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 algorithms
- Introduce regret and sample complexity, the two main performance metrics for RL algorithms
- Provide performance limits and state-of-the-art algorithms

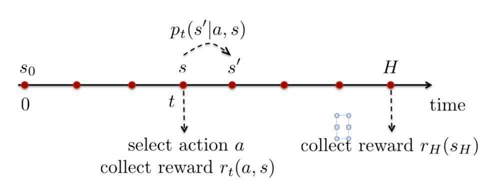
Lecture 3: Outline

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

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

Finite-horizon MDP to episodic RL problems

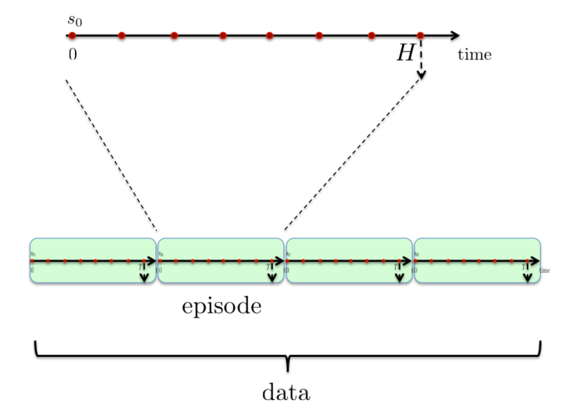


- Initial state s_0 (could be a r.v.)
- Transition probabilities at time t: $p_t(s'|s,a)$
- Reward at time t: $r_t(s,a)$ and at time H: $r_H(s)$
- **Unknown** transition probabilities and reward function
- Objective: *quickly* learn a policy $\pi^\star$ maximising over $\pi_0 \in MD$

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

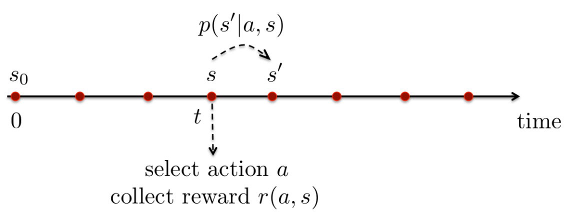
from either existing data (off-policy) or gathered data along the way (on-policy)

Finite-horizon MDP to episodic RL problems



- Data: K episodes of length H (actions, states, rewards)
- Learning algorithm π : data $\mapsto \pi_K \in MD$
- Performance of π: how close π_K is from the optimal policy $\pi^\star$

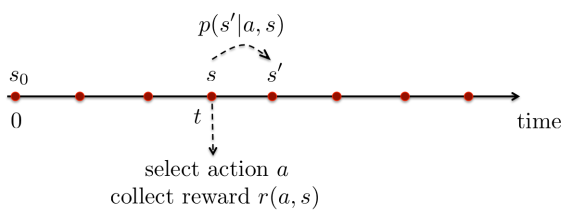
Infinite-horizon discounted MDP to discounted RL problems



- 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 $\pi \in MD$ maximising (over all possible policies)

$$V^{\pi}(s_0) := \lim_{T\to\infty} \mathbb{E}_{s_0}\left[\sum_{u=0}^{T} \lambda^u r(s_u^{\pi}, a_u^{\pi},)\right]$$

Expected average reward MDP to ergodic RL problems



- 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)

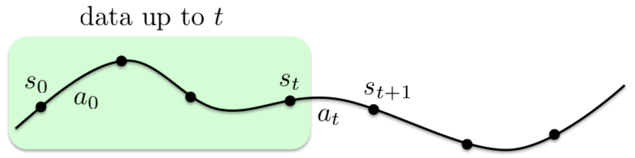
$$g^{\pi} = V^{\pi}(s_0) := \lim \inf_{T \to \infty} \frac{1}{T} \mathbb{E}_{s_0} \left[\sum_{u=0}^{T-1} r(s_u^{\pi}, a_u^{\pi},) \right]$$

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.

