CS 839 Probability and Learning in High Dimension Lecture 23 - 04/25/2022

Lecture 23: Linear MDPs III

Lecturer: Yudong Chen *Scribe: Peyman Morteza*

In this lecture,[1] we complete the proof of regret bound for LSVI-UCB and also give a preview of Randomized Numerical Linear Algebra (RandNLA).

1 Recap: LSVI-UCB Algorithm and Regret Bound

Recall the LSVI-UCB algorithm (Algorithm 1) and its regret guarantee (Theorem 1) from previous lecture.

Algorithm 1 LSVI-UCB

```
for episode $k = 1, 2, \dots, K$ do
    1. (Value estimation) for step $h = H, H-1, \dots, 1$ do
        (a) (Gram matrix) $\Lambda_h^k \leftarrow \sum_{\tau \in [k-1]} \phi(x_h^\tau, a_h^\tau)\phi(x_h^\tau, a_h^\tau)^\top + I$
        (b) (Least squares)
            $w_h^k \leftarrow \arg\min_{w \in \mathbb{R}^d} \sum_{\tau \in [k-1]} \left[ r_h(x_h^\tau, a_h^\tau) + V_{h+1}^k(x_{h+1}^\tau) - \langle w, \phi(x_h^\tau, a_h^\tau) \rangle \right]^2 + \|w\|^2$
            $= (\Lambda_h^k)^{-1} \sum_{\tau \in [k-1]} \phi(x_h^\tau, a_h^\tau) \cdot \left[ r_h(x_h^\tau, a_h^\tau) + V_{h+1}^k(x_{h+1}^\tau) ) \right].$
        (c) (Q estimate with UCB) $Q_h^k(\cdot,\cdot) = \left\langle w_h^k, \phi(\cdot,\cdot) \right\rangle + \beta\sqrt{\phi(\cdot,\cdot)^\top (\Lambda_h^k)^{-1} \phi(\cdot,\cdot)}$
        (d) (From Q to value function) $V_h^k(\cdot) = \max_a Q_h^k(\cdot, a)$.
    2. Receive initial state $x_1^k$
    3. (Policy execution) for step $h = 1, 2, \dots, H$ do
        Take action $a_h^k \leftarrow \arg\max_a Q_h^k(x_h^k, a)$; observe reward $r_h^k = r_h(x_h^k, a_h^k)$ and next state $x_{h+1}^k$.
```

In the following, $T := KH$ is the total number of steps over all episodes.

Theorem 1. *Set $\beta = cdH\sqrt{\iota}$ with $\iota := \log(2dT/p)$. With probability at least $1-p$, we have*

$$\mathrm{Regret}(K) := \sum_{k=1}^{K} \left[V_1^*(x_1^k) - V_1^{\pi^k}(x_1^k) \right] \lesssim \sqrt{d^3 H^3 T \iota^2} = \sqrt{d^3 H^4 K \iota^2}.$$

1.1 Remarks on Model and Algorithms

Before we continue with the proof of Theorem 1, we provide some remark that highlights the insights underlying the above results.

Remark 1 (Linear Models). A key step and contribution of the above result is identifying the appropriate linear model for MDPs. One might start by considering the following linear assumption on the transition kernel,

$$\mathbb{P}(\cdot|s,a) = \sum_{i=1}^{d} \beta_i \mathcal{K}_i(\cdot|s,a),$$

[1] *Reading:* [Jin et al., 2019]

where the coefficients β_i's are unknown, and the kernel $\mathcal{K}_i$ maps state action pairs to signed measures on $\mathcal{S}$ and is assumed to be known. This model is known as the "linear mixture MDP" model. In comparison, the "linear MDP" model we have been considering so far assumes that

$$\mathbb{P}(\cdot|s,a) = \sum_{i=1}^{d} \phi^i(s,a)\mu^i(\cdot),$$

where in the above μ^i's are signed measures on $\mathcal{S}$ and assumed to be unknown and the feature map ϕ is assumed to be known. We see that this model is substantially more flexible than the linear mixture MDP model.

