ECE239AS: Theoretical Foundation of Reinforcement Learning, Spring Quarter, 2023

Course Summary

Deyi Wang

Contents

1 Basic knowledge

1.1 Two systems

1.1.1 Multi-arm bandit system

The system is shown in Figure 1.

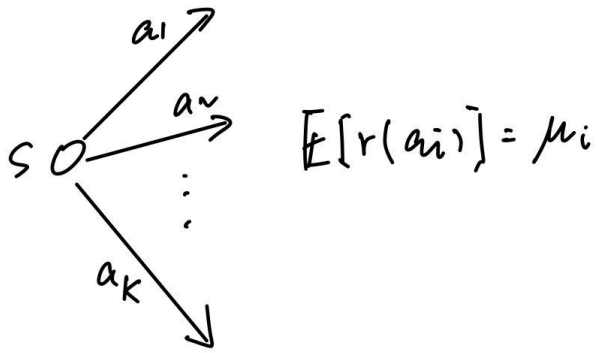

Figure 1: Multi-arm bandit system

In this system, the agent can choose from K actions $\{a_1, a_2, \ldots, a_K\}$ and when choosing each action a_i, the agent will receive a reward $r(a_i)$. The rewards satisfy that

$$\mathbb{E}[r(a_i)] = \mu_i$$

The goal is to find the arm with the maximum reward expectation.

1.1.2 Markov decision process (discounted)

A Markov decision process (MDP) is defined by

$$\mathcal{M} = (S, A, P, r, \gamma)$$

where S is the state set, A is the action set, $P : S \times A \to \Delta(S)$ is the transition function, $r : S \times A \to \mathbb{R}$ is the reward, and $\gamma < 1$ is the discounted factor.

In a MDP, the agent starts from a state $s_0 \in S$, chooses an action a_0 based on some policy, receives a reward $r(s_0, a_0)$, jumps to another state s_1 based on the transition function P, and repeats this process.

A policy is to describe how the agent chooses the next action based on the previous information

$$\pi : (S \times A \times r)^{0:\infty} \times S \to \Delta(A)$$

and we use Π to represent the set of all policies.

There are two kinds of special policies, stationary policies and stationary & deterministic policies. A stationary policy chooses the action only based on the current state:

$$\Pi^S = \{\pi | \pi : S \to \Delta(A)\}$$

A stationary & deterministic policy not only chooses the action only based on the current state, but also chooses one deterministic action:

$$\Pi^{SD} = \{\pi | \pi : S \to A\}$$

The goal is to find a good policy. Next we will see how to measure the performance of a policy.

1.2 Value function & Q-function

In a MDP, for a given policy π and a given initial state s or a given initial state action pair (s, a), the value function and the Q-function (action value function) are defined by

$$V^{\pi}(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | \pi, s_0 = s]$$
$$Q^{\pi}(s, a) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | \pi, s_0 = s, a_0 = a]$$

where $(s_0, a_0, r(s_0, a_0), s_1, a_1, r(s_1, a_1), \dots)$ is the trajectory the agent navigates following the policy π.

If the policy $\pi \in \Pi^S$, there are some useful properties of value function and Q-function.

1. $V^{\pi}(s) = \mathbb{E}_{a \sim \pi(s)}[Q^{\pi}(s, a)]$
2. $Q^{\pi}(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} V^{\pi}(s')$ and if we set $Q \in \mathbb{R}^{|S||A|}$, $r \in \mathbb{R}^{|S||A|}$, $P \in \mathbb{R}^{|S||A| \times |S|}$ and $V \in \mathbb{R}^{|S|}$, then we have
$$Q^{\pi} = r + \gamma P V^{\pi}$$

The goal is to find the policy with the highest value function or Q-function:

$$V^*(s) = \sup_{\pi} V^{\pi}(s)$$
$$Q^*(s, a) = \sup_{\pi} Q^{\pi}(s, a)$$

Although there are infinite complex polices, there is a very important theorem which can narrow down our search range.

Theorem 1.1 *$\exists \pi \in \Pi^{SD}$ such that $\forall s \in S, a \in A$,*

$$V^{\pi}(s) = V^*(s)$$
$$Q^{\pi}(s, a) = Q^*(s, a)$$

Next we will see how to find the optimal policy.

1.3 Bellman optimality, value iteration and policy iteration

For a given value function V, we define the greedy policy $\forall s \in S$,

$$\pi_V(s) = argmax_{a \in A}(r(s,a) + \gamma P(\cdot|, s, a)^T V)$$

We define the Bellman operator $\mathcal{T}V$ by $\forall s \in S$,

$$\mathcal{T}V(s) := \max_{a \in A}(r(s,a) + \gamma P(\cdot|, s, a)^T V)$$

and we have an important theorem of Bellman optimality:

Theorem 1.2 (Bellman Optimality) *$V = V^*$ if and only if*

$$V = \mathcal{T}V$$

and V^* is unique.

To prove this thorem, we need to first prove that $\mathcal{T}$ is contractive, i.e., $\forall V_1, V_2 \in \mathbb{R}^{|S|}$,

$$\|\mathcal{T}V_1 - \mathcal{T}V_2\|_2 \leq \gamma \|V_1 - V_2\|_2$$

and then use $\mathcal{T}V^* = V^*$ to finish the proof.

To be clear, usually, $V^{\pi_V} \neq \mathcal{T}V$, but we can prove that

$$V^{\pi_{V^*}} = \mathcal{T}V^* = V^*$$

Therefore, once we find the V^*, we can derive that greedy policy π_{V^*} and it will be the optimal policy.

1.3.1 Value iteration

Value iteration can help to find the V^*. The algorithm of value iteration is as follows:

Algorithm 1: Value iteration

```
Initialization: V^0 ← 0
for t = 1, 2, ... do
    V^t ← TV^{t-1}
end
```

With the property of contractive, we can easily prove that

$$\|V^t - V^*\|_2 \leq \frac{\gamma^t R_{max}}{1-\gamma}$$

where R_{max} is the maximum of all rewards.

Algorithm 2: Policy iteration

Initialization: $\pi^0 \leftarrow arbitrary$
for $t = 1, 2, \ldots$ **do**
  policy evaluation: get $V^{\pi^{t-1}}$
  policy optimization: $\pi^t = \pi_{V^{\pi^{t-1}}}$
end

1.3.2 Policy iteration

Policy iteration focuses on the policy, which means in each iteration, we will get a policy. The algorithm is shown in Algorithm 2:

It is proven that

$$\|V^{\pi^t} - V^*\|_2 \leq 2\frac{\gamma^t R_{max}}{1-\gamma}$$

1.4 Finite MDP and dynamical programming

A finite MDP $M = (S, A, P, r, H)$ replaces the discounted factor γ with the step number H, because to compute the value function or Q-function, we don't need infinite steps, but only H steps. We have $\forall s \in S, a \in A, h \in [0, H]$

$$V_h^\pi(s) = \mathbb{E}[\sum_{t=h}^{H} r(s_t, a_t)|\pi, s_h = s]$$
$$Q_h^\pi(s, a) = \mathbb{E}[\sum_{t=h}^{H} r(s_t, a_t)|\pi, s_h = s, a_h = a]$$

The goal is to find

$$V_0^* = \max_\pi V_0^\pi$$
$$Q_0^* = \max_\pi Q_0^\pi$$

The process is much easier and we call it dynamical programming. The algorithm is shown in Algorithm 3.

2 Model based RL

2.1 Generative model setting

For a MDP $M = (S, A, P, r, \gamma)$, if the transition function P is known, but we can query samples from the groundtruth P, then we can obtain an estimator the transition function $\hat{P}$. This

Algorithm 3: Dynamical programming

Initialization: $Q_H^*(s,a) = r(s,a)$
for $h = H-1, H-2, \ldots, 0$ **do**
  $V_{h+1}^*(s) = \max_a Q_{h+1}^*(s,a)$
  $Q_h^*(s,a) = r(s,a) + P(\cdot|s,a)^T V_{h+1}^*(s)$
  $\pi_h(s) = argmax_a Q_h^*(s,a)$
end

setting is called generative model setting, and reinforcement learning based on the generative model setting is called model based RL.