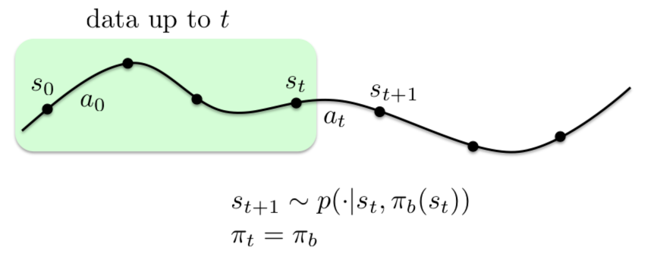


$$s_{t+1} \sim p(\cdot|s_t, \pi_t(s_t))$$

$$\pi_t = F_t(s_0, a_0, \ldots, s_{t-1}, a_{t-1}, s_t)$$

On vs. Off policies

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 π_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 π returns, after the end of the $(k-1)$-th episode, a policy π_k to be applied in the k-th episode.

> *The sample complexity of π is the minimum number SP^π of episodes such that for all $k \geq SP^\pi$, π_k is ϵ-optimal with probability at least $1-\delta$, i.e., for $k \geq SP^\pi$,*
>
> $$\mathbb{P}\left[V_T^{\pi_k} \geq V_T^{\pi^\star} - \epsilon\right] \geq 1-\delta$$

Sample complexity

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 π returns, after the end of the $(t-1)$-th step, a policy π_t to be applied in the t-th step.

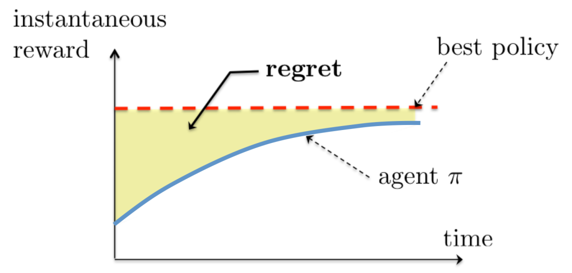
> *The sample complexity of π is the minimum number SP^π of steps such that for any $t \geq SP^\pi$, π_t is ϵ-optimal when starting in the current state s_t^π with probability at least $1-\delta$, i.e., for any $t \geq SP^\pi$,*
>
> $$\mathbb{P}\left[V^{\pi_t}(s_t^\pi) \geq V^{\pi^\star}(s_t^\pi) - \epsilon\right] \geq 1-\delta$$

Sample complexity can be defined for off-policies analogously.

Regret

A more appropriate performance metrics for on-policies? Capture the exploration vs. exploitation trade-off.

Regret of an algorithm π. Defined as the difference between the cumulative reward of the optimal policy and that gathered by π.



Regret

Regret of an algorithm π. Defined as the difference between the cumulative reward of the optimal policy and that gathered by π.

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

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

$$R^{\pi}(T) = K V_H^{\pi^\star} - \sum_{k=1}^{K} \mathbb{E}[V_H^{\pi_k}]$$

Regret

Regret of an algorithm π. Defined as the difference between the cumulative reward of the optimal policy and that gathered by π.

2. Discounted RL. An on-policy algorithm π returns, after the end of the $(t-1)$-th step, a policy π_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 π after T steps is:

$$R^{\pi}(T) = \sum_{t=1}^{T} \left(V^{\pi^\star}(s_t^{\pi^\star}) - V^{\pi_t}(s_t^{\pi}) \right)$$

Regret

Regret of an algorithm π. Defined as the difference between the cumulative reward of the optimal policy and that gathered by π.

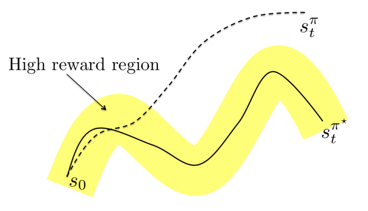
3. Ergodic RL. An on-policy algorithm π returns, after the end of the $(t-1)$-th step, a policy π_t to be applied in the t-th step.

