CS 839 Probability and Learning in High Dimension Lecture 21 - 04/18/2022

Lecture 21: Linear MDPs I

Lecturer: Yudong Chen *Scribe: Rahul Choudhary, Govind Gopakumar*

In this lecture,[1] we conclude our previous discussions on self-normalized concentration bounds. We introduce the concept of Linear MDPs, and give an outline of the general setting. In addition, we define the episodic setting and contrast it with the earlier infinite horizon setting of learning MDPs.

1 Self normalized concentration

We first conclude our discussion on self-normalized concentration bounds. For a vector $u \in \mathbb{R}^d$ and a positive definite matrix $\Lambda_t^{-1} \in \mathbb{R}^{d\times d}$, define the weighted norm $\|u\|^2_{\Lambda_t^{-1}} := u^\top \Lambda_t^{-1} u$. Note that if $\Lambda_t^{-1} = I$, where I is the identity matrix, the weighted norm reduces to the ℓ_2 norm, that is, $\|u\|^2_{\Lambda_t^{-1}} = \|u\|_2^2$.

The following lemma generalizes the bound discussed at the end of last lecture to the vector and stochastic process setting.

Lemma 1 (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*

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

Remark It is instructive to consider the scalar setting $d = 1$, in which case Lemma 1 becomes:

$$\frac{\left|\sum_{s=1}^t \phi_s\epsilon_s\right|}{\sqrt{1+\sum_{s=1}^t \phi_s^2}} \lesssim \sigma \sqrt{\log\left[\frac{1+\sum_{s=1}^t \phi_s^2}{\delta}\right]} \qquad \text{w.h.p.}$$

When $\{\phi_s\}$ are deterministic constants, this bound is essentially the usual Azuma-Hoeffding type concentration inequality (up to the log factor on the RHS). The above bound is powerful as it allows the denominator (namely, the normalization factor) to depend on the process (ϕ_s) itself (hence the name). In this sense, the bound automatically identifies the right "scale" for the sum $\sum_{s=1}^t \phi_s\epsilon_s$ to be the squared variation $\Lambda_t = I + \sum_{s=1}^t \phi_s\phi_s^\top$. Moreover, thanks to this self-normalization, the bound does not require boundedness/moment/tail assumptions on (ϕ_s). In particular, suppose that (ϵ_s) are iid Radamacher RVs and (ϕ_s) are *any* independent RV sequence. Then the RVs $X_s := \phi_s\epsilon_s, s = 1,\ldots,t$ is just a *general* sequence of independent zero-mean symmetric RVs (which may be heavy tailed). The above lemma ensures that the sum of $\{X_s\}$ normalized by its square variation satisfies a sub-Gaussian type tail bound:

$$\frac{\left|\sum_{s=1}^t X_s\right|}{\sqrt{1+\sum_{s=1}^t X_s^2}} = \widetilde{O}\left(\sqrt{\log\delta^{-1}}\right) \qquad \text{w.h.p.}$$

Remark As mentioned above, in the scalar setting the denominator of the LHS ensures that the sum of variables is scaled appropriately by its square variation. Note that in the general case, the LHS norm is

[1] *Reading:* [Jin et al., 2020]

taken with respect to the matrix Λ_t^{-1}. This "normalizes" the variation within the vector, in the following sense: if there are directions along which the random vector $\sum_{s=1}^{t} \phi_s \epsilon_s$ has high variance, the eigenvalues corresponding to eigenvectors (which will be along these directions) of the matrix Λ_t will likely be large, so multiplying by the inverse of this matrix forces the variation to be scaled appropriately.

See [Van Handel, 2014, Problems 7.3] for further discussion about concentration for self-normalized processes.

2 Linear MDPs

In our earlier discussions regarding MDPs, we conveniently assumed that our state and action space were both discrete and finite. This allowed us to view the MDP essentially as a giant table, and in particular, all of our methods relied on observing / estimating / working with a matrix that roughly scaled in the sizes of both these spaces. Going forward, we shall see how this can be a limiting setting. In particular, results that we obtained in the previous setting relied on there being a finite number of states and actions, and in the case where the state space is large or even infinite, those results stop making sense. This motivates the study of a setting where we can work with an infinite state space.