Suppose that for each state action pair $(s,a) \in S \times A$, we query m samples from $P(\cdot|s,a)$, and receive $s_{s,a}^{(1)}, s_{s,a}^{(2)}, \ldots, s_{s,a}^{(m)}$, then the estimator $\hat{P}$ could be defined as $\forall (s,a,s') \in S \times A \times S$,

$$\hat{P}(s'|s,a) = \frac{\sum_{i=1}^m 1(s_{s,a}^{(i)} = s')}{m}$$

To measure the error of the transition function, we explore the error between $P(\cdot|s,a)^T V$ and $\hat{P}(\cdot|s,a)^T V$ where $V \in \mathbb{R}^S$ is a value function.

It is obvious that $\hat{P}(\cdot|s,a)^T V = \frac{1}{m}\sum_{i=1}^m V(s_{s,a}^{(i)})$ and $P(\cdot|s,a)^T V = \mathbb{E}_{s' \sim P(\cdot|s,a)}[V(s')]$. Therefore, we can see $\{V(s_{s,a}^{(i)})\}_{i=1}^m$ as a series of i.i.d. random variables with the same expectation, and then we can apply the Hoeffding inequality. Since $\forall s' \in S$, $V(s') \leq \frac{1}{1-\gamma}$, finally we have $\forall (s,a) \in S \times A$ w.h.p.

$$|\hat{P}(\cdot|s,a)^T V - P(\cdot|s,a)^T V| \leq \tilde{O}(\frac{1}{(1-\gamma)\sqrt{m}})$$

where $\tilde{O}()$ omits the log factor which contains the union bound.

2.2 Value iteration

Algorithm 4 is a simple model based value iteration

Algorithm 4: Simple model based value iteration

Initialization: $V^0 \leftarrow 0$
for $t = 1, 2, \ldots$ **do**
  $V^t \leftarrow \hat{\mathcal{T}} V^{t-1}$ where $\forall s \in S$,

$$\hat{\mathcal{T}} V(s) := \max_a (r(s,a) + \hat{P}(\cdot|s,a)^T V)$$

end

Suppose that in each iteration we query m samples from $P(\cdot|s,a)$ for each $(s,a) \in S \times A$. Since the union bound only causes change on the log factor, the state of the error between

$P(\cdot|s,a)^T V$ and $\hat{P}(\cdot|s,a)^T V$ remains for each iteration. With the definition of $\hat{\mathcal{T}}$, we have for each iteration, $\forall(s,a) \in S \times A$ w.h.p.

$$\|\hat{\mathcal{T}}V - \mathcal{T}V\|_\infty \leq \tilde{O}(\frac{1}{(1-\gamma)\sqrt{m}})$$

Then we have

$$\begin{aligned}
\|V^{(i)} - V^*\|_\infty &= \|\hat{\mathcal{T}}V^{(i-1)} - \mathcal{T}V^*\|_\infty \\
&= \|\mathcal{T}V^{(i-1)} - \mathcal{T}V^* + \hat{\mathcal{T}}V^{(i-1)} - \mathcal{T}V^{(i-1)}\|_\infty \\
&\leq \|\mathcal{T}V^{(i-1)} - \mathcal{T}V^*\|_\infty + \|\hat{\mathcal{T}}V^{(i-1)} - \mathcal{T}V^{(i-1)}\|_\infty \\
&\leq \gamma\|V^{(i-1)} - V^*\|_\infty + \tilde{O}(\frac{1}{(1-\gamma)\sqrt{m}})
\end{aligned}$$

By recursion we have

$$\begin{aligned}
\|V^{(R)} - V^*\|_\infty &\leq \gamma^R\|V^*\|_\infty + \sum_{i=0}^{R-1}\gamma^i\tilde{O}(\frac{1}{(1-\gamma)\sqrt{m}}) \\
&\leq \frac{\gamma^R}{1-\gamma} + \frac{1}{1-\gamma}\tilde{O}(\frac{1}{(1-\gamma)\sqrt{m}})
\end{aligned}$$

If we aim to a ϵ-optimal value function, we first let the first term approaches 0:

$$\frac{\gamma^R}{1-\gamma} = \frac{(1-(1-\gamma))^R}{1-\gamma} \approx \frac{1-R(1-\gamma)}{1-\gamma}$$

then $R = O(\frac{1}{1-\gamma})$. Then let $\frac{1}{1-\gamma}\tilde{O}(\frac{1}{(1-\gamma)\sqrt{m}}) \leq \epsilon$, we get

$$m = \tilde{O}(\frac{1}{(1-\gamma)^4\epsilon^2})$$

Therefore, to get a ϵ-optimal value function, the number of samples we need is

$$|S||A|mR = \tilde{O}(\frac{|S||A|}{(1-\gamma)^5\epsilon^2})$$

2.3 Error of the optimal policy

Through value iteration or policy iteration, we can get the optimal policy $\hat{\pi}^*$ of the MDP $\hat{M} = (S, A, \hat{P}, r, \gamma)$. Let Q^π denote the value of the policy π in M, $\hat{Q}^\pi$ denote the value of the policy π in $\hat{M}$, and we measure the error of the optimal policy by $\|Q^{\hat{\pi}^*} - Q^*\|_\infty$.

We in advance claim a theorem which will be useful afterwards without the proof.

Theorem 2.1 ***(bound of*** $\|\cdot\|_{TV}$***)*** *$\forall(s,a) \in S \times A$, w.h.p.*

$$\|\hat{P}(\cdot|s,a) - P(\cdot|s,a)\|_{TV} := \sum_{s'}|\hat{P}(s'|s,a) - P(s'|s,a)| \leq \epsilon := \tilde{O}(\sqrt{\frac{|S|}{m}})$$

where m is the number of samples from $P(\cdot|s,a)$ for each (s,a).

First we have

$$\begin{aligned}\|Q^{\hat{\pi}^*} - Q^*\|_\infty &= \|Q^{\hat{\pi}^*} - \hat{Q}^{\hat{\pi}^*} + \hat{Q}^{\hat{\pi}^*} - Q^*\|_\infty \\ &\leq \|Q^{\hat{\pi}^*} - \hat{Q}^{\hat{\pi}^*}\|_\infty + \|\hat{Q}^{\hat{\pi}^*} - Q^*\|_\infty\end{aligned}$$

Based on the transition function P from a state action pair (s, a) to the next state s', we combine it with the stationary policy and propose a new transition function P^π from a state action pair (s, a) to the next state action pair (s', a'):

$$P^\pi_{(s',a'),(s,a)} := P(s'|s, a)Pr(\pi(s') = a')$$

then we have

$$Q^\pi = (I - \gamma P^\pi)^{-1} r$$

and then

$$\begin{aligned}Q^\pi - \hat{Q}^\pi &= Q^\pi - (I - \gamma \hat{P}^\pi)^{-1} r \\ &= Q^\pi - (I - \gamma \hat{P}^\pi)^{-1}(I - \gamma P^\pi)Q^\pi \\ &= (I - \gamma \hat{P}^\pi)^{-1}(I - \gamma \hat{P}^\pi)Q^\pi - (I - \gamma \hat{P}^\pi)^{-1}(I - \gamma P^\pi)Q^\pi \\ &= \gamma(I - \gamma \hat{P}^\pi)^{-1}(P^\pi - \hat{P}^\pi)Q^\pi \\ &= \gamma(I - \gamma \hat{P}^\pi)^{-1}(P - \hat{P})V^\pi\end{aligned}$$

According to Theorem 2.1, we have

$$\|(P - \hat{P})V\|_\infty = \|\sum_{s'}(P - \hat{P})(s'|s, a)V(s')\|_\infty \leq \|\hat{P}(\cdot|s, a) - P(\cdot|s, a)\|_{TV}\|V\|_\infty \leq \frac{\epsilon}{1 - \gamma}$$

then

$$\begin{aligned}\|Q^\pi - \hat{Q}^\pi\|_\infty &= \|\gamma(I - \gamma \hat{P}^\pi)^{-1}(P - \hat{P})V^\pi\|_\infty \\ &\leq \|\gamma(I - \gamma \hat{P}^\pi)^{-1}\frac{\epsilon}{1 - \gamma}1\|_\infty \\ &= \|\gamma \sum_{i=0}^{\infty} \gamma^i (\hat{P}^\pi)^i \frac{\epsilon}{1 - \gamma}1\|_\infty \\ &\leq \gamma \sum_{i=0}^{\infty} \gamma^i \frac{\epsilon}{1 - \gamma} \\ &= \frac{\gamma\epsilon}{(1 - \gamma)^2}\end{aligned}$$

For $\hat{Q}^{\hat{\pi}^*} - Q^*$ we have $\forall(s, a) \in S \times A$,

$$\begin{aligned}|\hat{Q}^{\hat{\pi}^*}(s, a) - Q^*(s, a)| &= |\max_\pi \hat{Q}^\pi(s, a) - \max_\pi Q^\pi(s, a)| \\ &\leq \max_\pi |\hat{Q}^\pi(s, a) - Q^\pi(s, a)| \\ &\leq \frac{\gamma\epsilon}{(1 - \gamma)^2}\end{aligned}$$

Therefore,

$$\|Q^{\hat{\pi}^*} - Q^*\|_\infty \leq \frac{2\gamma\epsilon}{(1-\gamma)^2} = \tilde{O}(\frac{\gamma}{(1-\gamma)^2}\sqrt{\frac{|S|}{m}})$$

If we aim to a ϵ'-optimal policy, then

$$m = \tilde{\Theta}(\frac{|S|}{\epsilon'^2(1-\gamma)^4})$$

where $\tilde{\Theta}()$ not only omits the log operator like $\tilde{O}()$, but also omits the factor γ^2 which is near 1.

