Reinforcement Learning

Lecture 3: RL problems, sample complexity and regret

Alexandre Proutiere, Sadegh Talebi, Jungseul Ok

KTH, The Royal Institute of Technology
Objectives of this lecture

- Introduce the different classes of RL problems
- Introduce the notion of on and off-policy algorithm
- Introduce regret and sample complexity, the two main performance metrics for RL algorithms
- Provide performance limits and state-of-the-art algorithms
Lecture 3: Outline

1. Different classes of RL problems
   a. Episodic problems
   b. Discounted problems
   c. Ergodic problems
2. On vs Off policies
3. Sample complexity and regret
4. Fundamental performance limits and state-of-the-art algorithms
1. **Different classes of RL problems**
   a. Episodic problems
   b. Discounted problems
   c. Ergodic problems
2. On vs Off policies
3. Sample complexity and regret
4. Fundamental performance limits and state-of-the-art algorithms
Initial state $s_0$ (could be a r.v.)
Transition probabilities at time $t$: $p_t(s'|s, a)$
Reward at time $t$: $r_t(a, s)$ and at time $H$: $r_H(s)$
Unknown transition probabilities and reward function
Objective: quickly learn a policy $\pi^*$ maximising over $\pi_0 \in MD$

$$V_{H}^{\pi_0} := \mathbb{E} \left[ \sum_{u=0}^{H-1} r_u(s_u^{\pi_0}, a_u^{\pi_0}) + r_H(s_H^{\pi_0}) \right]$$

from either existing data (off-policy) or gathered data along the way (on-policy)
Finite-horizon MDP to episodic RL problems

- Data: \( K \) episodes of length \( H \) (actions, states, rewards)
- Learning algorithm \( \pi : \text{data} \mapsto \pi_K \in MD \)
- Performance of \( \pi \): how close \( \pi_K \) is from the optimal policy \( \pi^* \)
Infinite-horizon discounted MDP to discounted RL problems

- Stationary transition probabilities $p(s'|s, a)$ and rewards $r(a, s)$, uniformly bounded: $\forall a, s, |r(s, a)| \leq 1$

- Objective: for a given discount factor $\lambda \in [0, 1)$, from the available data, find a policy $\pi \in \mathcal{MD}$ maximising (over all possible policies)

$$V^\pi(s_0) := \lim_{T \to \infty} \mathbb{E}_{s_0} \left[ \sum_{u=0}^{T} \lambda^u r(s_u^\pi, a_u^\pi) \right]$$
• Stationary transition probabilities $p(s'|s, a)$ and rewards $r(s, a)$, uniformly bounded: $\forall a, s, |r(s, a)| \leq 1$

• Objective: learn from data a policy $\pi \in MD$ maximising (over all possible policies)

$$g^\pi = V^\pi(s_0) := \lim_{T \to \infty} \inf \frac{1}{T} \mathbb{E}_{s_0} \left[ \sum_{u=0}^{T-1} r(s_u^\pi, a_u^\pi) \right]$$
Lecture 3: Outline

1. Different classes of RL problems
   a. Episodic problems
   b. Discounted problems
   c. Ergodic problems

2. On vs Off policies

3. Sample complexity and regret

4. Fundamental performance limits and state-of-the-art algorithms
On vs. Off policies

**Data:** the observations made by the agent along the trajectory of the dynamical system.

An **on-policy** learns the value of the policy being carried out by the agent including the exploration steps. The policy used by the agent is computed from the previous collected data. It is an **active learning** method as the gathered data is controlled.

\[
s_{t+1} \sim p(\cdot | s_t, \pi_t(s_t))
\]

\[
\pi_t = F_t(s_0, a_0, \ldots, s_{t-1}, a_{t-1}, s_t)
\]
An off-policy learner learns the value of the optimal policy independently of the agent’s actions. The policy used by the agent is often referred to as the behaviour policy, and denoted by $\pi_b$.

**Examples:** Q-learning is an off-policy, SARSA in an on-policy.
Lecture 3: Outline

1. Different classes of RL problems
   a. Episodic problems
   b. Discounted problems
   c. Ergodic problems

2. On vs Off policies

3. Sample complexity and regret

4. Fundamental performance limits and state-of-the-art algorithms
Sample complexity

How can we measure the performance of various learning algorithms?