In this lecture, we focus on *linear* function approximation. Within this section, our goal is to develop algorithms whose sample complexity and regret depends on the "effective size" of the problem (in particular, the dimension of the linear model), rather than the cardinalities of the state/action spaces.

2.1 Problem Setup

Notation: $[H] := \{1, 2, \dots, H\}$. $\|\cdot\|$ denotes the vector ℓ_2 norm.

Consider a **finite-horizon** MDP, expressed as a tuple $(\mathcal{S}, \mathcal{A}, r, \mathbb{P}, H)$, where

- $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space,
- $r = (r_h : \mathcal{S} \times \mathcal{A} \to [-1, 1])_{h \in [H]}$ represents the (deterministic, bounded) reward functions,
- $\mathbb{P} = (\mathbb{P}_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S}))_{h \in [H]}$ represents transition kernel, and
- H is the horizon (number of steps in each episode).

At state x at step $h \in [H] := \{1, \dots, H\}$, upon taking action a, the agent receives a reward $r_h(x, a)$ and then transitions to the next state $x' \sim \mathbb{P}_h(\cdot|x, a)$.[2]

Important to note here is that all these are allowed to vary with steps, as denoted by the subscript h.

2.2 Value functions and Bellman equations

A (stochastic) policy of the agent is of the form $\pi := (\pi_h : \mathcal{S} \to \Delta(\mathcal{A}))_{h \in [H]}$, where $\pi_h(\cdot|x)$ specifies the action distribution at state x at step h. It is important to contrast this with the earlier setting, here we have a sequence of π's corresponding to each step. For a fixed policy π, the value function and Q-function are defined as

$$V_h^\pi(x) := \mathbb{E}_\pi \left[\sum_{t=h}^{H} r_t(x_t, a_t) | x_h = x \right], \qquad Q_h^\pi(x, a) := \mathbb{E}_\pi \left[\sum_{t=h}^{H} r_t(x_t, a_t) | x_h = x, a_h = a \right],$$

where the expectation is taken under $a_t \sim \pi_t(\cdot|x_t)$ and $x_{t+1} \sim \mathbb{P}_t(\cdot|x_t, a_t)$, $t = 1, \dots, H$. That is, $V_h^\pi(x)$ is the expected cumulative rewards if the MDP starts from state $x_h = x$ and the agent follows the policy π. Similarly, $Q_h^\pi(x, a)$ is the expected cumulative rewards starting from $x_h = x, a_h = a$ and following π.

[2] Note that the reward function r_h and transition probabilities $\mathbb{P}_h(s'|s, a)$ are allowed to depend on the step h.

Let π^* be the optimal policy, which maximizes $V_h^\pi(x)$ for all $h \in [H]$ and $x \in \mathcal{S}$. Let V_h^* and Q_h^*, $h \in [H]$ be the corresponding optimal value and Q-functions.

Under the bounded reward assumption, it is easy see that all value functions are bounded:

$$\max_{s,a,h,\pi} \left\{ \left| V_h^\pi(x) \right|, \left| Q_h^\pi(x,a) \right|, \left| V_h^*(x) \right|, \left| Q_h^*(x,a) \right| \right\} \leq H.$$

We define the following shorthand for the conditional expectation under the transition kernel:

$$[\mathbb{P}_h V](x,a) := \mathbb{E}_{x' \sim \mathbb{P}_h(\cdot|x,a)}[V(x')].$$

Remark Note that when we work with continuous state space, the above expectation takes the form of an integral.

The value and Q-functions satisfy the following Bellman equations:

- $V_h^\pi(x) = Q_h^\pi(x, \pi(x))$ and $Q_h^\pi(x,a) := r_h(x,a) + (\mathbb{P}_h V_{h+1}^\pi)(x,a)$.
- $V_h^*(x) = \max_a Q_h^*(x,a)$ and $Q_h^*(x,a) := r_h(x,a) + (\mathbb{P}_h V_{h+1}^*)(x,a)$.