Remark 2 (Bonus in LSVI-UCB). Another key point about LSVI-UCB algorithm is where the bonus term is added. In Algorithm 1, a bonus term $\beta\sqrt{\phi^T(\Lambda_h^k)^{-1}\phi}$, is added in Step 1(c) when computing the Q_h^k function, which is in turn used to compute the V_h^k function. Note that V_h^k (as well as the bonus term it carries) is then used in computing Q_{h-1}^k and V_{h-1}^k for step $h-1$. Therefore, the bonus accumulates across the steps h. It is tempting to consider the following alternative algorithm (which does NOT work).

```
Algorithm 2 An Alternative Algorithm That Does NOT Work
for episode $k = 1, 2, \dots, K$ do
    1. (Value estimation) for step $h = H, H-1, \dots, 1$ do
        (a) (Gram matrix) $\Lambda_h^k \leftarrow \sum_{\tau\in[k-1]} \phi(x_h^\tau, a_h^\tau)\phi(x_h^\tau, a_h^\tau)^\top + I$
        (b) (Least squares)
            $w_h^k = (\Lambda_h^k)^{-1} \sum_{\tau\in[k-1]} \phi(x_h^\tau, a_h^\tau) \cdot \left[r_h(x_h^\tau, a_h^\tau) + V_{h+1}^k(x_{h+1}^\tau))\right].$
        (c) (Q estimate) $Q_h^k(\cdot,\cdot) = \left\langle w_h^k, \phi(\cdot,\cdot)\right\rangle$
        (d) (From Q to value function) $V_h^k(\cdot) = \max_a Q_h^k(\cdot, a)$.
    2. Receive initial state $x_1^k$
    3. (Policy execution with bonus) for step $h = 1, 2, \dots, H$ do
        Take action $a_h^k \leftarrow \arg\max_a \left\{Q_h^k(x_h^k, a) + \beta\sqrt{\phi(x_h^k, a)^\top(\Lambda_h^k)^{-1}\phi(x_h^k, a)}\right\}$
        Observe reward $r_h^k = r_h(x_h^k, a_h^k)$ and next state $x_{h+1}^k$.
```

In Algorithm 2, the bonus is added in step 3 when choosing actions. However, this would result in insufficient exploration, because it does not consider accumulating the bonus terms.

2 Completing the Proof of Theorem 1

As discussed in the previous lecture, the proof can be divided into 5 main steps:

1. Concentration
2. LS estimation error
3. UCB
4. Regret Decomposition
5. Regret Bound

We completed step 1 in the previous lecture. In this lecture we will complete the remaining steps.

Define the shorthand $\phi_h^\tau := \phi(x_h^\tau, a_h^\tau)$.

2.1 Concentration

We recall the following lemma.

Lemma 3 (Concentration of empirical measure)**.** *For each p, the following event $\mathfrak{E}$ holds with probability at least $1 - p/2$:*

$$\left\| \sum_{\tau \in [k-1]} \phi_h^\tau \left[V_{h+1}^k(x_{h+1}^\tau) - (\mathbb{P}_h V_{h+1}^k)(x_h^\tau, a_h^\tau) \right] \right\|_{(\Lambda_h^k)^{-1}} \lesssim dH\sqrt{\log(dT/p)}, \qquad \forall k, h.$$

The proof was given in the previous lecture. We provide a remark that highlights intuition for this lemma.

Remark 4. Roughly speaking, this lemma says that the empirical estimate $\sum_\tau V(x_{h+1}^\tau)$ approximates the true expectation $\sum_\tau (\mathbb{P}_h V)(x_h^\tau, a_h^\tau)$. In fact, the lemma says something more:

- Note the factor $\phi_h^\tau := \phi(x_h^\tau, a_h^\tau)$ that appears in the lemma. This means we have good concentration not only for expected value $(\mathbb{P}_h V)(x_h^\tau, a_h^\tau)$ associated with the visited state-action pair (x_h^τ, a_h^τ), but also for the direction spanned by the feature vector ϕ_h^τ of this pair. Therefore, the empirical estimate *generalizes* to other (unseen) state-action pairs whose feature vectors are similar to ϕ_h^τ.
- The approximation error is weighted by the Gram matrix $\Lambda_h^k := I + \sum_{\tau \in [k-1]} \phi_h^\tau (\phi_h^\tau)^\top$, where we recall that $\phi_h^\tau = \phi(x_h^\tau, a_h^\tau)$ are feature vectors of the previous visited state-action pairs (x_h^τ, a_h^τ). Therefore, we have better approximation in the directions that are better covered by the previous data.