3 Online RL

3.1 Outline

The setting of online reinforcement learning is that only by taking actions the agent can get access to the knowledge of the environment. Two main approaches of online reinforcement learning are model-free approach which is so called Q-learning and model-based approach.

Q-learning means that as the learning process moves on, the Q-value of each state action pair is updated; the model-based approach uses the history of trajectories to predict the transition function, and makes use of it.

There are four algorithm examples in this section. Two are about the multi-arms bandit problem, the other two are with the finite-horizon MDP setting.

The measurement of online reinforcement learning is regret which is the sum of the difference between the value of the optimal policy and the current policy.

3.2 Algorithms of the multi-arms bandit problem

3.2.1 Multi-arms bandit setting

There are one state and K actions in the multi-arms bandit setting, as shown in Figure 2.

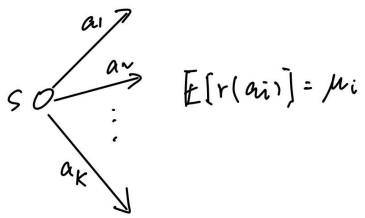

Figure 2: Multi-arms bandit setting

Different arms have different reward distributions with different expectation:

$$\mathbb{E}[r(a_i)] = \mu_i,\ i \in [K]$$

Every iteration t, the agent will pick one arm $I_t \in [K]$, and the regret at the T-th iteration is

$$R(T) = T \max_i \mu_i - \sum_{t=1}^{T} \mu_{I_t}$$

Algorithms of the multi-arms bandit problem aim to achieve $R(T) \leq O(T^\delta)$ where $\delta < 1$, so that the agent could find the ϵ-optimal arm with any arbitrary ϵ.

3.2.2 Explore first algorithm

The explore first algorithm explores all the arms several times at first to gain the knowledge of the rewards of the arms, and then chooses the best arm judged by the existed knowledge afterwards.

Algorithm 5: Explore first algorithm

1 **Exploration:** Try each arm N/K times, and for each arm a_i gain the rewards: $\{r_{i1}, r_{i2}, \ldots, r_{iN/K}\}$;

2 **Exploitation:** Play $a^* = argmax_{a_i} \mathbb{E}[\sum_{j=1}^{N/K} r_{ij}]$ in all remaining $T - N$ times.

In the first N times, the agent picks each arm uniform randomly, so the regret is bounded by N,

$$R(N) \leq N$$

After the first N times, the agent learnt the estimated expectation reward of each arm $\hat{\mu_i} = \mathbb{E}[\sum_{j=1}^{N/K} r_{ij}$. According to Hoeffding's inequality, w.p. at least $1 - \delta$,

$$|\hat{\mu_i} - \mu_i| \leq \sqrt{\frac{K \log(2/\delta)}{2N}}$$

To make the probability high enough, we set $\delta = \frac{2}{T^4}$, then we have

$$|\hat{\mu_i} - \mu_i| \leq \sqrt{\frac{2K \log(T)}{N}}$$

In the remaining $T - N$ times, the non zero regret occurs only if the agent chooses the arm which is not the true optimal one. However, even this, the difference between the chosen one and the true optimal one is bounded.

Let the chosen arm be a and the true optimal arm be a^*. The event that the agent chooses the not true optimal arm only happens if $\hat{\mu_a} \geq \hat{\mu_{a^*}}$, which means that

$$\mu_a + \sqrt{\frac{2K \log(T)}{N}} \geq \hat{\mu_a} \geq \hat{\mu_{a^*}} \geq \mu_{a^*} - \sqrt{\frac{2K \log(T)}{N}}$$

Then the regret of the whole T iterations can be written as

$$\begin{aligned} R(T) &\leq N + (T-N)(\mu_{a^*} - \mu_a) \\ &\leq N + 2\sqrt{\frac{2K\log(T)}{N}}(T-N) \\ &\leq N + \sqrt{\frac{8KT^2\log(T)}{N}} \end{aligned}$$

The minimum occurs when we choose $N = 2T^{2/3}(K\log T)^{1/3}$,

$$R(T) \leq 4T^{2/3}(K\log T)^{1/3}$$

Considering that the inequality could fail w.p. no more than $\delta = \frac{2}{T^4}$, and if so, $R(T) \leq T$, then,

$$\begin{aligned} R(T) &\leq 4T^{2/3}(K\log T)^{1/3}(1 - \frac{2}{T^4}) + T\frac{2}{T^4} \\ &\leq O(T^{2/3}(K\log T)^{1/3}) \end{aligned}$$

3.2.3 Upper Confidence Bound (UCB) algorithm

The UCB algorithm won't treat every arm equally. Instead, the algorithm will pick the arm with the largest reward upper confidence bound.

Algorithm 6: UCB bandit algorithm

Initialization: $\forall a_i,\ i \in [K],\ N_i \leftarrow 0,\ \hat{\mu}_i \leftarrow 0$
for $t = 1, 2, \ldots, T$ **do**
  Execute arm $I_t = argmax_i(\hat{\mu}_i + \sqrt{2\frac{\log\frac{2TK}{\delta}}{N_i}})$
  Observe r_{I_t}
  Update $\hat{\mu_{I_t}} \leftarrow \frac{N_{I_t}\hat{\mu_{I_t}} + r_{I_t}}{N_{I_t}+1}$, $N_{I_t} \leftarrow N_{I_t} + 1$
end

The name of the algorithm derives from that $\hat{\mu}_i + \sqrt{2\frac{\log\frac{2TK}{\delta}}{N_i}}$ is the upper bound estimator to μ_i.

For each arm a_i, suppose that its reward is observed as $\{r_1, r_2, \ldots, r_{N_i}\}$, then the sequence of $\{r_j - \mu_i\}$, $j = 1, 2, \ldots, N_i$ is a martingale difference sequence. According to Azuma Hoeffding's inequality, for all arms $a_i,\ i \in [K]$ and $t \in [T]$, w.p. at least $1 - \delta$,

$$|N_i\mu_i - \sum_{j=1}^{N_i} r_j| \leq \sqrt{2N_i\log\frac{2TK}{\delta}}$$

i.e.,

$$\hat{\mu}_i - \sqrt{2\frac{\log\frac{2TK}{\delta}}{N_i}} \leq \mu_i \leq \hat{\mu}_i + \sqrt{2\frac{\log\frac{2TK}{\delta}}{N_i}}$$

Let the chosen arm be a and the true optimal arm be a^*. The event that the agent chooses the not true optimal arm only happens if $\hat{\mu_a} \geq \hat{\mu_{a^*}}$, which means that

$$\mu_a + \sqrt{2\frac{\log\frac{2TK}{\delta}}{N_i}} \geq \hat{\mu_a} \geq \hat{\mu_{a^*}} \geq \mu_{a^*} - \sqrt{2\frac{\log\frac{2TK}{\delta}}{N_i}}$$

We use N_i^t to represent more clearly, then the regret

$$\begin{aligned} R(T) &\leq \sum_{t=1}^{T} 2\sqrt{2\frac{\log\frac{2TK}{\delta}}{N_i^t}} \\ &\leq 2\sqrt{2\log\frac{2TK}{\delta}} \sum_{t=1}^{T} \sqrt{\frac{1}{N_i^t}} \end{aligned}$$

We see $\sum_{t=1}^{T} \sqrt{\frac{1}{N_i^t}}$ in each arm, i.e., each arm a will be picked N_a^T times in total, then

$$\begin{aligned} \sum_{t=1}^{T} \sqrt{\frac{1}{N_i^t}} &= \sum_{a} \sum_{i=1}^{N_a^T} \sqrt{\frac{1}{i}} \\ &\leq \sum_{a} 2\sqrt{N_a^T} \\ &\leq 2K\sqrt{\frac{T}{K}} \\ &= 2\sqrt{KT} \end{aligned}$$

Therefore, the regret

$$R(T) \leq 4\sqrt{2KT\log\frac{2TK}{\delta}}$$

.

