PAC Model-Free Reinforcement Learning

Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, Michael L. Littman

RL^3, Rutgers University
CSE, Univ. of California, San Diego
TTI Chicago → Yahoo! Research

Presenter: Lihong Li

With thanks to:
Sham Kakade, Yishay Mansour, Ali Nouri, Satinder Singh, and Tom Walsh.

WARNING: This is a theoretical work about complexity results.

> "Someone told me that each equation I included in the book would halve the sales. I therefore resolved not to have any equations at all."

— Stephen Hawking (A Brief History of Time, 1988)

BUT we are computer scientists.

SO I'm going to use three equations.

Conclusions

Consider reinforcement learning

- of a single agent
- in a fully observable environment
- based on a single thread of experience (no resets or generative models)

Theoretical contributions: *Delayed Q-learning* which

- is model-free
- improves on previous complexity results
  - space complexity
  - per-step computational complexity
  - **sample complexity** (of exploration)
- answers the open question of efficient model-free RL affirmatively

Talk Outline

- Introduction
- Delayed Q-learning
- Proof Sketch
- Future Directions

▷ Introduction
Delayed Q-learning
Main Results
Conclusions

Introduction

Notation

Consider finite Markov decision processes (MDPs) with

- state space S,
- action space A,
- discount factor $\gamma \in [0, 1)$,
- transition function $T(s'|sa)$, and
- bounded rewards $R(s, a) \in [0, 1]$.

A deterministic Markov policy $\pi : S \mapsto A$.

Given a trajectory: $s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_t, a_t, r_t, \cdots$.

Value functions:

$$
\begin{aligned}
V^\pi(s) &:= \mathbb{E}\{r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots \mid s_1 = s, \pi\} \\
Q^\pi(s, a) &:= \mathbb{E}\{r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots \mid s_1 = s, a_1 = a, \pi\} \\
V^*(s) &:= V^{\pi^*}(s) = \max_\pi V^\pi(s) \\
Q^*(s, a) &:= Q^{\pi^*}(s, a) = \max_\pi Q^\pi(s, a)
\end{aligned}
$$

Objective

- to learn the optimal policy or value function
- based on sampling of (or interaction with) the environment
- without knowing T and R.

Challenges:

- **exploration vs. exploitation**
- temporal credit assignment
- scaling up
- generalization

Performance Criteria

We often trade one factor for another:

- per-step computational complexity
- space complexity
  - model-free: $o(S^2 A)$
  - model-based: $\Omega(S^2 A)$
- sample complexity
  - (Kakade, 2003): $\#$timesteps that the algorithm does *not* behave ϵ-optimally.
  - An algorithm is PAC-MDP if w.h.p. its sample complexity is bounded by a polynomial in relevant quantities.

Summary

	PAC-MDP	non-PAC-MDP/unknown
model-free		Q-learning, Sarsa
model-based	E^3, Rmax, MBIE	Dyna-Q, prioritized sweeping, certainty equivalence, adaptive RTDP

	computation	space	(best) sample
E^3	$\Omega(S^2A)$	$\Theta(S^2A)$	polynomial
Rmax	$\Omega(S^2A)$	$\Theta(S^2A)$	$\tilde{O}\left(\frac{S^2A}{\epsilon^3(1-\gamma)^6}\right)$
MBIE	$\Omega(S^2A)$	$\Theta(S^2A)$	$\tilde{O}\left(\frac{S^2A}{\epsilon^3(1-\gamma)^6}\right)$
Q-learning	$O(\log(A))$	$\Theta(SA)$	can be EXP
Sarsa	$O(\log(A))$	$\Theta(SA)$	can be EXP

Summary

	PAC-MDP	non-PAC-MDP/unknown
model-free	**Delayed Q-learning**	Q-learning, Sarsa
model-based	E^3, Rmax, MBIE	Dyna-Q, prioritized sweeping, certainty equivalence, adaptive RTDP

	computation	space	(best) sample
E^3	$\Omega(S^2A)$	$\Theta(S^2A)$	polynomial
Rmax	$\Omega(S^2A)$	$\Theta(S^2A)$	$\tilde{O}\left(\frac{S^2A}{\epsilon^3(1-\gamma)^6}\right)$
MBIE	$\Omega(S^2A)$	$\Theta(S^2A)$	$\tilde{O}\left(\frac{S^2A}{\epsilon^3(1-\gamma)^6}\right)$
Q-learning	$O(\log(A))$	$\Theta(SA)$	can be EXP
Sarsa	$O(\log(A))$	$\Theta(SA)$	can be EXP
Delayed Q-learning	$\mathbf{O}(\log(\mathbf{A}))$	$\boldsymbol{\Theta}(\mathbf{SA})$	$\tilde{\mathbf{O}}\left(\frac{\mathbf{SA}}{\epsilon^4(\mathbf{1}-\gamma)^8}\right)$