2.2 Least-squares estimation error

We next bound the difference between the algorithm's value function (without bonus) and the true value function of any policy π (including π^*), recursively in terms of the step h.

Lemma 5 (Least-squares error bound)**.** *If $\beta = dH\sqrt{\iota}$,then on the event $\mathfrak{E}$ in Lemma 3, we have for all (x, a, h, k, π):*

$$\left\langle \phi(x,a), w_h^k \right\rangle - Q_h^\pi(x,a) = \mathbb{P}_h (V_{h+1}^k - V_{h+1}^\pi)(x,a) + \Delta_h^k(x,a),$$

where

$$\left| \Delta_h^k(x,a) \right| \le \beta \sqrt{\phi(x,a)^\top \left(\Lambda_h^k\right)^{-1} \phi(x,a)}.$$

Proof By linearity and Bellman equation we have $Q_h^\pi(x,a) = \langle \phi(x,a), w_h^\pi \rangle = r(x,a) + (\mathbb{P}_h V_{h+1}^\pi)(x,a)$ and by algorithm specification we have $w_h^k = (\Lambda_h^k)^{-1} \sum_{\tau \in [k-1]} \phi_h^\tau \cdot \left[r_h(x_h^\tau, a_h^\tau) + V_{h+1}^k(x_{h+1}^\tau) \right]$. We want to bound the difference in the weights, $w_h^k - w_h^\pi$. We first use linearity to express w_h^π as also a least-square solution:

$$\begin{aligned} w_h^\pi &= \left(\sum_{\tau \in [k-1]} \phi_h^\tau \right)^{-1} \sum_{\tau \in [k-1]} \phi_h^\tau \cdot \left[r_h(x_h^\tau, a_h^\tau) + (\mathbb{P}_h V_{h+1}^\pi)(x_h^\tau, a_h^\tau) \right] \\ &= \left(\Lambda_h^k - I \right)^{-1} \sum_{\tau \in [k-1]} \phi_h^\tau \cdot \left[r_h(x_h^\tau, a_h^\tau) + (\mathbb{P}_h V_{h+1}^\pi)(x_h^\tau, a_h^\tau) \right] \end{aligned}$$

Multiplying both sides by $\left(\Lambda_h^k\right)^{-1} \left(\Lambda_h^k - I\right)$ gives

$$w_h^\pi - \left(\Lambda_h^k\right)^{-1} w_h^\pi = \left(\Lambda_h^k\right)^{-1} \sum_{\tau \in [k-1]} \phi_h^\tau \cdot \left[r_h(x_h^\tau, a_h^\tau) + (\mathbb{P}_h V_{h+1}^\pi)(x_h^\tau, a_h^\tau) \right]$$

It follows that

$$
\begin{aligned}
w_h^k - w_h^\pi &= (\Lambda_h^k)^{-1} \sum_{\tau \in [k-1]} \phi_h^\tau \cdot \left[V_{h+1}^k(x_{h+1}^\tau) - (\mathbb{P}_h V_{h+1}^\pi)(x_h^\tau, a_h^\tau) \right] - \left(\Lambda_h^k\right)^{-1} w_h^\pi \\
&= \underbrace{(\Lambda_h^k)^{-1} \sum_{\tau \in [k-1]} \phi_h^\tau \cdot \left[V_{h+1}^k(x_{h+1}^\tau) - (\mathbb{P}_h V_{h+1}^k)(x_h^\tau, a_h^\tau) \right]}_{q_2} \\
&\quad + \underbrace{(\Lambda_h^k)^{-1} \sum_{\tau \in [k-1]} \phi_h^\tau \cdot \left[\mathbb{P}_h (V_{h+1}^k - V_{h+1}^\pi)(x_h^\tau, a_h^\tau) \right]}_{q_3} - \underbrace{\left(\Lambda_h^k\right)^{-1} w_h^\pi}_{q_1}.
\end{aligned}
$$

whence

$$
\left\langle \phi(x,a), w_h^k \right\rangle - Q_h^\pi(x,a) = \left\langle \phi(x,a), q_1 + q_2 + q_3 \right\rangle.
$$