2.3 Linear structure

We assume that both the reward function and transition kernel have a linear structure, with respect to some *known* feature map. In the sequel $\|\cdot\|$ denotes the ℓ_2 norm on $\mathbb{R}^d$.

Assumption 1 (Linearity and Boundedness)**.** *For each $h \in [H]$ and $(x,a,x') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, it holds that*

$$\begin{aligned} \mathbb{P}_h(x'|x,a) &= \langle \phi(x,a), \mu_h(x') \rangle \qquad \textit{and} \\ r_h(x,a) &= \langle \phi(x,a), \theta_h \rangle, \end{aligned}$$

where

- *$\phi_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ is a **known** feature map,*
- *$\mu_h = (\mu_h^{(i)})_{i \in [d]}$ is a vector of d **unknown** (signed) measures on $\mathcal{S}$, and*
- *$\theta_h = (\theta_h^1, \ldots, \theta_h^d) \in \mathbb{R}^d$ is a vector of d **unknown** weights.*

We assume that $\max_{(x,a) \in \mathcal{S} \times \mathcal{A}} \|\phi_h(x,a)\| \leq 1$, $\|\mu_h(\mathcal{S})\| \leq \sqrt{d}$, $\|\theta_h\| \leq \sqrt{d}$ for all $h \in [H]$.[3]

Remark Note that the conditional distribution $\mathbb{P}_h(\cdot|x,a)$ is a linear combination of d unsigned measures $\mu_h^{(1)}, \ldots, \mu_h^{(d)}$. Assumption 1 requires that μ_h and ϕ are such that the linear combination results in a *probability* measure.

Remark Here is a toy example that illustrates when the linear MDP model is (approximately) satisfied. Consider the problem of training a self-driving car using reinforcement learning. Suppose the state $x \in \mathcal{S} := \{1, \ldots, 100\}$ describes how many cars are seen by the camera, and the action $a \in \mathcal{A} := 0, 1, 2, 3, 4$ describes the gear to be selected. In the tabular setting from previous classes, we would represent the reward function as a 100×4 table: $r = (r(x,a)) \in \mathbb{R}^{100 \times 4}$. It is possible that the actual rewards depend on a linear combination of x and a, say $r(x,a) = \theta^1 x + \theta^2 a$. An appropriate feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^2$ could simply be $\phi(x,a) = (x,a)^\top$, so $r(x,a) = \langle \phi(x,a), \theta \rangle$. This results in a much simpler, 2-dimensional setting.

[3] This normalization ensures consistency when reducing to tabular case.

Remark Note that the tabular MDP case is a special case of the linear formulation, where we simply have $d = |\mathcal{S}| \times |\mathcal{A}|$, and our feature map $\phi_h(x, a) = e_{x,a} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ is just the indicator vector (the $|S||A|$-dimensional vector with 1 at the (x, a) entry and 0 elsewhere). In this case, the vector $\theta_h \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ defining the reward function is given by

$$\theta_h = [r(1, 1), r(2, 1), \dots, r(x, a), \dots, r(|\mathcal{S}|, |\mathcal{A}|)]^\top.$$

One may verify that $r_h(x, a) = \langle \phi(x, a), \theta_h \rangle$.

The above assumption implies that the Q-function is linear for any policy (including the optimal policy).

Lemma 2 (Linearity of Q)**.** *For any policy π and $h \in [H]$, there exists a weight vector $w_h^\pi \in \mathbb{R}^d$ such that*

$$Q_h^\pi(x, a) = \langle \phi(x, a), w_h^\pi \rangle, \qquad \forall (x, a) \in \mathcal{S} \times \mathcal{A}.$$

In particular, the optimal Q-function satisfies $Q_h^*(x, a) = \langle \phi(x, a), w_h^* \rangle, \forall x, a$ for some $w_h^* \in \mathbb{R}^d$.