**Sample complexity.** Defined as the time required to find an approximately optimal policy. Well defined for any kind of RL problems. A Probably Approximately Correct (PAC) framework.

1. **Episodic RL.** An on-policy algorithm $\pi$ returns, after the end of the $(k - 1)$-th episode, a policy $\pi_k$ to be applied in the $k$-th episode.

   The sample complexity of $\pi$ is the minimum number $SP^\pi$ of episodes such that for all $k \geq SP^\pi$, $\pi_k$ is $\epsilon$-optimal with probability at least $1 - \delta$, i.e., for $k \geq SP^\pi$,

   $$\mathbb{P} \left[ V^\pi_k \geq V^\pi_* - \epsilon \right] \geq 1 - \delta$$
Sample complexity. Defined as the time required to find an approximately optimal policy. Well defined for any kind of RL problems.

2. Discounted and ergodic RL. An on-policy algorithm \( \pi \) returns, after the end of the \( (t - 1) \)-th step, a policy \( \pi_t \) to be applied in the \( t \)-th step.

The sample complexity of \( \pi \) is the minimum number \( SP^\pi \) of steps such that for any \( t \geq SP^\pi \), \( \pi_t \) is \( \epsilon \)-optimal when starting in the current state \( s^\pi_t \) with probability at least \( 1 - \delta \), i.e., for any \( t \geq SP^\pi \),

\[
\mathbb{P} \left[ V^{\pi_t} (s^\pi_t) \geq V^{\pi^*} (s^\pi_t) - \epsilon \right] \geq 1 - \delta
\]

Sample complexity can be defined for off-policies analogously.
A more appropriate performance metrics for on-policies? Capture the exploration vs. exploitation trade-off.

**Regret of an algorithm** $\pi$. Defined as the difference between the cumulative reward of the optimal policy and that gathered by $\pi$. 

![Regret Diagram](image)
Regret of an algorithm $\pi$. Defined as the difference between the cumulative reward of the optimal policy and that gathered by $\pi$.

1. Episodic RL. An on-policy algorithm $\pi$ returns, after the end of the $(k - 1)$-th episode, a policy $\pi_k$ to be applied in the $k$-th episode.

The regret of $\pi$ after $K = T/H$ episodes is:

$$R^\pi(T) = KV^\pi_H^* - \sum_{k=1}^{K} \mathbb{E}[V^\pi_H^k]$$
Regret of an algorithm $\pi$. Defined as the difference between the cumulative reward of the optimal policy and that gathered by $\pi$.

2. Discounted RL. An on-policy algorithm $\pi$ returns, after the end of the $(t - 1)$-th step, a policy $\pi_t$ to be applied in the $t$-th step. The regret is difficult to define as the cumulative reward is bounded (no scaling in $T$)

A tentative definition. The regret of $\pi$ after $T$ steps is:

$$R^\pi(T) = \sum_{t=1}^{T} \left( V^{\pi^*}(s^*_t) - V^\pi_t(s^\pi_t) \right)$$
Regret of an algorithm $\pi$. Defined as the difference between the cumulative reward of the optimal policy and that gathered by $\pi$.

3. Ergodic RL. An on-policy algorithm $\pi$ returns, after the end of the $(t - 1)$-th step, a policy $\pi_t$ to be applied in the $t$-th step.

The regret of $\pi$ after $T$ steps is:

$$ R^\pi(T) = Tg^\pi - \sum_{t=1}^{T} \mathbb{E}[r(s^\pi_t, a^\pi_t)] $$
A remark on these definitions

For discounted and ergodic RL problems, along which trajectory should we define the performance metrics?

Sample complexity:

\[ \mathbb{P} \left[ V_{\pi t}^t (s_t^\pi) \geq V_{\pi}^* (s_t^\pi) - \epsilon \right] \geq 1 - \delta \]

Following the trajectory of the algorithm can be misleading: Due to exploration, we can end up in states with very low rewards, and being optimal from there does not mean much – especially for discounted problems.
Lecture 3: Outline

1. Different classes of RL problems
   a. Episodic problems
   b. Discounted problems
   c. Ergodic problems
2. On vs Off policies
3. Sample complexity and regret
4. Fundamental performance limits and state-of-the-art algorithms
Fundamental performance limits

- Performance metrics:
  - Sample complexity: amount of data required to learn a near-optimal policy (for off and on-line policies)
  - Regret: cumulative rewards loss due to the need of learning, quantifies the exploration-exploitation trade-off (for on-policies)

- Fundamental performance limits: e.g. regret

Denote by $M = (S, A, (p(s' | s, a), r(s, a))_{s \in S, a \in A})$ a generic MDP.