We apply Cauchy-Schwarz to bound each RHS term:

1. First term: we have

$$
\left\langle \phi(x,a), q_1 \right\rangle \le \| w_h^\pi \|_{(\Lambda_h^k)^{-1}} \cdot \| \phi(x,a) \|_{(\Lambda_h^k)^{-1}} \le \| w_h^\pi \| \cdot \| \phi(x,a) \|_{(\Lambda_h^k)^{-1}} \lesssim H\sqrt{d} \cdot \| \phi(x,a) \|_{(\Lambda_h^k)^{-1}}
$$

   using $\Lambda_h^k \succeq I$ and $\| w_h^\pi \| \lesssim H\sqrt{d}$.

2. Second term: we have

$$
\left\langle \phi(x,a), q_2 \right\rangle \lesssim dH\sqrt{\log(dT/p)} \cdot \| \phi(x,a) \|_{(\Lambda_h^k)^{-1}}
$$

   using the last Lemma 3.

3. Third term: letting $\Phi_h^k = \sum_{\tau \in [k-1]} \phi_h^\tau \phi_h^{\tau \top} = \Lambda_h^k - I$ and observing that $\mathbb{P}(\cdot | x, a) = \phi(x,a)^\top \mu_h(\cdot)$, we have

$$
\begin{aligned}
\left\langle \phi(x,a), q_3 \right\rangle &= \left\langle \phi(x,a), (\Lambda_h^k)^{-1} \cdot \Phi_h^k \cdot \mu_h(\cdot)(V_{h+1}^k - V_{h+1}^\pi) \right\rangle \\
&= \left\langle \phi(x,a), \mu_h(\cdot)(V_{h+1}^k - V_{h+1}^\pi) \right\rangle + \left\langle \phi(x,a), (\Lambda_h^k)^{-1} \mu_h(\cdot)(V_{h+1}^k - V_{h+1}^\pi) \right\rangle \\
&= \mathbb{P}_h(V_{h+1}^k - V_{h+1}^\pi)(x,a) + \left\langle \phi(x,a), (\Lambda_h^k)^{-1} \mu_h(\cdot)(V_{h+1}^k - V_{h+1}^\pi) \right\rangle \\
&\lesssim \mathbb{P}_h(V_{h+1}^k - V_{h+1}^\pi)(x,a) + \| \phi(x,a) \|_{(\Lambda_h^k)^{-1}} \cdot H\sqrt{d/\lambda}
\end{aligned}
$$

   using $\Lambda_h^k \succeq I$,$\| \mu_h(S) \| \le \sqrt{d}$ and $V_{h+1}^k(\cdot) \le H, V_{h+1}^\pi \le H$.

Combining, we can derive the desired bound

$$
\begin{aligned}
\left\langle \phi(x,a), w_h^k \right\rangle - Q_h^\pi(x,a) &\lesssim \mathbb{P}_h(V_{h+1}^k - V_{h+1}^\pi)(x,a) + dH \| \phi(x,a) \|_{(\Lambda_h^k)^{-1}} \\
&\lesssim \mathbb{P}_h(V_{h+1}^k - V_{h+1}^\pi)(x,a) + \beta \| \phi(x,a) \|_{(\Lambda_h^k)^{-1}}
\end{aligned}
$$

under our choice of $\beta = dH\sqrt{\iota}$. □

2.3 UCB property

Next, we establish the desired UCB property, i.e., Q_h^k constructed in the algorithm always upper bounds the true Q function $Q_h^*(x,a)$.