This is only the general case and some other limitation to N_a^T is omitted here.

3.3 Algorithms with the finite-horizon MDP setting

3.3.1 Finite-horizon MDP setting

Consider a tabular episodic MDP (S, A, H, P, r) here. The algorithms aim to find the optimal policy $\pi^* = \{\pi_1^*, \pi_2^*, \ldots, \pi_H^*\}$, where $\pi_h^* : S \to A$ is the policy used at the h-th step of one episode.

The definition of regret varies with different approaches.

3.3.2 Q-learning with UCB algorithm

In the Q-learning with UCB algorithm, the agent will learn the Q-value of the state action pair it chooses, and chooses the action based on the Q-value. In each episode, the agent will learn a policy starting from one arbitrary state.

Suppose that in the k-th episode ($k \in [K]$), given the state state x_1^k, the agent learns a policy π_k, then the regret

$$R(K) = \sum_{k=1}^{K} (V_1^*(x_1^k) - V_1^{\pi_k}(x_1^k))$$

Algorithm 7: Q-learning with UCB algorithm

Initialization: $\forall (x, a, h) \in S \times A \times [H]$, $Q_h(x, a) \leftarrow H$, $N_h(x, a) \leftarrow 0$
for $k = 1, 2, \ldots, K$ **do**
  receive arbitrary x_1
  for $h = 1, 2, \ldots, H$ **do**
    $a_h \leftarrow argmax_{a'} Q_h(x_h, a')$ (receive x_{h+1})
    $N_h(x_h, a_h) \leftarrow N_h(x_h, a_h) + 1$
    $t \leftarrow N_h(x_h, a_h), b_t \leftarrow c\sqrt{\frac{H^3\iota}{t}}, \iota \leftarrow \log \frac{2SAT}{p}, \alpha_t \leftarrow \frac{H+1}{H+t}$
    $Q_h(x_h, a_h) \leftarrow (1 - \alpha_t) Q_h(x_h, a_h) + \alpha_t [r_h(x_h, a_h) + V_{h+1}(x_{h+1}) + b_t]$
    $V_h(x_h) \leftarrow \min\{H, \max_{a'} Q_h(x_h, a')\}$
  end
end

Sketch of Proof:

1. Q-update

Suppose before the k-th episode, (x, a) has been visited at the h-th step for t times ($k_1, k_2, \ldots, k_t < k$), then before the start of the k-th episode,

$$Q_h^k(x, a) = \alpha_t^0 H + \sum_{i=1}^{t} \alpha_t^i [r_h(x, a) + V_{h+1}^{k_i}(x_{h+1}^{k_i}) + b_i]$$

where $\sum_{i=0}^{t} \alpha_t^i = 1$ and $\alpha_t^0 = 1$ if $t = 0$, $\alpha_t^0 = 0$ if $t \neq 0$. α_t^i are different from α_t and have some properties.

2. Bound on $Q^k - Q^*$

To apply the Azuma Hoeffding's inequality, we need to construct martingale difference sequence in the difference between Q^k and Q^*. The key here is to create $+V_{h+1}^*(x_{h+1}^{k_i}) - V_{h+1}^*(x_{h+1}^{k_i})$ from nothing and peer them with different items. Accordingly, we have,

$$(Q_h^k - Q_h^*)(x, a) = \alpha_t^0 (H - Q_h^*(x, a)) + \sum_{i=1}^{t} \alpha_t^i [b_i + (V_{h+1}^{k_i} - V_{h+1}^*)(x_{h+1}^{k_i}) + V_{h+1}^*(x_{h+1}^{k_i}) - P(\cdot|x, a)^T V_{h+1}^*]$$

Obviously, $0 \leq H - Q_h^*(x, a) \leq H$;

$V_{h+1}^*(x_{h+1}^{k_i})$ is the sample from V_{h+1}^* and $P(\cdot|x, a)^T V_{h+1}^*$ is the expectation. So $\{V_{h+1}^*(x_{h+1}^{k_i}) - P(\cdot|x, a)^T V_{h+1}^*\}$ is a martingale difference sequence. Therefore, with Azuma Hoeffding's inequality, union bound, and some properties of α_t^i, we have

$$|\sum_{i=1}^{t} \alpha_t^i (V_{h+1}^*(x_{h+1}^{k_i}) - P(\cdot|x, a)^T V_{h+1}^*)| \leq c\sqrt{\frac{H^3\iota}{t}}$$

for some constant c and $\forall(x,a,h)$.

Choosing $b_i = c\sqrt{\frac{H^3\iota}{i}}$, with some properties of α_t^i, we have

$$c\sqrt{\frac{H^3\iota}{t}} \leq \sum_{i=1}^{t} \alpha_i^t b_i \leq 2c\sqrt{\frac{H^3\iota}{t}}$$

Let $\frac{\beta_t}{2} = \sum_{i=1}^{t} \alpha_i^t b_i$, then we have

$$(Q_h^k - Q_h^*)(x,a) \leq \alpha_t^0 H + \sum_{i=1}^{t} \alpha_t^i (V_{h+1}^{k_i} - V_{h+1}^*)(x_{h+1}^{k_i}) + \beta_t$$

and

$$(Q_h^k - Q_h^*)(x,a) \geq \sum_{i=1}^{t} \alpha_t^i (V_{h+1}^{k_i} - V_{h+1}^*)(x_{h+1}^{k_i})$$

and using recursion from $h = H$ to $h = 1$ we have

$$(Q_h^k - Q_h^*)(x,a) \geq 0$$

2. Bound on $R(K)$

With the bound on $Q^k - Q^*$, the regret is also bounded

$$R(K) \leq \sum_{k=1}^{K} (V_1^k(x_1^k) - V_1^{\pi_k}(x_1^k))$$

Denote by

$$\delta_h^k := (V_h^k - V_h^{\pi_k})(x_h^k)$$

We aim to bound $R(K) \leq \sum_{k=1}^{K} \delta_1^k$ by recursion.

Since π_k is the policy summarized from the agent's choice in the k-th episode, we have

$$\delta_h^k \leq (Q_h^k - Q_h^{\pi_k})(x_h^k, a_h^k)$$

By creating Q_h^* from nothing, we have

$$\delta_h^k \leq \alpha_t^0 H + \sum_{i=1}^{t} \alpha_t^i \phi_{h+1}^{k_i} + \beta_t - \phi_{h+1}^k + \delta_{h+1}^k + \xi_{h+1}^k$$

where $\phi_h^k = (V_h^k - V_h^*)(x_h^k)$ and $\xi_{h+1}^k = P_h(\cdot|x_h^k, a_h^k)^T(V_{h+1}^* - V_{h+1}^k) - (V_{h+1}^* - V_{h+1}^k)(x_{h+1}^k)$

Since once visited, the α_t^0 for each (x,a) at step h will turn from 1 to 0 forever, then

$$\sum_{k=1}^{K} \alpha_t^0 H \leq SAH$$

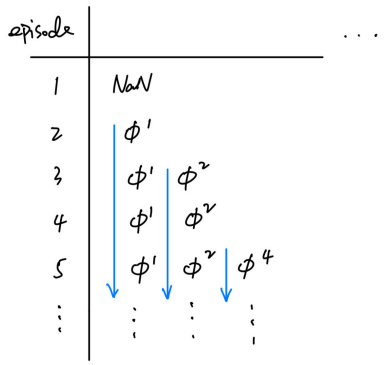

Figure 3: Demonstration to $\sum_{i=1}^{t} \alpha_t^i \phi_{h+1}^{k_i}$.