Two types of regret lower bounds:

- **Problem-specific** lower bound: for all $M$, $\forall$ algorithm $\pi$, $R^\pi(T) \geq F(M, T)$
  
  (most often in the form of $\liminf_{T \to \infty} \frac{R^\pi(T)}{f(T)} \geq c(M)$)

- **Minimax lower** bound: $\exists M$ such that $\forall$ algorithm $\pi$, $\forall T$, $R^\pi(T) \geq G(S, A, T, ...)$
1. Episodic RL problems: State-of-the-art

- **Regret minimisation**
  - Minimax lower bound $\Omega(\sqrt{HSA})$
    No problem-dependent lower bound is derived so far
  - Algorithms: PSRL (Osband et al., 2013), UCBVI (Gheshlaghi Azar et al., 2017), UBEV (Dann et al., 2017)

- **Sample complexity**
  - Minimax lower bound $\Omega\left(\frac{H^2SA}{\varepsilon^2} \log \delta^{-1}\right)$
    No problem-dependent lower bound is derived so far
  - Algorithms: UCFH (Dann & Brunskill, 2015), UBEV (Dann et al., 2017)

Note: UBEV simultaneously satisfies performance guarantees on both regret and sample complexity.
1. Episodic RL problems: State-of-the-art

Regret

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Setup</th>
<th>Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSRL</td>
<td>–</td>
<td>$\widetilde{O}(H S \sqrt{AT})$ – Bayesian regret</td>
</tr>
<tr>
<td>UCBVI-CH</td>
<td>known rewards</td>
<td>$\widetilde{O}(H \sqrt{SAT})$, $T \geq HS^3 A$, $SA \geq H$</td>
</tr>
<tr>
<td>UCBVI-BF</td>
<td>known rewards</td>
<td>$\widetilde{O}(\sqrt{HSAT})$, $T \geq H^3 S^3 A$, $SA \geq H$</td>
</tr>
<tr>
<td>UBEV</td>
<td>–</td>
<td>$\widetilde{O}(H^2 \sqrt{SAT})$, $T \geq S^5 A^3$</td>
</tr>
<tr>
<td>Lower Bound</td>
<td>known rewards</td>
<td>$\Omega(\sqrt{HSAT})$</td>
</tr>
</tbody>
</table>

* $\widetilde{O}(\cdot)$ hides poly-logarithmic (in $S, A, H, T$) terms

Sample complexity

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Setup</th>
<th>Sample Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCFH</td>
<td>known rewards</td>
<td>$O\left(\frac{H^2 CSA}{\varepsilon^2} \log \delta^{-1}\right)^*$</td>
</tr>
<tr>
<td>UBEV</td>
<td>–</td>
<td>$O\left(\frac{H^4 S^2 A}{\varepsilon^2} \text{polylog} \delta^{-1}\right)$</td>
</tr>
<tr>
<td>Lower Bound</td>
<td>–</td>
<td>$\widetilde{\Omega}\left(\frac{H^2 SA}{\varepsilon^2} \log \delta^{-1}\right)$</td>
</tr>
</tbody>
</table>

* $C$ is an upper bound on the number of reachable states under any pair $(s, a)$
UCBVI is an extension of Value Iteration, guaranteeing that the resultant value function is a (high-probability) upper confidence bound (UCB) on the optimal value function.

At the beginning of episode $k$, it computes state-action values using empirical transition kernel and reward function. In step $h$ of backward induction (to update $Q_{k,h}(s,a)$ for any $(s,a)$), it adds a bonus $b_{k,h}(s,a)$ to the value, and ensures that $Q_{k,h}$ never exceeds $Q_{k,h-1}$.

Two variants of UCBVI, depending on the choice of bonus $b_{k,h}$:

- UCBVI-CH
- UCBVI-FB
**UCBVI-CH**: defines the bonus for state-action \((s, a)\), and step \(h\) at episode \(k\) as

\[
b_{k,h}(s, a) = \frac{7H}{\sqrt{N_k(s, a)}} \log(5SAH/\delta)
\]

Here \(N_k(s, a)\) is the number of visits to \((s, a)\) prior to episode \(k\).