Lemma 6 (UCB). *On the event $\mathfrak{E}$ in Lemma 3, we have $Q_h^k(x,a) \ge Q_h^*(x,a)$ for all (x,a,k,h).*

Proof We fix k and perform induction on h. The base case $h = H$ holds since the terminal cost is zero. For the induction step, we have

$$\begin{aligned} Q_h^k(x,a) &:= \left\langle w_h^k, \phi(x,a) \right\rangle + \beta \underbrace{\sqrt{\phi(x,a)^\top \left(\Lambda_h^k\right)^{-1} \phi(x,a)}}_{\text{bonus}} && \text{by construction} \\ &= Q_h^*(x,a) + \mathbb{P}_h(V_{h+1}^k - V_{h+1}^*)(x,a) + \Delta_h^k(x,a) + \beta \cdot \text{bonus} && \text{Lemma 5 with } \pi{=}\pi^* \\ &\geq Q_h^*(x,a) + 0 + 0. && V_{h+1}^k \geq V_{h+1}^* \text{ by induction} \end{aligned}$$

□

2.4 Regret decomposition

Finally, we have the following recursive bound for $V_h^k(x_h^k) - V_h^{\pi^k}(x_h^k)$, the difference between the UCB value and true values of the agent's policy π^k.

Lemma 7 (Recursive formula)**.** *Let* $\delta_h^k := V_h^k(x_h^k) - V_h^{\pi^k}(x_h^k)$*, and* $\zeta_{h+1}^k := \mathbb{E}\left[\delta_{h+1}^k \mid x_h^k, a_h^k\right] - \delta_{h+1}^k$*. Then on the event* $\mathfrak{E}$ *in Lemma 3, we have for all* (k,h)*,*

$$\underbrace{\delta_h^k}_{\textit{error for step h}} \leq \underbrace{\delta_{h+1}^k}_{\textit{error for step h+1}} + \underbrace{\zeta_{h+1}^k}_{\textit{statistical error}} + 2 \underbrace{\beta\sqrt{(\phi_h^k)^\top (\Lambda_h^k)^{-1} \phi_h^k}}_{\textit{UCB bonus}}.$$

Proof By construction we have

$$\delta_h^k = Q_h^k(x_h^k, a_h^k) - Q_h^{\pi^k}(x_h^k, a_h^k).$$

By Lemma 5 we have for all (x,a),

$$\begin{aligned} Q_h^k(x,a) - Q_h^{\pi^k}(x,a) &\leq \mathbb{P}_h(V_{h+1}^k - V_{h+1}^{\pi^k})(x,a) + \Delta_h^k(x,a) + \underbrace{\beta\sqrt{(\phi_h^k)^\top (\Lambda_h^k)^{-1} \phi_h^k}}_{\text{bonus}} \\ &\leq \mathbb{P}_h(V_{h+1}^k - V_{h+1}^{\pi^k})(x,a) + 2 \cdot \text{bonus} \end{aligned}$$

Combining, we obtain

$$\begin{aligned} \delta_h^k &\leq \mathbb{P}_h(V_{h+1}^k - V_{h+1}^{\pi^k})(x_h^k, a_h^k) + 2 \cdot \text{bonus} \\ &= \mathbb{E}\left[\delta_{h+1}^k \mid x_h^k, a_h^k\right] + 2 \cdot \text{bonus} \\ &= \delta_{h+1}^k + \zeta_{h+1}^k + 2 \cdot \text{bonus} \end{aligned}$$

as desired. □

2.5 Putting together

We are now ready to prove the regret bound $O\left(\sqrt{d^3 H^3 T \iota^2}\right)$ in the main theorem.

First, note that the regret is

$$\begin{aligned}
\text{Regret}(K) &:= \sum_{k=1}^{K} \left[V_1^*(x_1^k) - V_1^{\pi^k}(x_1^k) \right] && \text{definition} \\
&\le \sum_{k=1}^{K} \left[V_1^k(x_1^k) - V_1^{\pi^k}(x_1^k) \right] && V^k \text{ is UC by Lemma 6} \\
&= \sum_{k=1}^{K} \delta_1^k && \text{definition} \\
&\le \sum_{k=1}^{K} \sum_{h=1}^{H} \zeta_h^k + 2\beta \sum_{k=1}^{K} \sum_{h=1}^{H} \sqrt{(\phi_h^k)^\top (\Lambda_h^k)^{-1} \phi_h^k}. && \text{Lemma 7}
\end{aligned}$$