Introduction

▷ Delayed Q-learning

Main Results

Conclusions

Delayed Q-learning

Algorithm Overview

During execution

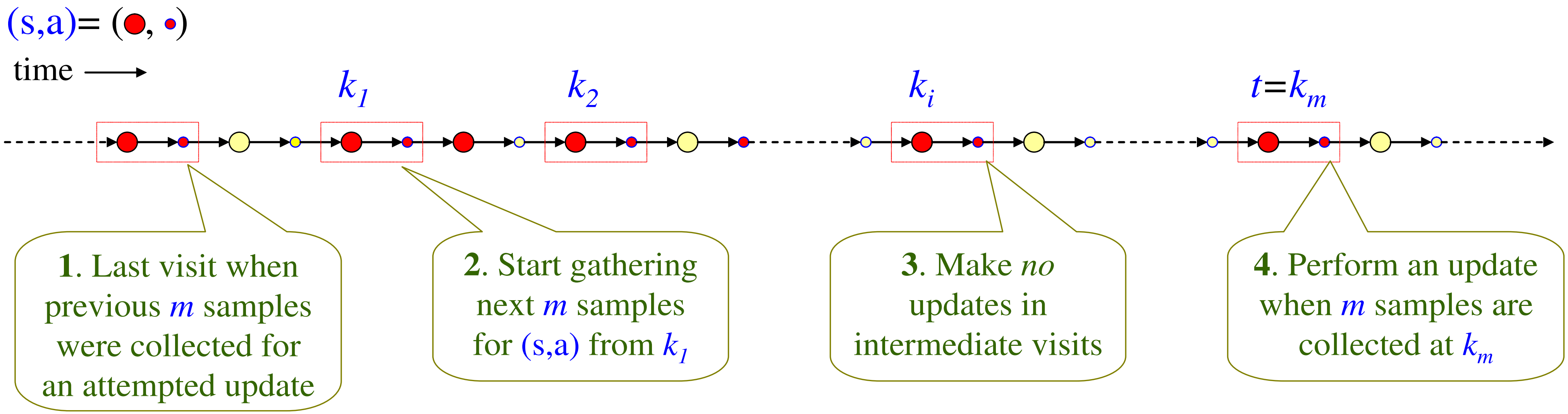
- ☐ Maintain Q-values for all (s, a), denoted by $Q_t(s, a)$ at time t;
- ☐ Define $V_t(s) = \max_a Q_t(s, a)$.

Delayed Q-learning:

1. Start state: s_1.
2. Optimistic initialization: $Q_1(s, a) \leftarrow Q_{\max} \, (= \frac{1}{1-\gamma})$.
3. At time $t = 1, 2, 3, \cdots$:
   (a) selects greedy action: $a_t \leftarrow \arg\max_a Q_t(s_t, a)$;
   (b) observes immediate reward r_t and next state s_{t+1};
   (c) one-step lookahead backup value: $r_t + \gamma \max_a Q_t(s_{t+1}, a)$;
   (d) **updates $\mathbf{Q_t(s_t, a_t)}$:**

“Raw” Update Rule

Suppose (s, a) is visited m times since last update:



The respective backup values:

$$r_{k_1} + \gamma V_{k_1}(s_{k_1+1}), \quad r_{k_2} + \gamma V_{k_2}(s_{k_2+1}), \quad \cdots, \quad r_{k_m} + \gamma V_{k_m}(s_{k_m+1})$$

Q-learning at time k_i:

$$Q_{k_i+1}(s, a) \quad \leftarrow \quad (1 - \alpha) Q_{k_i}(s, a) + \alpha \left(r_{k_i} + \gamma V_{k_i}(s_{k_i+1}) \right).$$

The *delayed* update rule at time k_m:

$$Q_{t+1}(s, a) \quad \leftarrow \quad \frac{1}{m} \sum^{m} \left(r_{k_i} + \gamma V_{k_i}(s_{k_i+1}) \right).$$

"Refined" Update Rule

The "raw" update rule: $Q_{t+1}(s,a) \leftarrow \frac{1}{m}\sum_{i=1}^{m}\big(r_{k_i} + \gamma V_{k_i}(s_{k_i+1})\big)$.

To prove PAC-MDP-ness, make several changes:

- Add a bonus $\epsilon_1 = \Theta(\frac{\epsilon}{1-\gamma})$:

$$Q_{t+1}(s,a) \quad \leftarrow \quad \frac{1}{m}\sum_{i=1}^{m}\big(r_{k_i} + \gamma V_{k_i}(s_{k_i+1})\big) + \boldsymbol{\epsilon_1}.$$

- Update of $Q(s,a)$ succeeds only when
  - it results in a minimum decrease of ϵ_1, and
  - some $Q(\cdot,\cdot)$ is changed since last update of $Q(s,a)$.
- If update unsuccessful
  - keep current Q-values,
  - **discard these $\mathbf{m}$ samples**, and
  - **start collecting another $\mathbf{m}$ samples.**

Comparison to Q-learning

Similarities:

- model-free, learns Q-values, algorithmic structure, online, etc.