Remark: the definition of \(b_{k,h}(s, a)\) is inspired by Hoeffding concentration inequality.
**UCBVI-BF**: defines the bonus for state-action \((s, a)\), and step \(h\) at episode \(k\) as

\[
b_{k,h}(s, a) = \sqrt{\frac{8L}{N_k(s, a)} \text{Var} \hat{p}_k(\cdot|s,a)(V_{k,h+1}(Y)) + \frac{14HL}{3N_k(s, a)}} + \sqrt{\frac{8}{N_k(s, a)} \sum_y \hat{p}_k(y|s,a) \min \left\{ \frac{10^4H^3S^2AL^2}{N^{'k,h+1}(y)}, H^2 \right\}}
\]

where \(L = \log(5SAH/\delta)\), and where \(N^{'k,h}(s)\) is the number of times state \(s\) has been visited in the \(h\)-th step of the first \((k - 1)\) episodes.

Remark: the definition of \(b_{k,h}\) is inspired by Bernstein inequality and concentration of empirical variances.
1. Episodic RL: UCBVI

Algorithm 1 UCBVI

Input: Set confidence parameter $\delta \in (0, 1]$

for episode $k \geq 1$ do
    $Q_{k,H+1}(s,a) = 0$ for all $s,a$
    for step $h = H, H-1, \ldots, 1$ do
        for $(s,a) \in S \times A$ do
            if $(s,a)$ is sampled so far then
                $Q_{k,h}(s,a) = \min\left(Q_{k-1,h}(s,a), H, r(s,a) + \sum_y \hat{p}_k(y|s,a)V_{k,h+1}(s) + b_{k,h}(s,a)\right)$
            end if
        end for
    end for
    for step $h = 1, \ldots, H$ do
        Take action $a_{k,h} = \arg\max_a Q_{k,h}(s_{k,h},a)$
    end for
end for
No regret analysis for this class of RL problems. Two definitions for sample complexity:

- **Sample complexity**: The sample complexity of algorithm $\pi$ is the minimum number $SP^\pi$ of steps such that for any $t \geq SP^\pi$, $\pi_t$ is $\varepsilon$-optimal when starting in the current state $s^\pi_t$ with probability at least $1 - \delta$:

\[
P\left[V^\pi_t(s^\pi_t) \geq V^{\pi^*}(s^\pi_t) - \varepsilon\right] \geq 1 - \delta
\]

- **Q-sample complexity**: The sample complexity of algorithm $\pi$ is the minimum number $QSP^\pi$ of steps such that for any $t \geq QSP^\pi$, the current state-action values $Q^\pi_t$ maintained by $\pi$ is $\varepsilon$-close to $Q^*$ with probability at least $1 - \delta$, and for all state-action pairs, i.e.,

\[
P\left[\|Q^\pi_t - Q^*\|_\infty \leq \varepsilon\right] \geq 1 - \delta
\]

Remark: the latter definition is stronger and implies the former
Lower bound for sample complexity:

- Minimax lower bound: $\Omega \left( \frac{SA}{(1-\lambda)^3 \varepsilon^2} \log \delta^{-1} \right)$

Lower bound for Q-sample complexity:

- Minimax lower bound: $\Omega \left( \frac{SA}{(1-\lambda)^3 \varepsilon^2} \log \delta^{-1} \right)$

No problem-dependent lower bound for any of aforementioned notions is derived so far.
Two classes of algorithms:

**Model-based algorithms**

- Maintain an approximate MDP model by estimating transition probabilities and reward function, and derive a value function from the approximate MDP. The policy is then derived from the value function.
- E3 (Kearns & Singh, 2002), R-max (Brafman & Tennenholtz, 2002), MoRmax (Szita & Szepesvári, 2010), UCRL-\(\gamma\) (Lattimore & Hutter, 2012)

**Model-free algorithms**

- Directly learn a value (or state-value) function, which results in a policy.
- Delayed Q-Learning (Strehl et al. 2006), Median-PAC (Pazis et al., 2016)
2. Discounted RL problems: State-of-the-art