Since t in the second term $\sum_{i=1}^{t} \alpha_t^i \phi_{h+1}^{k_i}$ means that before the k-th episode, the state action pair has been visited t times in total, then if the state action pair is visited at the k-th episode, it will be counted again and again afterwards, as shown in Figure 3.

Obviously, we have

$$\sum_{k=1}^{K} \sum_{i=1}^{t} \alpha_t^i \phi_{h+1}^{k_i} \leq \sum_{k=1}^{K} \phi_{h+1}^{k} \sum_{t=k+1}^{\infty} \alpha_t^k$$

and with some property of α_t^i, we continuously have

$$\sum_{k=1}^{K} \sum_{i=1}^{t} \alpha_t^i \phi_{h+1}^{k_i} \leq (1 + \frac{1}{H}) \sum_{k=1}^{K} \phi_{h+1}^{k}$$

Now with the fact that $\phi_h^k \leq \delta_h^k$ we have

$$\sum_{k=1}^{K} \delta_h^k \leq SAH + (1 + \frac{1}{H}) \sum_{k=1}^{K} \delta_{h+1}^k + \sum_{k=1}^{K} (\beta_{n_h^k} + \xi_{h+1}^k)$$

Using recursion we have

$$\sum_{k=1}^{K} \delta_1^k \leq O(H^2 SA + \sum_{h=1}^{H} \sum_{k=1}^{K} (\beta_{n_h^k} + \xi_{h+1}^k))$$

For $\sum_{h=1}^{H} \sum_{k=1}^{K} \beta_{n_h^k}$, we can use the fact that $\beta_{n_h^k} = O(1)\sqrt{\frac{H^3 \iota}{n_h^k}}$ and $\sum_{x,a} n_h^K(x,a) = K$, then

$$\sum_{h=1}^{H} \sum_{k=1}^{K} \beta_{n_h^k} \leq O(\sqrt{H^5 SAK\iota})$$

For $\sum_{h=1}^{H} \sum_{k=1}^{K} \xi_{h+1}^k$, we can apply Azuma Hoeffding inequality and

$$\sum_{h=1}^{H} \sum_{k=1}^{K} \xi_{h+1}^k \leq c\sqrt{H^3 K\iota}$$

Since $\sqrt{H^5SAK\iota} > \sqrt{H^3K\iota}$, we finally have

$$\sum_{k=1}^{K} \delta_1^k \leq O(H^2SA + \sqrt{H^5SAK\iota})$$

3.3.3 UCB value iteration algorithm

The UCB value iteration algorithm is a model-based algorithm, which means it will use the history information to estimate the transition function P, and use the estimator $\hat{P}$ to run value iteration to obtain the near optimal policy.

Here we will omit the detail of the proof and introduce the algorithm roughly.

Algorithm 8: UCB value iteration algorithm

for $k = 1, 2, \ldots, K$ **do**
  Compute the estimator $\hat{P}^k$ using the history information
  Compute the reward bonus b^k using the history information
  Run value iteration with transition function $\hat{P}^k$ and reward with additional bonus $r + b_k$
  Through the value iteration gain a policy π_k
  Execute the policy π_k to generate a new trajectory, where there will be some data added to the history information
end

The procedure of the proof is as follow:

1. The error of the transition function is bounded; Azuma Hoeffding's inequality
2. With the reward bonus, the error of the transition function could be covered, which results in optimism, i.e., the estimated value of each state is larger than the ground truth optimal value;
3. Similar with the previous Q-learning algorithm, the regret is bounded by the sum of difference between the learnt value and the policy value, which is also bounded by the error bound of the transition function and reward bonus.

4 Linear MDPs

4.1 Outline

The linear setting is to deal with the problem with infinite states and actions. First we focus on the offline setting. We will use the bandit setting to learn the error of estimation in a simple case, then we apply this knowledge to linear MDP setting and explore the error along with the value iteration process. In the online setting, we focus on how to deal with "optimism in the face of uncertainty"(OFU). In the bandit setting, we use the "confidence set" to ensure that it contains the groundtruth mapping μ^* w.h.p.; in the MDP setting, we use the ϵ-cover Net to deal with the union bound of infinite functions.

4.2 Model based linear bandit

There is a feature map mapping the action space to a d dimension space: $\phi : A \to \mathbb{R}^d$, and the reward of an action is,

$$\begin{aligned} R_a &= \langle \phi(a), \theta^* \rangle + \epsilon \\ \mathbb{E}[R_a|a] &= \langle \phi(a), \theta^* \rangle \end{aligned}$$

where ϵ is a zero-mean noise, and without the loss of generalization, we usually set that $\forall a, \|\phi(a)\| \leq 1$.

If θ^* is known, it is obvious that the optimal arm is $a^* = argmax_{a \in A} \langle \phi(a), \theta^* \rangle$. Our goal is to find a good estimator of θ^* when θ^* is unknown.

Suppose that we play the action $a_1, a_2, \ldots, a_n$ and receive the rewards $r_1, r_2, \ldots, r_n$, then by solving

$$\hat{\theta}_{LS} = argmax_\theta \sum_{i=1}^{n} (\phi_i^T \theta - r_i)$$

, the estimator is

$$\hat{\theta}_{LS} = \Phi^\dagger r = (\Phi^T \Phi)^{-1} \Phi^T r$$

where $\Phi = \begin{bmatrix} \phi_1^T \\ \phi_2^T \\ \vdots \\ \phi_n^T \end{bmatrix} \in \mathbb{R}^{d \times n}, r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} \in \mathbb{R}^n$ and $\Phi^\dagger$ is the Moore–Penrose inverse of Φ.

By proving that $\forall \phi, |\phi^T \hat{\theta}_{LS} - \phi^T \theta^*|$ is bounded we can say $\hat{\theta}_{LS}$ is a good estimator. Let $\Sigma = \Phi^T \Phi$, we have,

$$\begin{aligned} |\phi^T \hat{\theta}_{LS} - \phi^T \theta^*| &= |\phi \Sigma^{-1/2} \Sigma^{1/2} (\hat{\theta}_{LS} - \theta^*)| \\ &= \|\phi\|_{\Sigma^{-1}} \|\Phi(\hat{\theta}_{LS} - \theta^*)\|_2 \end{aligned}$$

For $\|\phi\|_{\Sigma^{-1}}$, we use a simple setting to bound it.

Assume that $\phi_1 = e_1, \phi_2 = e_2, \ldots, \phi_d = e_d, \phi_{d+1} = e_1, \ldots,$ where $\{e_1, e_2, \ldots, e_d\}$ is an orthonormal basis of $\mathbb{R}^d$, then

$$\begin{aligned} \Sigma &= \frac{n}{d} I \\ \phi^T \Sigma^{-1} \phi &\leq \frac{d}{n} \|\phi\|_2^2 \\ \|\phi\|_{\Sigma^{-1}} &\leq \sqrt{\frac{d}{n}} \|\phi\|_2 \leq \sqrt{\frac{d}{n}} \end{aligned}$$

For $\|\Phi(\hat{\theta}_{LS} - \theta^*)\|_2 = \sqrt{nMSE(\Phi\hat{\theta}_{LS})}$, where $MSE(\Phi\hat{\theta}_{LS}) = \frac{1}{n}\|\Phi(\hat{\theta}_{LS} - \theta^*)\|_2^2$, we have a theorem to bound $MSE(\Phi\hat{\theta}_{LS})$.

Theorem 4.1 *Suppose that the zero-mean noise ϵ is σ-subgaussian, then*

$$\mathbb{E}[MSE(\Phi\hat{\theta}_{LS})] \precsim \sigma^2 \frac{d}{n}$$

and w.p. at least $1-\delta$,

$$MSE(\Phi\hat{\theta}_{LS}) \precsim \sigma^2 \frac{d+\log 1/\delta}{n}$$

Therefore, $\forall\phi, |\phi^T\hat{\theta}_{LS} - \phi^T\theta^*|$ is bounded, and we can say $\hat{\theta}_{LS}$ is a good estimator.

4.3 Model based linear MDP

With the knowledge of the previous section, we can solve a special case of linear MDP problem.