Differences:

- optimistic initialization
- updates
  - delayed until m samples
  - no learning rates
  - may fail
  - finite $\#$updates
  - Q-values monotonically decrease
- always chooses greedy actions
- never has exponential sample complexity

Introduction
Delayed Q-learning
▷ Main Results
Conclusions

Main Results

Main Results

Set

$$m = \Theta\left(\frac{\log\left(\frac{SA}{\epsilon\delta(1-\gamma)}\right)}{\epsilon^2(1-\gamma)^4}\right).$$

Then Delayed Q-learning enjoys provable efficiency:

- Per-step computational complexity: $O(\log(A))$
- Space complexity: $O(SA)$
- Sample complexity: $\tilde{O}(SA)$

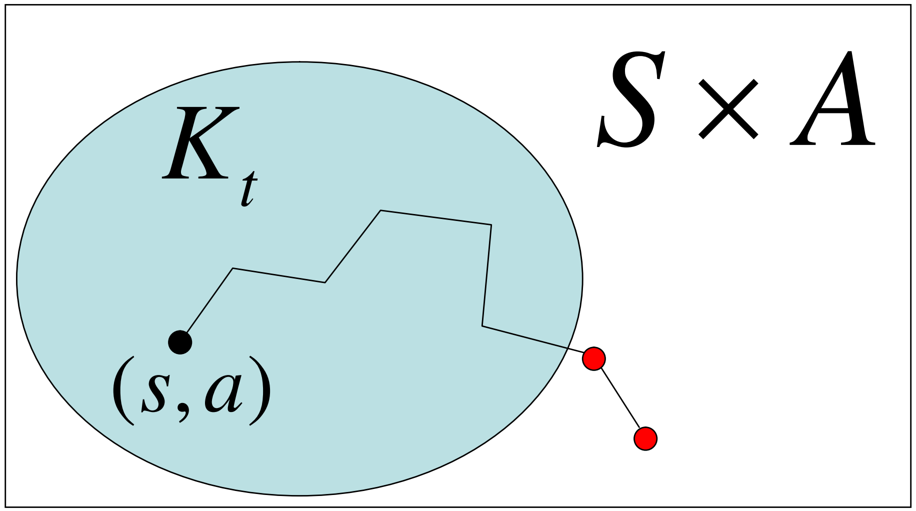
Similar results for finite-horizon cases.

"Known State-Actions":

$$K_t = \left\{ (s,a) \middle| Q_t(s,a) - \left(R(s,a) + \gamma \sum_{s'} T(s'|sa) V_t(s') \right) \leq 3\epsilon_1 \right\}$$

Escape probability:

$$p = \Pr\left\{ \text{escape } K_t \text{ in } H = O\left(\frac{1}{1-\gamma} \log\left(\frac{1}{\epsilon(1-\gamma)} \right) \right) \text{ steps} \right\}.$$



Proof Sketch

1. Bound # updates of Q-values by a polynomial P
   - because of the refined update rule
   - allows Hoeffding's bound be used in our proof below
2. Case 1 ("p small enough"): **near-optimal**
   - p small $\implies$ Bellman residuals small w.h.p.
   - $\implies$ actual value functions are close to V^*
   - $\implies$ near-optimal policies
3. Case 2 ("p not small enough"): **except polynomial #steps**
   - $(s, a) \notin K_{k_1}$ and is visited m times
     $\implies$ $Q(s, a)$ is updated at time k_m w.h.p.
   - but #updates is bounded by P
   - $\implies$ bound #occurrences of this "undesired" case

Conclusion: Delayed Q-learning is PAC-MDP.

Related Work on Sample Complexity

- Model-based
  - **(Fiechter, COLT'94)**: assumes a reset
  - **E^3 (Kearns-Singh ICML'98)**: explicitly explores or exploits
  - **Rmax (Brafman-Tennenholtz IJCAI'01) / MBIE (Strehl-Littman ICML'05)**: optimism in the face of uncertainty
  - **RTDP-RMAX and RTDP-IE (Strehl-Li-Littman UAI'06)**: $O(S \log A)$ computational complexity
- Model-free
  - **Phased Q-learning (Kearns-Singh NIPS'99)**: averaging updates to simulate Bellman backups
  - **(Even-dar-Mansour JMLR'03)**: assumes an efficient exploration policy

Introduction
Delayed Q-learning
Main Results
▷ Conclusions

Conclusions

Future Directions

- Closing the gap between upper and lower bounds of sample complexity.
  - best known lower bound (Kakade 2003): $\tilde{\Omega}(\frac{SA}{\epsilon(1-\gamma)^2})$
- Extending results to possibly infinite MDPs
  - generalization
- Employing structures
  - factored representations (e.g., factored E^3)
  - state abstraction

Take-Home Messages

- Solved the open question of efficient model-free RL
- Delayed Q-learning: the first algorithm that is
  - model free
  - proven to be efficient, and
  - without resets or generative models
- Sample complexity ($\tilde{O}(SA)$) is less than MDP description complexity ($O(S^2A)$)
  - only $O(SA)$ quantities are to be estimated;
  - MDP representations are not *compact* in the sense of efficiently learning near-optimal behavior.
- Sample complexity does not increase significantly compared to deterministic MDPs ($O(SA)$).

?