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Setup</th>
<th>Sample Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-max</td>
<td>–</td>
<td>$\tilde{O}\left(\frac{S^2A}{(1-\lambda)^6\varepsilon^3}\log\delta^{-1}\right)$</td>
</tr>
<tr>
<td>MBIE</td>
<td>–</td>
<td>$\tilde{O}\left(\frac{S^2A}{(1-\lambda)^6\varepsilon^3}\log\delta^{-1}\right)$</td>
</tr>
<tr>
<td>Delayed Q-Learning</td>
<td>known reward</td>
<td>$\tilde{O}\left(\frac{SA}{(1-\lambda)^8\varepsilon^4}\log\delta^{-1}\right)$</td>
</tr>
<tr>
<td>MoRmax</td>
<td>–</td>
<td>$\tilde{O}\left(\frac{SA}{(1-\lambda)^6\varepsilon^2}\log\delta^{-1}\right)$</td>
</tr>
<tr>
<td>UCRL-$\gamma$</td>
<td>$</td>
<td>\text{supp}(p(\cdot</td>
</tr>
<tr>
<td>Lower Bound</td>
<td>–</td>
<td>$\tilde{\Omega}\left(\frac{SA}{(1-\lambda)^3\varepsilon^2}\log\delta^{-1}\right)$</td>
</tr>
</tbody>
</table>

$^*$ \(N\) denotes the number of non-zero transitions in the true MDP

$\tilde{O}(\cdot)$ hides poly-logarithmic (in \(S, A, \varepsilon\)) terms
2. Discounted RL problems: State-of-the-art

Q-sample complexity for variants of synchronous Q-learning:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Q-Sample Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q-Learning*</td>
<td>$\tilde{O} \left( \frac{SA}{(1-\lambda)^5 \varepsilon^{5/2}} \text{polylog}\delta^{-1} \right)$</td>
</tr>
<tr>
<td>Model-free QVI</td>
<td>$\tilde{O} \left( \frac{SA}{(1-\lambda)^5 \varepsilon^2} \text{polylog}\delta^{-1} \right)$</td>
</tr>
<tr>
<td>Model-based QVI</td>
<td>$\tilde{O} \left( \frac{SA}{(1-\lambda)^4 \varepsilon^2} \text{polylog}\delta^{-1} \right)$</td>
</tr>
<tr>
<td>Speedy Q-Learning</td>
<td>$\tilde{O} \left( \frac{SA}{(1-\lambda)^4 \varepsilon^2} \text{polylog}\delta^{-1} \right)$</td>
</tr>
</tbody>
</table>

*with optimized learning rate $\alpha_t = 1/(t + 1)^{4/5}$

$\tilde{O}(\cdot)$ hides poly-logarithmic (in $S, A, \varepsilon$) terms
3. Ergodic RL problems: Preliminaries

**Optimal policy**
Recall Bellman’s equation

\[ g^* + h^*(s) = \max_{a \in A} \left( r(s, a) + h^\top p(\cdot|s, a) \right), \quad \forall s \]

where \( g^* \) is the maximal gain, and \( h^* \) is the bias function (\( h^* \) is uniquely determined up to an additive constant). Note: \( g^* \) does not depend on the initial state for communicating MDPs.

Let \( a^*(s) \) denote any optimal action for state \( s \) (i.e., a maximizer in the above). Define the gap for sub-optimal action \( a \) at state \( s \):

\[ \phi(s, a) := (r(s, a^*(s)) - r(s, a)) + h^* \top (p(\cdot|s, a^*(s)) - p(\cdot|s, a)) \]
3. Ergodic RL problems: Preliminaries

MDP classes

$C_1$: **Ergodic**: every policy induces a single recurrent class, so any policy will reach every state after a finite number of steps.

$C_2$: **Unichain**: every policy induces a single recurrent class plus a (possibly empty) set of transient states.

$C_3$: **Communicating**: each two states are connected by some policy.

$C_4$: **Weakly communicating**: state space can be decomposes into two sets: a set where each state is reachable from every other state in the set under some policy, whereas all states in the other set are transient under all policies.

$$C_1 \subset C_2 \subset C_3 \subset C_4$$
Diameter $D$: defined as

$$D := \max_{s \neq s'} \min_{\pi} \mathbb{E}[T_{\pi}^{s,s'}]$$

where $T_{\pi}^{s,s'}$ denotes the first time step in which $s'$ is reached under $\pi$ starting from initial state $s$.

Remark: all communicating MDPs have a finite diameter.