Setting: for a MDP $\mathcal{M} = (S, A, P, r, \gamma)$, the transition function P could be written as

$$P(s'|s,a) = \phi(s,a)^T\mu(s')$$

where $\phi : S \times A \to \mathbb{R}^d$ is known and $\mu : S \to \mathbb{R}^d$ is unknown. What's more, there exists a Barycentric spanner $\Omega = \{(s_1,a_1),(s_2,a_2),\ldots,(s_{d'},a_{d'})\}$ such that $\forall(s,a) \in S \times A$,

$$\|\phi(s,a)\|_{(\Phi^T\Phi)^{-1}} \leq Ld$$

where $\Phi = \begin{bmatrix} \phi(s_1,a_1)^T \\ \phi(s_2,a_2)^T \\ \vdots \\ \phi(s_{d'},a_{d'})^T \end{bmatrix}$ and L is a constant.

With this setting, we can obtain a near-optimal value by using approximate value iteration.

$\forall(s,a) \in S \times A, V \in \mathbb{R}^S$,

$$\begin{aligned} P(\cdot|s,a)V &= \sum_{s'} P(s'|s,a)V(s') \\ &= \sum_{s'} \phi(s,a)^T\mu(s')V(s') \\ &= \phi(s,a)^T \sum_{s'} \mu(s')V(s') \end{aligned}$$

Let θ_V denote $\sum_{s'} \mu(s')V(s')$, then $P(\cdot|s,a)V = \phi(s,a)^T\theta_V$. Then for value iteration, the key problem is to estimate θ_V.

Here we use the generative model and the algorithm is as follow:

Algorithm 9: Approximate value iteration with the generative model

Initialization: $V^0 \leftarrow 0$
for $t = 1, 2, \ldots$ **do**
  for each $(s_i,a_i) \in \Omega$ we query m samples $s_i^{(1)}, s_i^{(2)}, \ldots, s_i^{(m)}$ from $P(\cdot|s_i,a_i)$
  Estimate $\theta_{V^{t-1}}$**:** $\theta_{V^{t-1}} = argmin_\theta \sum_i \sum_{j=1}^m (\phi(s_i,a_i)^T\theta - V^{t-1}(s_i^{(j)}))$
  Iteration: $\forall s \in S$, $V^t(s) = \max_a(r(s,a) + \gamma\phi(s,a)^T\theta_{V^{t-1}})$
end

The error of this approximate method comes from $\phi(s,a)^T\theta_{V^{t-1}}$. With the knowledge from the previous section we have $\forall\phi(s,a)$

$$\begin{aligned}|\phi(s,a)^T\theta_V - P(\cdot|s,a)V| &= |\phi(s,a)^T\theta_V - \phi(s,a)^T\theta_V^*| \\ &= \|\phi(s,a)\|_{(m\Sigma)^{-1}}\sqrt{md'MSE} \\ &\leq \frac{Ld}{\sqrt{m}}\frac{1}{1-\gamma}\sqrt{d+\log 1/\delta}\end{aligned}$$

The error will decrease when the number of samples increases.

4.4 Online RL: linear bandit

Starting from Section 2.2: With the online RL setting, we still want to estimate θ^* using the history information. However, because of the online RL setting, we are in face of uncertainty. Therefore, instead of using the estimator $\hat{\theta}$ directly, we form an uncertainty region, and we call this "optimism in the face of uncertainty"(OFU).

Setting:

1. The reward is defined by,
$$\mathbb{E}[r_t|x_t = x] = \mu^* x \in [-1, 1]$$
where x is the state and μ^* is unknown.

2. The noise sequence,
$$\eta_t = r_t - \mu^* x_t$$
is a martingale difference sequence.

4.4.1 LinUCB algorithm & theorem of regret bound

The LinUCB algorithm is as follow,

Algorithm 10: LinUCB algorithm

Input: λ, β_t.
for $t = 0, 1, \ldots$ **do**
  $$x_t = argmax_x \max_{\mu \in BALL_t} \mu x$$
  where $BALL_t$ is the confidence set defined by λ, β_t and the history information.
  Observe the reward r_t for x_t in order to update the confidence set.
end

Theorem 4.2 *If:*

1. *$BALL_t$ is defined by,*
$$BALL_t := \{\mu | \|\mu - \hat{\mu}_t\|^2_{\Sigma_t} \leq \beta_t\}$$
where $\Sigma_t = \lambda I + \sum_{\tau=0}^{t-1} x_\tau x_\tau^T$ and $\hat{\mu}_t = \Sigma_t^{-1} \sum_{\tau=0}^{t-1} r_\tau x_\tau$;

2. *β_t is defined by,*
$$\beta_t := \sigma^2(2 + 4d \log(1+t) + 8 \log(4/\delta))$$
where σ means that η_t is σ^2 sub-Gaussian;

3. *$\lambda = \sigma^2$*

Then w.p. at least $1 - \delta$,
$$R_T \leq c\sigma\sqrt{T}(d \log(1 + T/\sigma^2) + \log(4/\delta))$$
where c is a constant.

To prove the theorem, we need to prove in two steps. First, we need to prove that the setting of the confidence sets is reasonable, i.e., w.h.p. $\forall t, \mu^* \in BALL_t$; second, under the assumption that w.h.p. $\forall t, \mu^* \in BALL_t$, we need to prove the bound of the regret.

4.4.2 Sum of squares regret bound

Theorem 4.3 *Suppose that $\forall x, \|x\| \leq B$, β_t is increasing and $\beta_t \geq 1$. If $\forall t, \mu^* \in BALL_t$, then*
$$\sum_{t=0}^{T-1} regret_t^2 \leq 8\beta_T d \log(1 + \frac{TB^2}{d\lambda})$$

With **Theorem 2.3**, we can bound R_T by Cauchy-Schwarz inequality,
$$R_T = \sum_{t=0}^{T-1} regret_t \leq \sqrt{T \sum_{t=0}^{T-1} regret_t^2}$$

Since the regret is defined by,
$$regret_t = \mu^* x^* - \mu^* x_t$$

then we can first bound $\mu^* x^*$ by $\mu_t x_t$ where x_t is chosen by the algorithm and μ_t is the corresponding μ (similar to the previous Online RL section, we call this process "optimism"), and then use the center of the $BALL_t$ $\hat{\mu}_t$ as the intermediate state, and then bound them using the property of the ellipsoid.
$$regret_t = \mu^* x^* - \mu^* x_t \leq \mu_t x_t - \mu^* x_t = (\mu_t - \hat{\mu}_t)x_t + (\hat{\mu}_t - \mu^*)x_t \leq 2(somebound)$$

With the knowledge of the ellipsoid, we can derive that $\forall \mu \in BALL_t$,
$$|(\mu - \hat{\mu}_t)x_t| \leq \sqrt{\beta_t} w_t \leq \sqrt{\beta_T} w_t$$

where $w_t = \sqrt{x_t^T \Sigma^{-1} x_t} = \|x_t\|_{\Sigma^{-1}}$.

Since $regret_t \leq 2$ and $\beta_t \geq 1$, we can set that $w_t \leq 1$. Therefore, we use $\log(1+y) \geq y/2$ for $0 < y < 1$ to bound w_t.

$$\sum_{t=0}^{T-1} regret_t^2 \leq 4\beta_T \sum_{t=0}^{T-1} w_t^2 \leq 8\beta_T \sum_{t=0}^{T-1} \log(1+w_t^2)$$

Therefore, our goal is to bound $\prod_{t=0}^{T-1}(1+w_t^2)$. With the knowledge of eigenvalue and by induction, we have

$$det\Sigma_T = det\Sigma_0 \prod_{t=0}^{T-1}(1+w_t^2)$$

For $\Sigma_T = \lambda I + \sum_{t=0}^{T-1} x_t x_t^T$ and $\Sigma_0 = \lambda I$, we have

$$\log(det\Sigma_T / det\Sigma_0) = \log det(I + \frac{1}{\lambda}\sum_{t=0}^{T-1} x_t x_t^T)$$

Assume that the eigenvalues of $\sum_{t=0}^{T-1} x_t x_t^T$ is $\sigma_1, \sigma_2, \ldots, \sigma_d$, then

$$\log det(I + \frac{1}{\lambda}\sum_{t=0}^{T-1} x_t x_t^T) = \log \prod_{i=1}^{d}(1+\frac{\sigma_i}{\lambda}) = d\log \prod_{i=1}^{d}(1+\frac{\sigma_i}{\lambda})^{1/d} \leq d\log(1/d\sum_{i=1}^{d}(1+\frac{\sigma_i}{\lambda}))$$

Since $\sum_{i=1}^{d}\sigma_i$ is the trace of $\sum_{t=0}^{T-1} x_t x_t^T$, so

$$\sum_{i=1}^{d}\sigma_i = Trace(\sum_{t=0}^{T-1} x_t x_t^T) = \sum_{t=0}^{T-1}\|x_t\|^2 \leq TB^2$$

With all above we can bound the regret, with the assumption that $\forall t, \mu^* \in BALL_t$. Next we will prove that this assumption happens with high probability.