The regret of π after T steps is:

$$R^{\pi}(T) = Tg^{\pi^\star} - \sum_{t=1}^{T} \mathbb{E}[r(s_t^{\pi}, a_t^{\pi})]$$

A remark on these definitions

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



Sample complexity:

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

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 $M = (\mathcal{S}, \mathcal{A}, (p(\cdot|s,a), r(s,a))_{s\in S, a\in A})$ a generic MDP.
  Two types of regret lower bounds:

  Problem-specific lower bound: for all M, $\forall$ algorithm π,
  $R^\pi(T) \geq F(M,T)$
  (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 π, $\forall T$,
  $R^\pi(T) \geq G(S, A, T, ...)$

1. Episodic RL problems: State-of-the-art

- **Regret minimisation**
  - Minimax lower bound $\Omega(\sqrt{HSAT})$
    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

Algorithm	Setup	Regret
PSRL	–	$\widetilde{\mathcal{O}}\left(HS\sqrt{AT}\right)$ – Bayesian regret
UCBVI-CH	known rewards	$\widetilde{\mathcal{O}}\left(H\sqrt{SAT}\right), \quad T \geq HS^3A,\, SA \geq H$
UCBVI-BF	known rewards	$\widetilde{\mathcal{O}}\left(\sqrt{HSAT}\right), \quad T \geq H^3S^3A,\, SA \geq H$
UBEV	–	$\widetilde{\mathcal{O}}\left(H^2\sqrt{SAT}\right), \quad T \geq S^5A^3$
Lower Bound	known rewards	$\Omega\left(\sqrt{HSAT}\right)$

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

Sample complexity

Algorithm	Setup	Sample Complexity
UCFH	known rewards	$\mathcal{O}\left(\frac{H^2CSA}{\varepsilon^2}\log\delta^{-1}\right)^*$
UBEV	–	$\mathcal{O}\left(\frac{H^4S^2A}{\varepsilon^2}\mathrm{polylog}\delta^{-1}\right)$
Lower Bound	–	$\widetilde{\Omega}\left(\frac{H^2SA}{\varepsilon^2}\log\delta^{-1}\right)$

* C is an upper bound on the number of reachable states under any pair (s, a)

1. Episodic RL: UCBVI

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

1. Episodic RL: UCBVI

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.

