ECE 586 BH: Interplay between Control and Machine Learning **Fall 2023**

Lecture 15

A Jump System Perspective on Temporal Difference Learning, Part I

Lecturer: Bin Hu, Date:10/10/2023

In the previous lectures, we have shown that jump system theory can be used to provide a general analysis for stochastic optimization methods. Now we will extend this idea to address the complexity analysis of reinforcement learning (RL) algorithms. Specifically, we will consider the policy evaluation problem in RL, and study how jump system theory can be used to analyze the famous temporal difference (TD) learning algorithm.

15.1 Background: Policy Evaluation in RL

RL refers to a collection of techniques for solving Markov decision processes (MDPs). Here we briefly review some background materials for MDPs and policy evaluation. A MDP is defined by a tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the transition kernel, R is the reward, and γ is the discount factor. The goal is to choose the action sequence $\{a_t\}$ to maximize the total accumulated rewards $V(s_0) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k) \big| s_0\right]$.

The action a_t is typically determined using a feedback function of s_t. This is a natural consequence of dynamic programming and the concept of feedback is very important for managing uncertainty. The feedback law mapping s_t to a_t is called "policy" (in the reinforcement learning literature) or "controller" (in some controls literature). A policy can be deterministic (i.e. $s_t = \pi(a_t)$ where π is a fixed nonlinear function) or stochastic (i.e. it maps each state to a probability distribution over the action space $\mathcal{A}$). Then the goal can be equivalently formulated as finding an optimal policy that maximizes the total accumulated rewards, i.e.

$$\pi^* = \arg\max_{\pi} \ \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k) \big| a_k \sim \pi(\cdot|s_k), s_0\right].$$

If both $\mathcal{S}$ and $\mathcal{A}$ are finite, then the policy parameterization is straightforward. Any deterministic policy can be actually represented by a vector. For simplicity, consider $\mathcal{S} = \{1, 2, \ldots, n\}$ and $\mathcal{A} = \{1, 2, \ldots, L\}$. Then a deterministic policy π can be specified as as vector

$$\begin{bmatrix} \pi(1) \\ \pi(2) \\ \vdots \\ \pi(n) \end{bmatrix}$$

where $\pi(i) \in \mathcal{A}$. Clearly the above vector is in $\mathbb{R}^n$. If we further consider a stochastic policy, then each $\pi(i)$ is a probability distribution over $\mathcal{A}$ and hence is a vector in $\mathbb{R}^L$. Therefore, we have $\pi \in \mathbb{R}^{nL}$ for any stochastic policy.

To find the best policy, at least we need some tools to assess the performance of a given policy. This task is called policy evaluation. The goal here is to calculate the value function for any given policy.

$$V^\pi(s_0) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k) \big| a_k \sim \pi(\cdot|s_k), s_0\right].$$

For simplicity, let's first consider the value evaluation of a deterministic policy. If both $\mathcal{S}$ and $\mathcal{A}$ are finite, then the policy π can be represented as a vector. Then the value function becomes

$$V^\pi(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_k, \pi(s_k)) \big| s_0 = s\right].$$

Now we can apply the law of total probability to show:

$$\begin{aligned} V^\pi(s) &= \mathbb{E}R(s_0, \pi(s_0)) + \mathbb{E}\left[\sum_{k=1}^{\infty} \gamma^k R(s_k, \pi(s_k)) \big| s_0 = s\right] \\ &= \mathbb{E}R(s, \pi(s)) + \sum_{s' \in \mathcal{S}} \left(\mathbb{E}[\sum_{k=1}^{\infty} \gamma^k R(s_k, \pi(s_k)) \big| s_1 = s']\right) P(s_1 = s' \big| s_0 = s) \end{aligned}$$

When π is fixed, the state $\{s_k\}$ becomes a Markov chain. We have $P(s_1 = s' \big| s_0 = s) = P(s_1 = s' \big| s_0 = s,\, a_0 = \pi(s)) = P_{ss'}^{\pi(s)}$. Notice $\mathbb{E}\left[\sum_{k=1}^{\infty} \gamma^k R(s_k, \pi(s_k)) \big| s_1 = s'\right] = \gamma V^\pi(s')$. If we denote $\bar{R}^\pi(s) := \mathbb{E}R(s, \pi(s))$, then the equation on V^π can be rewritten as

$$V^\pi(s) = \bar{R}^\pi(s) + \gamma \sum_{s' \in \mathcal{S}} V^\pi(s') P_{ss'}^{\pi(s)} \tag{15.1}$$

which is the so-called Bellman equation. Recall $\mathcal{S} = \{1, 2, \ldots, n\}$. We can actually rewrite the above Bellman equation in the following matrix form:

$$\begin{bmatrix} V^\pi(1) \\ \vdots \\ V^\pi(n) \end{bmatrix} = \begin{bmatrix} \bar{R}^\pi(1) \\ \vdots \\ \bar{R}^\pi(n) \end{bmatrix} + \gamma \begin{bmatrix} P_{11}^{\pi(1)} & \cdots & P_{1n}^{\pi(1)} \\ \vdots & \ddots & \vdots \\ P_{n1}^{\pi(n)} & \cdots & P_{nn}^{\pi(n)} \end{bmatrix} \begin{bmatrix} V^\pi(1) \\ \vdots \\ V^\pi(n) \end{bmatrix}$$

If we use the following vector notation:

$$V^\pi = \begin{bmatrix} V^\pi(1) \\ V^\pi(2) \\ \vdots \\ V^\pi(n) \end{bmatrix},\ \bar{R}^\pi = \begin{bmatrix} \bar{R}^\pi(1) \\ \bar{R}^\pi(2) \\ \vdots \\ \bar{R}^\pi(n) \end{bmatrix},\ P^\pi = \begin{bmatrix} P_{11}^{\pi(1)} & \cdots & P_{1n}^{\pi(1)} \\ \vdots & \ddots & \vdots \\ P_{n1}^{\pi(n)} & \cdots & P_{nn}^{\pi(n)} \end{bmatrix}$$

we can rewrite the Bellman equation as

$$V^\pi = \bar{R}^\pi + \gamma P^\pi V^\pi. \tag{15.2}$$

Therefore, the policy evaluation becomes an equation solving problem. Notice P^π is actually the transition matrix for the Markov chain $\{s_k\}$. Clearly, this matrix is right stochastic and the spectral radius is 1. Therefore, $I - \gamma P^\pi$ is nonsingular for any $0 < \gamma < 1$, and the Bellman equation admits a unique solution

$$V^\pi = (I - \gamma P^\pi)^{-1} \bar{R}^\pi$$

If we want to avoid matrix inversion, we can use an iterative scheme:

$$V^\pi_{k+1} = \bar{R}^\pi + \gamma P^\pi V^\pi_k$$

which is equivalent to a linear time-invariant system:

$$V^\pi_{k+1} - V^\pi = \gamma P^\pi (V^\pi_k - V^\pi)$$

Since the spectral radius of γP^π is γ, we immediately know the above system converges to V^π at a linear rate γ. When a stochastic policy is used, the Bellman equation still holds. We only need to slightly modify the definitions of $\bar{R}^\pi$ and P^π. For example, now we have $\bar{R}^\pi(s) = \mathbb{E}\left[R(s,a) \middle| a \sim \pi(\dot{|}s)\right]$. I will let you figure out how to modify P^π. In general, when a fixed stochastic policy is used, the state $\{s_k\}$ still becomes a Markov chain and P^π is the associated transition matrix. Then V^π can still be solved via the Bellman equation.

15.2 Model-Free Setting: TD Learning

The above discussion assumes knowing P^π. How to solve the Bellman equation when we do not know P^π? Here is the basic idea of TD learning. Suppose we are using an iterative method and estimate the value function as V^π_k at step k. We sample the trajectory of the underlying Markov chain as $\{s_k\}$. Based on the Bellman equation, $V^\pi(s_k)$ can be estimated in two ways: either $V^\pi_k(s_k)$ or $r(s_k, \pi(s_k)) + \gamma V^\pi_k(s_{k+1})$ (this is called TD target). One reasonable way to do things is to minimize the difference of these two things: $(V^\pi_k(s_k) - r(s_k, \pi(s_k)) - \gamma V^\pi_k(s_{k+1}))^2$. How to make this difference small? Averaging!

$$\begin{aligned} V^\pi_{k+1}(s_k) &= (1-\varepsilon) V^\pi_k(s_k) + \varepsilon (r(s_k, \pi(s_k)) + \gamma V^\pi_k(s_{k+1})) \\ &= V^\pi_k(s_k) + \varepsilon \left(r(s_k, \pi(s_k)) + \gamma V^\pi_k(s_{k+1}) - V^\pi_k(s_k) \right) \end{aligned}$$

Denote $\delta_k = r(s_k, \pi(s_k)) + \gamma V^\pi_k(s_{k+1}) - V^\pi_k(s_k)$. The term δ_k is the so-called TD error. The above algorithm is called TD(0)[1]. Usually ε is small, and this means that we do not want to make too much change for a given random sample point.

