

To decide action

- Sample a $\tilde{P}_{s,a}$ for every s, a
- Compute the optimal policy $\tilde{\boldsymbol{\pi}}$ for sample MDP $(S, A, \tilde{P}, r, s_0)$
- Choose $a_t = \tilde{\pi}(s_t)$

Exploration-exploitation

- Exploitation: With more observations Dirichlet posterior concentrates, $\tilde{P}_{s,a}$ approaches empirical mean $\hat{P}_{s,a}$
- Exploration: Anti-concentration of Dirichlet ensures exploration for states/actions/policies less explored

Optimistic Posterior Sampling [A., Jia, NIPS 2017]

- Proceed in epochs, an epoch ends when the number of visits $N_{s,a}$ of any state-action pair doubles.

In every epoch

- For every s, a, generate **multiple** $\psi = \tilde{O}(S)$ independent samples from a Dirichlet posterior for $\mathrm{P}_{s,a}$
- Form **extended** sample MDP $(S, \psi A,\ \tilde{P},\ r, s_0)$
- Find optimal policy $\tilde{\pi}$ and use through the epoch

Further, initial exploration:

- For s, a with very small $N_{s,a} < \sqrt{\frac{TS}{A}}$, use a simple optimistic sampling, that provides extra exploration

Main result [A., Jia NIPS 2017]

- An algorithm based on posterior sampling with high probability near-optimal worst-case regret upper bound
- Recall regret:

$$\text{Regret}(M, T) = T\,\rho^* - \sum_{t=1}^{T} r(s_t, a_t)$$

- Theorem: For any **communicating** MDP M with (unknown) diameter D, and for $\mathrm{T} \geq S^5 A$, with high probability:

$$\text{Regret}(M, T) \leq \tilde{O}(D\sqrt{SAT})$$

- Improvement of $\sqrt{S}$ factor above UCB based algorithm

Next…

- UCRL
- Posterior sampling algorithm for MDPs
- **Proof techniques**

Regret Analysis (outline)

- Average regret in an epoch k

$$\rho^* - \frac{1}{T_k}\sum_{t\in T_k} r_{s_t,a_t} = (\rho^* - \tilde{\rho}) + \left(\tilde{\rho} - \frac{1}{T_k}\sum_{t\in T_k} r_{s_t,a_t}\right)$$

  - Optimal Gain (Asymptotic average reward) for true MDP M (policy π^*)
  - Optimal Gain for sample extended MDP (policy $\tilde{\pi}_k$)
- **First term:** we show optimism $\boxed{\tilde{\rho} \geq \rho^*}$
  - Immediate for UCB
  - Needs to use spread (anti-concentration) of posterior in Thompson Sampling

Regret Analysis (main insights)

- Average regret in an epoch k

$$\rho^* - \frac{1}{T_k}\sum_{t\in T_k} r_{s_t,a_t} = \overbrace{(\rho^* - \tilde{\rho})}^{\leq 0} + \boxed{\left(\tilde{\rho} - \frac{1}{T_k}\sum_{t\in T_k} r_{s_t,a_t}\right)}$$

  - Optimal Gain for true MDP M (policy π^*)
  - Optimal Gain for sample extended MDP (policy $\tilde{\pi}_k$)

- **Second term**
  - Same policy but different MDP
  - $\tilde{\rho}$: follows estimated/sampled transition probability vector
  - $\frac{1}{T_k}\sum_{t\in T_k} r_{s_t,a_t}$: follows true transition probability vector
  - Bounded using **concentration** of estimated/sampled transition probability vector