1. Episodic RL: UCBVI

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

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

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 $(s,a) \in \mathcal{S} \times \mathcal{A}$ do
            if $(s,a)$ is sampled so far then
                $Q_{k,h}(s,a) = \min\Big(Q_{k-1,h}(s,a), H, r(s,a) + \sum_y \hat{p}_k(y|s,a) V_{k,h+1}(s) + b_{k,h}(s,a)\Big)$
            end if
        end for
    end for
    for step $h = 1, \ldots, H$ do
        Take action $a_{k,h} = \mathrm{argmax}_a\, Q_{k,h}(s_{k,h}, a)$
    end for
end for
```

2. Discounted RL problems: State-of-the-art

No regret analysis for this class of RL problems. Two definitions for sample complexity:

- **Sample complexity:** The sample complexity of algorithm π is the minimum number SP^π of steps such that for any $t \geq SP^\pi$, π_t is ε-optimal when starting in the current state s_t^π with probability at least $1-\delta$:
$$\mathbb{P}\Big[V^{\pi_t}(s_t^\pi) \geq V^{\pi^\star}(s_t^\pi) - \varepsilon\Big] \geq 1-\delta$$
- **Q-sample complexity:** The sample complexity of algorithm π is the minimum number QSP^π of steps such that for any $t \geq QSP^\pi$, the current state-action values Q_t^π maintained by π is ε-close to $Q^\star$ with probability at least $1-\delta$, and for all state-action pairs, i.e.,
$$\mathbb{P}\Big[\|Q_t^\pi - Q^\star\|_\infty \leq \varepsilon\Big] \geq 1-\delta$$

Remark: the latter definition is stronger and implies the former

2. Discounted RL problems: Fundamental performance limit

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.

2. Discounted RL problems: State-of-the-art

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-γ (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

Algorithm	Setup	Sample Complexity
R-max	–	$\widetilde{\mathcal{O}}\left(\frac{S^2A}{(1-\lambda)^6\varepsilon^3}\log\delta^{-1}\right)$
MBIE	–	$\widetilde{\mathcal{O}}\left(\frac{S^2A}{(1-\lambda)^6\varepsilon^3}\log\delta^{-1}\right)$
Delayed Q-Learning	known reward	$\widetilde{\mathcal{O}}\left(\frac{SA}{(1-\lambda)^8\varepsilon^4}\log\delta^{-1}\right)$
MoRmax	–	$\widetilde{\mathcal{O}}\left(\frac{SA}{(1-\lambda)^6\varepsilon^2}\log\delta^{-1}\right)$
UCRL-γ	$\lvert\mathrm{supp}(p(\cdot\lvert s,a))\rvert \le 2,\ \forall(s,a)$	$\widetilde{\mathcal{O}}\left(\frac{N}{(1-\lambda)^3\varepsilon^2}\log\delta^{-1}\right)^*$
Lower Bound	–	$\widetilde{\Omega}\left(\frac{SA}{(1-\lambda)^3\varepsilon^2}\log\delta^{-1}\right)$

*N denotes the number of non-zero transitions in the true MDP

$\widetilde{\mathcal{O}}(\cdot)$ hides poly-logarithmic (in S, A, ε) terms

2. Discounted RL problems: State-of-the-art

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

Algorithm	**Q-Sample Complexity**
Q-Learning*	$\widetilde{\mathcal{O}}\left(\frac{SA}{(1-\lambda)^5\varepsilon^{5/2}}\text{polylog}\delta^{-1}\right)$
Model-free QVI	$\widetilde{\mathcal{O}}\left(\frac{SA}{(1-\lambda)^5\varepsilon^{2}}\text{polylog}\delta^{-1}\right)$
Model-based QVI	$\widetilde{\mathcal{O}}\left(\frac{SA}{(1-\lambda)^4\varepsilon^{2}}\text{polylog}\delta^{-1}\right)$
Speedy Q-Learning	$\widetilde{\mathcal{O}}\left(\frac{SA}{(1-\lambda)^4\varepsilon^{2}}\text{polylog}\delta^{-1}\right)$

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

$\widetilde{\mathcal{O}}(\cdot)$ hides poly-logarithmic (in S, A, ε) terms

3. Ergodic RL problems: Preliminaries

Optimal policy
Recall Bellman's equation

$$g^\star + h^\star(s) = \max_{a \in \mathcal{A}} \Big(r(s,a) + h^\top p(\cdot|s,a) \Big), \quad \forall s$$

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

Let $a^\star(s)$ 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:

$$\phi(s,a) := \big(r(s, a^\star(s)) - r(s,a) \big) + {h^\star}^\top \big(p(\cdot|s, a^\star(s)) - p(\cdot|s,a) \big)$$

3. Ergodic RL problems: Preliminaries

MDP classes

$\mathcal{C}_1$: **Ergodic:** every policy induces a single recurrent class, so any policy will reach every state after a finite number of steps.