Proof By Bellman equation and linearity of r_h and $\mathbb{P}_h$, we have

$$Q_h^\pi(x, a) = r_h(x, a) + \mathbb{P}_h V_{h+1}^\pi(x, a) = \phi(x, a)^\top \theta_h + \int V_{h+1}^\pi(x')\phi(x, a)^\top \mathrm{d}\mu_h(x').$$

Letting $w_h^\pi := \theta_h + \int V_{h+1}^\pi(x')\mathrm{d}\mu_h(x')$ proves the lemma. □

Remark Note that we do not try to learn the vector μ_h at any point: each $\mu_h^{(i)}$ is a measure on $\mathcal{S}$ and hence a infinite-dimensional object when $\mathcal{S}$ is infinite, and it is impossible to learn it with only finite data. Instead, we are only concerned with learning the value/Q function V and Q. The existence of such a μ_h is assumed and used only in the analysis. There are some other related assumptions that are usually studied in this area of function approximation, [Jin et al., 2021, Section 2.1]

Remark Linearity of reward/transition is *strictly stronger* than linearity of the optimal Q-function Q^*. There is evidence that if one only assumes Q^* is linear, then the problem is statistically hard [Du et al., 2019].

2.4 Episodic setting and regrets

The agent interacts with the MDP in K episodes. At the beginning of episode k, the agent picks a policy $\pi^k = (\pi_1^k, \dots, \pi_H^k)$ and receives an (arbitrary) initial state x_1^k. The agents then executes the policy for H steps, resulting in the trajectory $x_1^k, a_1^k, r_1^k, \dots, x_H^k, a_H^k, r_H^k$. The system then resets and episode $(k + 1)$ begins.

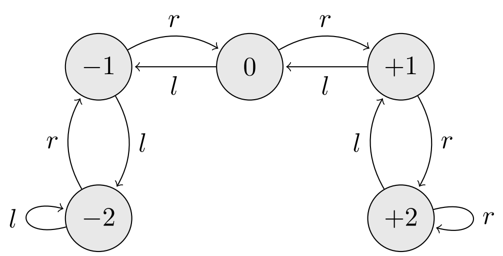
The regret over K episodes is defined as

$$\text{Regret}(K) := \sum_{k=1}^{K} \left[V_1^*(x_1^k) - V_1^{\pi^k}(x_1^k) \right],$$

which is the difference between the total value of the agent's policy $\pi^1, \dots, \pi^K$ and that of the optimal policy. We want to find an algorithm (for picking the policies $\pi^1, \dots, \pi^K$) that achieves a low regret.

Remark Note that the regret is defined using the true value, but when running the algorithm we do not have access to this. We can only construct estimates of the true value, and thus our algorithms need to ensure that they work with estimated values and still obtain low regret.

Remark The episodic setting can be contrasted with the simulator setting we studied in the earlier classes. In a simulator setting, we assume that we can independently draw samples from any state-action pair that we choose. In the episodic setting, this is not possible. To see a particular state or action, and the resulting reward, we must *first learn to reach that state* and take that action. This makes this setting more challenging, because we need to balance **exploration vs exploitation** when designing algorithms.



As an illustration, consider a simple 5-state MDP shown on the left, where states are represented by $\{-2, -1, 0, +1, +2\}$ and actions by $\{l, r\}$ (going left/right). Suppose 0 is the initial state. In the episodic setting, to see the reward for taking a particular action at state +2, we must first learn how to reach state +2 from the initial state 0 (this can be done by taking the action r twice). In contrast, in the simulator setting, we assume that we can directly draw samples from state +2 (and any other state/action) via the simulator (aka sampling oracle).

References

[Abbasi-Yadkori et al., 2011] Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24.

[Du et al., 2019] Du, S. S., Kakade, S. M., Wang, R., and Yang, L. F. (2019). Is a good representation sufficient for sample efficient reinforcement learning? *arXiv preprint arXiv:1910.03016*.

[Jin et al., 2021] Jin, C., Liu, Q., and Miryoosefi, S. (2021). Bellman Eluder dimension: New rich classes of RL problems, and sample-efficient algorithms. *Advances in Neural Information Processing Systems*, 34.

[Jin et al., 2020] Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR.

[Van Handel, 2014] Van Handel, R. (2014). Probability in high dimension. Technical report, PRINCETON UNIV NJ.