Important parameters impacting performance

- Diameter $D$
- Gap $\Phi := \min_{s,a \neq a^*(s)} \phi(s,a)$
- Gap $\Delta := \min_{\pi} (g^* - g^\pi)$
3. Ergodic RL problems: Fundamental performance limits

- **Problem-specific regret lower bound**: (Burnetas-Katehakis)
  
  For any algorithm $\pi$,
  
  $$\lim_{T \to \infty} \inf \frac{R^\pi(T)}{\log(T)} \geq c_{bk} := \sum_{s,a} \phi(s,a) \inf \{ \text{KL}(p(\cdot|s,a), q) : q \in \Theta_{s,a} \}$$

  where $\Theta_{s,a}$ is the set of distributions $q$ s.t. replacing (only) $p(\cdot|s,a)$ by $q$ makes $a$ the unique optimal action in state $s$.
  
  - asymptotic (valid as $T \to \infty$)
  - valid for any ergodic MDP
  - scales as $\Omega(DSA \frac{A}{\Phi} \log(T))$ for specific MDPs

- **Minimax regret lower bound**: $\Omega(\sqrt{DSA T})$
  
  - non-asymptotic (valid for all $T \geq DSA$)
  - derived for a specific family of hard-to-learn communicating MDPs
Two types of algorithms targeting different regret guarantees:

- **Problem-specific guarantees**
  - MDP-specific regret bound scaling as $O(\log(T))$

- **Minimax guarantees**
  - Valid for a class of MDPs with $S$ states and $A$ actions, and (typically) diameter $D$
  - Scaling as $\widetilde{\Omega}(\sqrt{T})$
### 3. Ergodic RL problems: State-of-the-art

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Setup</th>
<th>Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-K</td>
<td>ergodic MDPs, known rewards</td>
<td>$O\left(c_{bk} \log(T)\right)$ – asympt.</td>
</tr>
<tr>
<td>OLP</td>
<td>ergodic MDPs, known rewards</td>
<td>$O\left(\frac{D^2 S A}{\Phi} \log(T)\right)$ – asympt.</td>
</tr>
<tr>
<td>UCRL</td>
<td>unichain MDPs</td>
<td>$O\left(\frac{S^5 A}{\Delta^2} \log(T)\right)$</td>
</tr>
<tr>
<td>UCRL2, KL-UCRL</td>
<td>communicating MDPs</td>
<td>$O\left(\frac{D^2 S A^2}{\Delta} \log(T)\right)$</td>
</tr>
<tr>
<td>Lower Bound</td>
<td>ergodic MDPs, known rewards</td>
<td>$\Omega\left(c_{bk} \log(T)\right), \Omega\left(\frac{DS A}{\Phi} \log(T)\right)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Setup</th>
<th>Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCRL2</td>
<td>communicating MDPs</td>
<td>$\tilde{O}\left(D S \sqrt{A T}\right)$</td>
</tr>
<tr>
<td>KL-UCRL</td>
<td>communicating MDPs</td>
<td>$\tilde{O}\left(D S \sqrt{A T}\right)$</td>
</tr>
<tr>
<td>REGAL</td>
<td>weakly comm. MDPs, known rewards</td>
<td>$\tilde{O}\left(B S \sqrt{A T}\right)^*$</td>
</tr>
<tr>
<td>A-J</td>
<td>communicating MDPs, known rewards</td>
<td>$\tilde{O}\left(D \sqrt{S A T}\right), T \geq S^5 A$</td>
</tr>
<tr>
<td>Lower Bound</td>
<td>known rewards</td>
<td>$\Omega\left(\sqrt{D S A T}\right), T \geq D S A$</td>
</tr>
</tbody>
</table>

* $B$ denotes the span of bias function of true MDP, and $B \leq D$
UCRL2 is an optimistic algorithm that works in episodes of increasing lengths.