4.4.3 Confidence set

The proof of the confidence set is mainly about a Lemma in the original paper of the LinUCB algorithm.

Lemma A.9 (Self-Normalized Bound for Vector-Valued Martingales; [Abbasi-Yadkori et al., 2011]). *Let $\{\varepsilon_i\}_{i=1}^{\infty}$ be a real-valued stochastic process with corresponding filtration $\{\mathcal{F}_i\}_{i=1}^{\infty}$ such that ε_i is $\mathcal{F}_i$ measurable, $\mathbb{E}[\varepsilon_i|\mathcal{F}_{i-1}] = 0$, and ε_i is conditionally σ-sub-Gaussian with $\sigma \in \mathbb{R}^+$. Let $\{X_i\}_{i=1}^{\infty}$ be a stochastic process with $X_i \in \mathcal{H}$ (some Hilbert space) and X_i being $\mathcal{F}_t$ measurable. Assume that a linear operator $\Sigma : \mathcal{H} \to \mathcal{H}$ is positive definite, i.e., $x^\top \Sigma x > 0$ for any $x \in \mathcal{H}$. For any t, define the linear operator $\Sigma_t = \Sigma_0 + \sum_{i=1}^{t} X_i X_i^\top$ (here $xx^\top$ denotes outer-product in $\mathcal{H}$). With probability at least $1-\delta$, we have for all $t \geq 1$:*

$$\left\|\sum_{i=1}^{t} X_i \varepsilon_i\right\|^2_{\Sigma_t^{-1}} \leq \sigma^2 \log\left(\frac{\det(\Sigma_t)\det(\Sigma_0)^{-1}}{\delta^2}\right).$$

Therefore, the key of the proof is to derive $\| \sum_{\tau=0}^{t-1} x_\tau \eta_t \|_{\Sigma_t^{-1}}$ from $\hat{\mu}_t - \mu^*$.

$$\begin{aligned}
\hat{\mu}_t - \mu^* &= \Sigma_t^{-1} \sum_{\tau=0}^{t-1} r_\tau x_\tau - \mu^* \\
&= \Sigma_t^{-1} \sum_{\tau=0}^{t-1} (\mu^* x_\tau^T + \eta_\tau) x_\tau - \mu^* \\
&= \mu^* \Sigma_t^{-1} (\Sigma_t - \lambda I) - \mu^* \Sigma_t^{-1} \Sigma_t + \Sigma_t^{-1} \sum_{\tau=0}^{t-1} \eta_\tau x_\tau \\
&= -\lambda \Sigma_t^{-1} \mu^* + \Sigma_t^{-1} \sum_{\tau=0}^{t-1} \eta_\tau x_\tau
\end{aligned}$$

With this we can prove that $\forall t, \mu^* \in BALL_t$ will happen w.h.p.

4.5 Online RL: Linear MDP

In this section we will introduce the UCB value iteration algorithm for the linear MDP setting. In section 3.3.3 we omit the detail of the proof. The algorithm with linear MDP setting is a more difficult one and we will introduce more detail here.

4.5.1 Setting

Consider a finite horizontal MDP with the probability transition function $\{P_h\}$ and the reward $\{r_h\}$ defined as,

$$\begin{aligned}
r_h(s, a) &= \theta_h^* \phi(s, a) \\
P_h(\cdot | s, a) &= \mu_h^* \phi(s, a)
\end{aligned}$$

where $\{\theta_h^*\}, \phi$ are known and $\{\mu_h^*\}$ is unknown.

With online RL, the basic method is at each episode, to estimate μ_h^* using the history information, set a reasonable upper bound of the estimator (usually achieved by set a bound to the reward), and then use the upper bound to do dynamical programming.

4.5.2 Dynamical programming

For the finite MDP, when doing the value iteration, in each episode we do dynamical programming. Given the setting from 4.5.1, and assume that μ_h^* is known, the dynamical programming is as follows,

4.5.3 Estimation

Now consider that μ_h^* is unknown and we need to estimate it.

Before the n-th episode, we already have the dataset of the previous episodes

$$D_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=0}^{n-1}$$

Algorithm 11: Dynamical programming of finite MDP

Initialization: $V_H = 0$
for $h = H-1, H-2, \ldots, 0$ **do**
  $Q_h(s,a) = \phi(s,a)w_h$ where $w_h = \theta_h^* + \mu_h^* V_{h+1}$
  $V_h(s) = \max_a Q_h(s,a)$
  $\pi_h(s) = argmax_a Q_h(s,a)$
end

Then we can use the estimator:

$$\hat{\mu}_h^n = \sum_{i=0}^{n-1} \delta s_{h+1}^i \phi(s_h^i, a_h^i)(\Lambda_h^n)^{-1}$$

where $\Lambda_h^n = \sum_{i=0}^{n-1} \phi(s_h^i, a_h^i)\phi(s_h^i, a_h^i)^T + \lambda I$, which is very similar to the estimator in the linear bandit problem.

It can be proven that

$$\hat{\mu}_h^n - \mu_h^* = \lambda \mu_h^* (\Lambda_h^n)^{-1} + \sum_{i=1}^{n-1} \epsilon_h^i \phi(s_h^i, a_h^i)^T (\Lambda_h^n)^{-1}$$

where $\epsilon_h^i = P(\cdot|, s_h^i, a_h^i) - \delta(s_{h+1}^i)$.

Then for arbitrary V, it can be proven that the error between the estimated Q-value and the actual Q-value is

$$|\phi(s,a)^T(\hat{\mu}_h^n - \mu_h^n)^T V| \leq |\lambda \phi(s,a)^T (\Lambda_h^n)^{-1} (\mu_h^*)^T V| + |\sum_{i=1}^{n-1} \phi(s,a)^T (\Lambda_h^n)^{-1} \phi(s_h^i, a_h^i)(\epsilon_h^i)^T V|$$

4.5.4 Upper bound of the error

The first term of the error $|\lambda \phi(s,a)^T (\Lambda_h^n)^{-1} (\mu_h^*)^T V|$ is easy to deal with.

$$|\lambda \phi(s,a)^T (\Lambda_h^n)^{-1} (\mu_h^*)^T V| \leq \|\phi(s,a)\|_{(\Lambda_h^n)^{-1}} \lambda \|(\mu_h^*)^T V\|_{(\Lambda_h^n)^{-1}}$$

With the setting that $\forall v$ such that $\|v\|_\infty \leq 1$, $\|(\mu_h^*)^T v\| \leq \sqrt{d}$, we have

$$|\lambda \phi(s,a)^T (\Lambda_h^n)^{-1} (\mu_h^*)^T V| \leq H\sqrt{d} \|\phi(s,a)\|_{(\Lambda_h^n)^{-1}}$$

But the second term $|\sum_{i=1}^{n-1} \phi(s,a)^T (\Lambda_h^n)^{-1} \phi(s_h^i, a_h^i)(\epsilon_h^i)^T V|$ is difficult. First we have,

$$|\sum_{i=1}^{n-1} \phi(s,a)^T (\Lambda_h^n)^{-1} \phi(s_h^i, a_h^i)(\epsilon_h^i)^T V| \leq \|\phi(s,a)\|_{(\Lambda_h^n)^{-1}} \|\sum_{i=1}^{n-1} \phi(s_h^i, a_h^i)(\epsilon_h^i)^T V\|_{(\Lambda_h^n)^{-1}}$$

Using the self-normalized vector-valued martingale bound which we mentioned in 4.4.3, we have for a fix V, w.p. at least $1-\delta$,

$$\|\sum_{i=1}^{n-1} \phi(s_h^i, a_h^i)(\epsilon_h^i)^T V\|_{(\Lambda_h^n)^{-1}}^2 \leq 9H^2 \log \frac{H(1+N)^d}{\delta}$$

But the problem is there are infinite number of V and the union bound doesn't work here! Therefore, we will find the ϵ-cover Net $\mathcal{N}_\epsilon$ first, and then compute the bound.