$\mathcal{C}_2$: **Unichain:** every policy induces a single recurrent class plus a (possibly empty) set of transient states.

$\mathcal{C}_3$: **Communicating:** any pair of states are connected by some policy.

$\mathcal{C}_4$: **Weakly communicating:** the state space can be decomposed 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.

$$\mathcal{C}_1 \subset \mathcal{C}_2 \subset \mathcal{C}_3 \subset \mathcal{C}_4$$

3. Ergodic RL problems: Preliminaries

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 π staring from initial state s.

Remark: all communicating MDPs have a finite diameter.

Important parameters impacting performance

- Diameter D
- Gap $\Phi := \min_{s,a \neq a^\star(s)} \phi(s,a)$
- Gap $\Delta := \min_{\pi}(g^\star - g^\pi)$

3. Ergodic RL problems: Fundamental performance limits

- **Problem-specific regret lower bound:** (Burnetas-Katehakis)
  For any algorithm π,

$$\liminf_{T\to\infty} \frac{R^\pi(T)}{\log(T)} \geq c_{\mathrm{bk}} := \sum_{s,a} \frac{\phi(s,a)}{\inf\{\mathrm{KL}(p(\cdot|s,a),q) : q \in \Theta_{s,a}\}}$$

  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 $\Omega(\frac{DSA}{\Phi}\log(T))$ for specific MDPs

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

3. Ergodic RL problems: State-of-the-art

Two types of algorithms targeting different regret guarantees:

- Problem-specific guarantees
  - MDP-specific regret bound scaling as $\mathcal{O}(\log(T))$
  - Algorithms: B-K (Burnetas & Katehakis, 1997), OLP (Tewari & Bartlett, 2007), UCRL2 (Jaksch et al. 2009), KL-UCRL (Filippi et al. 2010)

- Minimax guarantees
  - Valid for a class of MDPs with S states and A actions, and (typically) diameter D
  - Scaling as $\widetilde{\Omega}(\sqrt{T})$
  - Algorithms: UCRL2 (Jaksch et al. 2009), KL-UCRL (Filippi et al. 2010), REGAL (Bartlett & Tewari, 2009), A-J (Agrawal & Jia, 2010)

3. Ergodic RL problems: State-of-the-art

Algorithm	Setup	Regret
B-K	ergodic MDPs, known rewards	$\mathcal{O}\left(c_{\text{bk}}\log(T)\right)$ – asympt.
OLP	ergodic MDPs, known rewards	$\mathcal{O}\left(\frac{D^2SA}{\Phi}\log(T)\right)$ – asympt.
UCRL	unichain MDPs	$\mathcal{O}\left(\frac{S^5A}{\Delta^2}\log(T)\right)$
UCRL2, KL-UCRL	communicating MDPs	$\mathcal{O}\left(\frac{D^2S^2A}{\Delta}\log(T)\right)$
Lower Bound	ergodic MDPs, known rewards	$\Omega\left(c_{\text{bk}}\log(T)\right),\ \Omega\left(\frac{DSA}{\Phi}\log(T)\right)$

Algorithm	Setup	Regret
UCRL2	communicating MDPs	$\widetilde{\mathcal{O}}\left(DS\sqrt{AT}\right)$
KL-UCRL	communicating MDPs	$\widetilde{\mathcal{O}}\left(DS\sqrt{AT}\right)$
REGAL	weakly comm. MDPs, known rewards	$\widetilde{\mathcal{O}}\left(BS\sqrt{AT}\right)^*$
A-J	communicating MDPs, known rewards	$\widetilde{\mathcal{O}}\left(D\sqrt{SAT}\right),\ T \geq S^5A$
Lower Bound	known rewards	$\Omega\left(\sqrt{DSAT}\right),\ T \geq DSA$

*B denotes the span of bias function of true MDP, and $B \leq D$

3. Ergodic RL: UCRL2

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 $\mathcal{M}_k$ (which contains the true MDP w.h.p.)
- It then computes an optimal policy π_k, which has the largest gain over all MDPs in $\mathcal{M}_k$ ($\pi_k \in \mathrm{argmax}_{M' \in \mathcal{M}_k, \pi} \, g^{\pi}(M')$).
    - 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 π_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
- π_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 δ, define

$$\mathcal{M}_k = \Bigg\{ M' = (\mathcal{S}, \mathcal{A}, \tilde{r}, \tilde{p}) : |\tilde{r}(s,a) - \hat{r}_k(s,a)| \le \sqrt{\frac{3.5 \log(2SAt/\delta)}{N_k(s,a)^+}}$$

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

 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 $s \in \mathcal{S}$, starting from $u_0(s) = 0$:

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

  - $\mathcal{P}_k$ is a polyhedron, so the inner maximization above can be done in $\mathcal{O}(S)$ computations.
  - To obtain an ε-optimal policy, the update is stopped when $\max_s(u_{i+1}(s) - u_i(s)) - \min_s(u_{i+1}(s) - u_i(s)) \le \varepsilon$

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
```