- At the beginning of each episode $k$, it maintains a set of plausible MDPs $M_k$ (which contains the true MDP w.h.p.)
- It then computes an optimal policy $\pi_k$, which has the largest gain over all MDPs in $M_k$ ($\pi_k \in \arg\max_{M' \in M_k} g^\pi(M')$).
  - For computational efficiency, UCRL2 computes an $\frac{1}{\sqrt{t_k}}$-optimal policy, where $t_k$ is the starting step of episode $k$
  - To find a near-optimal policy, UCRL2 uses Extended Value Iteration
- It then follows $\pi_k$ within episode $k$ until the number of visits for some pair $(s, a)$ is doubled (and so, a new episode starts).
3. Ergodic RL: UCRL2

Notations:

- \( k \in \mathbb{N} \): index of an episode
- \( N_k(s, a) \): total no. visits of pairs \((s, a)\) until \( k \)
- \( \hat{p}_k(\cdot|s, a) \): empirical transition probability of \((s, a)\) made by observations up to \( k \)
- \( \hat{r}_k(s, a) \): empirical reward distribution of \((s, a)\) made by observations up to \( k \)
- \( \pi_k \): policy followed in episode \( k \)
- \( \mathcal{M}_k \): set of models for episode \( k \) (defined next)
3. Ergodic RL: UCRL2

- **The set of plausible MDPs** $\mathcal{M}_k$: for confidence parameter $\delta$, define

$$\mathcal{M}_k = \left\{ M' = (S, A, \tilde{r}, \tilde{p}) : |\tilde{r}(s, a) - \hat{r}_k(s, a)| \leq \sqrt{\frac{3.5 \log(2SA/t/\delta)}{N_k(s, a)^+}} \right\}$$

$$\|\tilde{p}(\cdot|s, a) - \hat{p}_k(\cdot|s, a)\|_1 \leq \sqrt{\frac{14S \log(2At/\delta)}{N_k(s, a)^+}} \right\}$$

Remark: $\mathcal{M}_k$ is defined using Hoeffding and Weissman inequalities (concentration of $L_1$ norm of categorical distributions)

- **Extended Value Iteration**: Define $\mathcal{P}_k$ as the collection of all transition kernels in $\mathcal{M}_k$. For all $s \in S$, starting from $u_0(s) = 0$:

$$u_{i+1}(s) = \max_{a \in A} \left\{ \tilde{r}(s, a) + \max_{q \in \mathcal{P}_k(s,a)} u_i^\top q \right\}.$$ 

- $\mathcal{P}_k$ is a polyhedron, so the inner maximization above can be done in $O(S)$ computations.
- To obtain an $\varepsilon$-optimal policy, the update is stopped when

$$\max_s (u_{i+1}(s) - u_i(s)) - \min_s (u_{i+1}(s) - u_i(s)) \leq \varepsilon$$
3. Ergodic RL: UCRL2

Algorithm 2 UCRL2

**Input:** Set confidence parameter $\delta \in (0, 1]$

**Initialize:** $N_0(s, a) = 0$, $v_0(s, a) = 0$ for all $(s, a)$. Set $t = 1$, $k = 1$

for episodes $k \geq 1$ do

Set $t_k = t$

Set $N_k(s, a) = N_{k-1}(s, a) + v_{k-1}(s, a)$ for all $(s, a)$

Compute empirical estimates $\hat{r}_k(s, a)$ and $\hat{p}_k(\cdot|s, a)$ for all $(s, a)$, and form $\mathcal{M}_k$

Find an $\frac{1}{\sqrt{t_k}}$-optimal policy $\tilde{\pi}_k$ using Extended Value Iteration

while $v_k(s_t, a_t) \geq N_k(s_t, a_t)$ do

Play action $a_t = \tilde{\pi}_k(s_t)$, and observe the next state $s_{t+1}$ and reward $r_t$

Update $N_k(s, a, x)$ and $v_{t+1}(s, a)$ for all actions $a$ and states $s, x$

end while

end for
Episodic RL

- **PSRL algorithm:**

- **ECFH algorithm:**

- **UBEV algorithm:**

- **UCBVI algorithm:**
Discounted RL

- **Sample complexity definition:**

- **R-max algorithm:**

- **MBIE algorithm:**

- **Delayed Q-Learning algorithm:**
Discounted RL

- **MoRmax algorithm:**

- **Minimax LB:**

- **UCRL-γ algorithm:**

- **Median-PAC algorithm:**
Discounted RL

- **Q-Learning algorithm:**

- **Convergence rates of Q-Learning algorithm:**

- **Speedy Q-Learning algorithm:**
References

Ergodic RL

- **B-K algorithm and problem-dependent LB:**

- **UCRL algorithm:**

- **OLP algorithm:**

- **UCRL2 algorithm and minimax LB:**
Ergodic RL

- **REGAL algorithm:**

- **KL-UCRL algorithm:**

- **A-J algorithm:**