The ϵ-cover Net $\mathcal{N}_\epsilon$ contains finite number of V such that $\forall V$, $\exists V' \in \mathcal{N}_\epsilon$ such that $\|V - V'\|_2 \leq \epsilon$.

Now let's focus on V. Since in the algorithm we will introduce next, we compute V with a bonus reward like $\beta\sqrt{\phi(s,a)^T\Lambda^{-1}\phi(s,a)}$, then the function class of V is,

$$\mathcal{F} = \{f | f = \max_a(w^T\phi(s,a) + \beta\sqrt{\phi(s,a)^T\Lambda^{-1}\phi(s,a)})\}$$

With the setting $\|w\|_2 \leq L$, $\beta \in [0, B]$ and $\sigma_{min}(\Lambda) \geq \lambda$, the scale of its $\mathcal{N}_\epsilon$ is

$$|\mathcal{N}_\epsilon| \leq (1 + \frac{6L}{\epsilon})^d(1 + \frac{6B}{\sqrt{\lambda}\epsilon})(1 + \frac{18B^2\sqrt{d}}{\lambda\epsilon^2})^{d^2}$$

Therefore, $\forall V \in \mathcal{N}_\epsilon$ we can use the union bound, w.p. at least $1 - \delta$,

$$\|\sum_{i=1}^{n-1}\phi(s_h^i, a_h^i)(\epsilon_h^i)^T V\|^2_{(\Lambda_h^n)^{-1}} \leq 9H^2 \log\frac{|\mathcal{N}_\epsilon|H(1+N)^d}{\delta}$$

$\forall V \in \mathcal{F}$, there exists $V' \in \mathcal{N}_\epsilon$ such that $\|V - V'\|_2 \leq \epsilon$, then

$$\begin{aligned}
\|\sum_{i=1}^{n-1}\phi(s_h^i, a_h^i)(\epsilon_h^i)^T V\|^2_{(\Lambda_h^n)^{-1}} &\leq 2\|\sum_{i=1}^{n-1}\phi(s_h^i, a_h^i)(\epsilon_h^i)^T V'\|^2_{(\Lambda_h^n)^{-1}} + 2\|\sum_{i=1}^{n-1}\phi(s_h^i, a_h^i)(\epsilon_h^i)^T (V - V')\|^2_{(\Lambda_h^n)^{-1}} \\
&\leq 2\|\sum_{i=1}^{n-1}\phi(s_h^i, a_h^i)(\epsilon_h^i)^T V'\|^2_{(\Lambda_h^n)^{-1}} + 2\|\sum_{i=1}^{n-1}\phi(s_h^i, a_h^i)2\epsilon\|^2_{(\Lambda_h^n)^{-1}} \\
&\leq 2\|\sum_{i=1}^{n-1}\phi(s_h^i, a_h^i)(\epsilon_h^i)^T V'\|^2_{(\Lambda_h^n)^{-1}} + 8\epsilon^2 N
\end{aligned}$$

With all above we can derive the upper bound of the error.

4.5.5 Algorithm

The UCB value iteration algorithm is as follows,

Algorithm 12: UCBVI for linear MDP

```
for n = 1, 2, ..., N do
    Compute P̂_h^n
    Compute reward bonus b_h^n by b_h^n(s,a) = β‖φ(s,a)‖_{(Λ_h^n)^{-1}}
    Use {P̂_h^n, r_h + b_h^n} to run dynamical programming
    Obtain the policy π^n
end
```

To bound the regret of this algorithm, we should,

1. *Prove optimism:* denote $\beta = \tilde{O}(Hd)$ and prove that w.h.p. $\hat{V}_h^n \geq V_h^*$;
2. *Regret decomposition:* bound $Regret = \sum_{n=0}^{N-1}(V^* - V^{\pi_n})$ by $Regret \leq \sum_{n=0}^{N-1}(\hat{V}_h^n - V^{\pi_n})$, and continuously bound it by the reward bonus,

$$Regret \precsim \sum_{n=0}^{N-1}\sum_{h=0}^{H-1} b_h^n(s_h^n, a_h^n)$$

3. *Final regret bound:*

$$\begin{aligned}
\sum_{n=0}^{N-1}\sum_{h=0}^{H-1} b_h^n(s_h^n, a_h^n) &= \beta \sum_{n=0}^{N-1}\sum_{h=0}^{H-1} \|\phi(s,a)\|_{(\Lambda_h^n)^{-1}} \\
&\leq \beta \sum_{h=0}^{H-1} \sqrt{N \sum_{n=0}^{N-1} \|\phi(s,a)\|^2_{(\Lambda_h^n)^{-1}}} \\
&\leq \beta \sum_{h=0}^{H-1} \sqrt{N \sum_{n=0}^{N-1} \frac{d}{n}} \\
&\precsim \beta H \sqrt{Nd\log(N)}
\end{aligned}$$

Plug in that $\beta = \tilde{O}(Hd)$, we derive that

$$Regret \leq \tilde{O}(H^2\sqrt{N^3 d})$$

5 Special topic: bandit problem with general function approximation

5.1 Bandit setting

With online RL setting, in each episode t, the agent pulls an arm a_t and receives a reward

$$r_t = f^*(a) + \eta_t$$

where f^* is unknown and η_t is independent to the history and is η-sub Gaussian.

The goal is to minimize the regret

$$R(T) = \sum_{t=1}^{T}(f^*(a^*) - f^*(a_t))$$

where $a^* = argmax_a f^*(a)$.

Suppose that $f^* \in G$ and $\forall f \in G$, $\|f\|_\infty \leq c$. Similar to the linear bandit problem, the key is to establish the confidence sets of function f.

Algorithm 13: UCB general function bandit algorithm

Initialize: $G_1 = G$

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

$a_t = argmax_a \sup_{f \in G_t} f(a)$

$G_{t+1} = \{f | \|f - \hat{f}_t\|_{D_t}^2 \leq \beta_t\}$ where

$$\|g\|_{D_t}^2 = \sum_{i=1}^{t} g(a_i)^2$$

and

$$\hat{f}_t = argmin_{f \in G} L_{2,t} f := \sum_{i=1}^{t} (f(a_i) - r_i)^2$$

and β_t is some parameters.

end

5.2 UCB algorithm

The UCB algorithm is shown in Algorithm 13.

5.3 Regret analysis

Without proving, we give the bound of $\|f - \hat{f}_t\|_{D_t}^2$ directly.

Lemma 5.1 *W.p. at least* $1 - \delta$,

$$\|f - \hat{f}_t\|_{D_t}^2 \precsim \eta^2 \log(\frac{N_\alpha}{\delta}) + \alpha t + \alpha t \sqrt{\eta^2 \log(\frac{N_\alpha}{\delta})}$$

where N_α *is the* α*-covering number of* G.

With this Lemma, we can define β_t.

w.h.p. the regret is bounded by

$$R(T) \leq \sum_{t=1}^{T} w_{G_t}(a_t)$$

where $w_G(a) := \sup_{f_1, f_2}(f_1(a) - f_2(a))$.

Without proving, we give a proposition of $\{w_{G_t}(a_t)\}_{t=1}^T$.

Proposition 5.2

$$\sum_{t=1}^{T} 1(w_{G_t}(a_t) > \epsilon) \leq (\frac{4\beta_T}{\epsilon^2} + 1) dim_E(G, \epsilon)$$

where $dim_E(G, \epsilon)$ *is the Eluder dimension of* G.

Therefore, the regret is bounded by,

$$\begin{aligned} R(T) &\leq \sum_{t=1}^{T} w_{G_t}(a_t) \\ &\leq \epsilon T + \sum_{t=1}^{T} 1(w_{G_t}(a_t) > \epsilon) w_{G_t}(a_t) \\ &\precsim \epsilon T + \int_{\epsilon}^{c} \frac{4\beta_T}{\epsilon'^2} dim_E(G, \epsilon') d\epsilon' \\ &\leq \epsilon T + (\frac{1}{\epsilon} - \frac{1}{c}) 4\beta_T dim_E(G, \epsilon) \end{aligned}$$

when $\epsilon = \tilde{O}(\sqrt{\frac{\beta_T dim_E(G)}{T}})$ the bound reaches the maximum, and we finally have,

$$R(T) \precsim \sqrt{\beta_T dim_E(G) T}$$