References

Episodic RL

- **PSRL algorithm:**
  I. Osband, D. Russo, and B. Van Roy, "(More) efficient reinforcement learning via posterior sampling," *Proc. NIPS*, 2013.
- **ECFH algorithm:**
  C. Dann and E. Brunskill, "Sample complexity of episodic fixed-horizon reinforcement learning," *Proc. NIPS*, 2015.
- **UBEV algorithm:**
  C. Dann, T. Lattimore, and E. Brunskill, "Unifying PAC and regret: uniform PAC bounds for episodic reinforcement learning," *Proc. NIPS*, 2017.
- **UCBVI algorithm:**
  M. Gheshlaghi Azar, I. Osband, and R. Munos, "Minimax regret bounds for reinforcement learning," *Proc. ICML*, 2017.

References

Discounted RL

- **Sample complexity definition:**
  S. M. Kakade, “On the sample complexity of reinforcement learning,” *PhD thesis, University of London*, 2003.
- **R-max algorithm:**
  R. I. Brafman and M. Tennenholtz, “R-max -a general polynomial time algorithm for near-optimal reinforcement learning,” *J. Machine Learning Research*, 2002.
- **MBIE algorithm:**
  A. L. Strehl and M. L. Littman, “A theoretical analysis of model-based interval estimation,” *Proc. ICML*, 2005.
- **Delayed Q-Learning algorithm:**
  A. L. Strehl, L. Li, and M. L. Littman, “Reinforcement learning in finite MDPs: PAC analysis,” *J. Machine Learning Research*, 2009.

References

Discounted RL

- **MoRmax algorithm:**
  I. Szita and Cs. Szepesvári, "Model-based reinforcement learning with nearly tight exploration complexity bounds," *Proc. ICML*, 2010.
- **Minimax LB:**
  M. Gheshlaghi Azar, R. Munos, and H. Kappen, "On the sample complexity of reinforcement learning with a generative model," *Proc. ICML*, 2012.
- **UCRL-γ algorithm:**
  T. Lattimore and M. Hutter, "Near-optimal PAC bounds for discounted MDPs," *Theoretical Computer Science*, 2014.
- **Median-PAC algorithm:**
  J. Pazis, R. Parr, and J. How, "Improving PAC Exploration Using the Median Of Means," *Proc. NIPS*, 2016.

References

Discounted RL

- **Q-Learning algorithm:**
  C. J. C. H. Watkins, “Learning from delayed rewards,” *PhD thesis, University of Cambridge,* 1989.
- **Convergence rates of Q-Learning algorithm:**
  E. Even-Dar and Y. Mansour, “Learning rates for Q-learning,” *J. Machine Learning Research*, 2003.
- **Speedy Q-Learning algorithm:**
  M. Gheshlaghi Azar, R. Munos, M. Ghavamzadeh, and H. J. Kappen, “Speedy Q-learning,” *Proc. NIPS*, 2011.

References

Ergodic RL

- **B-K algorithm and problem-dependent LB:**
  A. N. Burnetas and M. N. Katehakis, "Optimal adaptive policies for Markov decision processes," *Math. of OR*, 1997.
- **UCRL algorithm:**
  P. Auer & R. Ortner, "Logarithmic online regret bounds for undiscounted reinforcement learning," *Proc. NIPS*, 2006.
- **OLP algorithm:**
  A. Tewari and P. L. Bartlett, "Optimistic linear programming gives logarithmic regret for irreducible MDPs," *Proc. NIPS*, 2007.
- **UCRL2 algorithm and minimax LB:**
  P. Auer, T. Jaksch, and R. Ortner, "Near-optimal regret bounds for reinforcement learning," *J. Machine Learning Research*, 2010.

References

Ergodic RL

- **REGAL algorithm:**
  P. L. Bartlett and A. Tewari, "REGAL: A regularization based algorithm for reinforcement learning in weakly communicating MDPs," *Proc. UAI*, 2009.
- **KL-UCRL algorithm:**
  S. Filippi, O Cappé, and A. Garivier, "Optimism in reinforcement learning and Kullback-Leibler divergence," *Proc. Allerton*, 2010.
- **A-J algorithm:**
  S. Agrawal and R. Jia, "Posterior sampling for reinforcement learning: worst-case regret bounds," *Proc. NIPS*, 2017.