- (This is the concentration part.) For the first term, we know that (ζ_h^k) is a martingale difference sequence (with respect to both h and k), and $\left|\zeta_h^k\right| \le H$. Hence by Azuma-Hoeffding, we have w.h.p.

$$\sum_{k=1}^{K} \sum_{h=1}^{H} \zeta_h^k \lesssim H \cdot \sqrt{KH\iota} = H\sqrt{T\iota}.$$

- (This is the real regret part.) For the second term, we apply the elliptical potential Lemma 10 to obtain

$$\begin{aligned}
\sum_{h=1}^{H} \sum_{k=1}^{K} \sqrt{(\phi_h^k)^\top (\Lambda_h^k)^{-1} \phi_h^k} &\le \sum_{h=1}^{H} \sqrt{K} \sqrt{\sum_{k=1}^{K} (\phi_h^k)^\top (\Lambda_h^k)^{-1} \phi_h^k} && \text{Jensen's or Cauchy-Schwarz} \\
&\le \sum_{h=1}^{H} \sqrt{K} \cdot \sqrt{2 \log \left(\frac{\det \Lambda_h^K}{\det \Lambda_h^0} \right)} && \text{Lemma 10} \\
&\le \sum_{h=1}^{H} \sqrt{K} \cdot \sqrt{2 \log \left(\frac{(1 + k \max_k \left\| \phi_h^k \right\|^2)^d}{1} \right)} && \text{by construction of } \Lambda_h^k \\
&\le \sum_{h=1}^{H} \sqrt{K} \cdot \sqrt{2d \log \left(\frac{1+k}{1} \right)} && \left\| \phi_h^k \right\| \le 1, \forall h, k \text{ by assumption} \\
&\le H\sqrt{2Kd\iota}.
\end{aligned}$$

Combining, we obtain

$$\text{Regret}(K) \lesssim H\sqrt{T\iota} + \beta \cdot H\sqrt{2Kd\iota} \lesssim \sqrt{d^3 H^3 T \iota^2}$$

by our choice of $\beta \asymp dH\sqrt{\iota}$. This completes the proof of the main theorem.

3 Preview: Randomized Numerical Linear Algebra (RandNLA)

In the next few lectures we will discuss Randomized Numerical Linear Algebra (RandNLA). The goal of RandNLA is to design randomized algorithms to obtain approximate solution to common linear algebra problems, hopefully with better computational efficiency than deterministic exact solutions. Probability is used as tool in both algorithm design and analysis.

We have the following three problems in mind:

1. *Matrix Multiplication*
   Given two $n \times n$ matrices A and B, we can compute the product AB with $O(n^3)$ running time. There are sub-cubic algorithms such as *Strassen algorithm* which computes the product with approximately $O(n^{2.8})$ running time; however, such algorithms are less common in practice. Can we design randomized algorithm to approximate AB with efficient computational time? In other words we would like to design algorithms with efficient running time by finding approximation solutions that are good enough for, say, machine learning applications.

2. *Least Squares Problems*
   We would like to solve the following problem:

   $$\min_{x \in \mathbb{R}^d} \|Ax - b\|^2 ,$$

   where $A \in \mathbb{R}^{n \times d}$, $b \in \mathbb{R}^n$ and $n > d$. There are two common ways to solve the above problem. First, noting that the optimal solution satisfies the normal equation

   $$0 = \nabla(\|Ax - b\|^2) \Longleftrightarrow A^\top (Ax - b) = 0,$$

   one can solve this linear equation system using Gaussian elimination-type methods in $O(nd^2)$ time. We could also solve the optimization problem using an iterative algorithm like gradient descent,

   $$x_{t+1} = x_t - \eta \nabla(\|Ax_t - b\|^2) = x_t - \eta A^\top (Ax_t - b).$$

   Here, each update only involves matrix-vector multiplication and can be done in $O(nd)$ time. To obtain a solution with accuracy ϵ, the number of iterations should be roughly proportional to $\kappa(A)\log(\frac{1}{\epsilon})$, where $\kappa(A)$ is the condition number of A. The overall running time would be $O(nd\kappa(A)\log(\frac{1}{\epsilon}))$. Can we improve the computational time and approximate the solution by designing randomized algorithms?

3. *Low-Rank Approximation*
   We would like to find a low-rank matrix that approximates a given matrix. We will discuss this problem in more details in the subsequent lectures.

Appendices

A Technical lemmas

We begin with a simple upper bound on the Gram matrix.

Lemma 8 (Simple upper bound)**.** *If $\Lambda_t = \lambda I + \sum_{i \in [t]} \phi_i \phi_i^\top \in \mathbb{R}^d$ and $\lambda > 0$, then*

$$\sum_{i \in [t]} \phi_i^\top \Lambda_t^{-1} \phi_i \leq d.$$

Proof If $\lambda = 0$, then it is easy to see that $\sum_{i \in [t]} \phi_i^\top \Lambda_t^{-1} \phi_i = \mathrm{tr}(I_d) = d$. The regularization $\lambda > 0$ only makes the LHS smaller. □

The next lemma ensures boundedness of the linear weights.

Lemma 9 (Weights are bounded)**.** *(i) For each policy π and its Q function $Q_h^\pi(x, a) = \langle \phi(x, a), w_h^\pi \rangle$, we have $\|w_h^\pi\| \leq 2H\sqrt{d}, \forall h$. (ii) The weights $\{w_h^k\}$ in the LSVI-UCB algorithm satisfies $\left\|w_h^k\right\| \leq 2H\sqrt{dk}, \quad \forall k, h$.*

Proof Part (i) follows from Assumption on the linearity and boundedness. Part (ii) holds since the Gram matrix Λ_h^k has minimum eigenvalue ≥ 1 and satisfies the bound in Lemma 8. □

The analysis of multi-arm bandit frequently makes use of the scalar inequality

$$\log(t+1) \leq \sum_{j \in [t]} \frac{1}{j+1} \leq 2\log(t+1).$$

The next lemma, standard in the linear bandit literature, generalizes the above inequality.

Lemma 10 (Elliptical potential lemma [Jin et al., 2019]). *Suppose that* $\|\phi_t\| \leq 1, \forall t$, $\Lambda_0 \in \mathbb{R}^{d \times d}$ *is psd, and* $\Lambda_t = \Lambda_0 + \sum_{i \in [t]} \phi_i \phi_i^\top$. *If* $\lambda_{\min}(\Lambda_0) \geq 1$, *then*

$$\log\left(\frac{\det\Lambda_t}{\det\Lambda_0}\right) \leq \sum_{j \in [t]} \phi_j^\top \Lambda_j^{-1} \phi_j \leq 2\log\left(\frac{\det\Lambda_t}{\det\Lambda_0}\right), \forall t.$$

Our last lemma is a powerful concentration inequality.

Lemma 11 (Concentration for self-normalized processes [Abbasi-Yadkori et al., 2011, Theorem 1]). *Suppose* $(\epsilon_s)_{s=1,2,\ldots}$ *is a scalar stochastic process adapted to the filtration* $(\mathcal{F}_s)$, *and* $\epsilon_s|\mathcal{F}_{s-1}$ *is zero mean and* σ*-sub-Gaussian. Let* $(\phi_s)_{s=1,2,\ldots}$ *be an* $\mathbb{R}^d$*-valued stochastic process with* $\phi_s \in \mathcal{F}_{s-1}$. *Let* $\Lambda_t = I + \sum_{s=1}^t \phi_s \phi_s^\top \in \mathbb{R}^{d \times d}$. *Then with probability at least* $1-\delta$, *we have*

$$\left\|\sum_{s=1}^t \phi_s \epsilon_s\right\|_{\Lambda_t^{-1}}^2 \leq 2\sigma^2 \log\left[\frac{\det(\Lambda_t)^{1/2}}{\delta}\right], \qquad \forall t \geq 0.$$

References

[Abbasi-Yadkori et al., 2011] Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320.

[Jin et al., 2019] Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2019). Provably efficient reinforcement learning with linear function approximation. *arXiv preprint arXiv:1907.05388*.