Now we switch our focus to the function approximation case. Suppose n is too large and we do not want to learn an n-dimensional vector for V^π. Instead, we estimate $V^\pi(s) \approx \theta^\mathsf{T} \phi(s)$ where ϕ is the feature vector. Here, after we get a sample trajectory, what shall we do? We

[1] An extension which interpolates TD(0) and Monte Carlo method is TD(λ), which relies on the use of the eligibility traces. We skip the details of such extensions.

can try to minimize $\frac{1}{2}\|\theta^{\mathsf{T}}\phi(s_k) - r(s_k, \pi(s_k)) - \gamma\theta_k^{\mathsf{T}}\phi(s_{k+1})\|^2$. However, we do not want to completely solve this problem since we do not want to trust one sample point too much. Instead, we can perform one step gradient descent as

$$\theta_{k+1} = \theta_k + \varepsilon \left(r(s_k, \pi(s_k)) + \gamma\theta_k^{\mathsf{T}}\phi(s_{k+1}) - \theta_k^{\mathsf{T}}\phi(s_k)\right)\phi(s_k)$$

Again, we have TD error $\delta_k = r(s_k, \pi(s_k)) + \gamma\theta_k^{\mathsf{T}}\phi(s_{k+1}) - \theta_k^{\mathsf{T}}\phi(s_k)$. Then we can ask what types of theoretical guarantees can be proved. Notice TD(0) can be modeled as a linear stochastic approximation scheme $\theta_{k+1} = \theta_k + \varepsilon(A_{i_k}\theta_k + b_{i_k})$ where i_k is the augmented vector of (s_{k+1}, s_k). As $\varepsilon \to 0$, we will have an ODE to model its asymptotic dynamics: $\dot{\theta} = \bar{A}\theta + \bar{b}$ where $\bar{A}$ and $\bar{b}$ are some averaged quantities. Based on the so-called ODE approach, the asymptotic convergence of TD(0) with diminishing step size can be guaranteed by the stability of this ODE and several extra technical conditions. More recently, the finite sample bounds have also been developed. We will discuss how to leverage the jump system theory to derive such finite-time analysis. Before moving to cover such control-theoretic analysis, we first make a few remarks.

Off policy vs. on policy. The above method is on policy since the data has to be sampled using the policy π. Sometimes we do not use to directly use π to gather data since that may be dangerous. Then we will use off-policy methods. In off-policy TD learning, the data is sampled from a behavior policy μ. Then importance sampling trick is used to evaluate the value function of the target policy π using the data generated by the behavior policy μ. What is importance sampling? Originally important sampling is used in rare event simulations to sample "important but rare" events. Just imagine $X \sim \mathcal{N}(0, 1)$ and one wants to estimate $P(X > 10) = \mathbb{E}\mathbf{1}_{X>10}$ using Monte Carlo simulations. Since it is extremely unlikely to get a sample with $X > 10$, the Monte Carlo simulation may just return 0. Suppose $f(x)$ is the density function for $\mathcal{N}(0, 1)$ and $g(x)$ is the density function for $\mathcal{N}(10, 1)$. Then importance sampling uses the following reformulation:

$$\mathbb{E}\mathbf{1}_{X>10} = \int_{-\infty}^{\infty} \mathbf{1}_{X>10} f(x)dx = \int_{-\infty}^{\infty} \mathbf{1}_{X>10} \frac{f(x)}{g(x)} g(x)dx$$

Now we can sample X from $\mathcal{N}(10, 1)$ and average the quantity $\mathbf{1}_{X>10}\frac{f(x)}{g(x)}$. Clearly by doing this, we see the event $X > 10$ much more often, and this is exactly the idea of importance sampling. Here the ratio of the two densities is called importance sampling ratio. Off-policy TD learning uses a similar idea, and requires using the importance sampling ratio between π and μ. To study some details of off-policy TD learning, see Sections 2-4 in [3].

Online vs. off-line: Least square methods. TD(0) is an online method. When a new data point s_k is observed, the weight θ_k is updated and then the data point s_k is completely thrown away after this update. How to make more efficient use of data? If all the data are available, then we can use off-line methods. For example, when the linear approximation is used, we can apply the least square method to fit θ directly. If we

look at the recursive formula for TD(0), eventually the method will converge if the term $\big(r(s_k, \pi(s_k)) + \gamma\theta_k^\mathsf{T}\phi(s_{k+1}) - \theta_k^\mathsf{T}\phi(s_k)\big)\,\phi(s_k)$ is roughly 0. Therefore, we want to choose θ to have

$$\phi(s_k)\left(\phi(s_k) - \gamma\phi(s_{k+1})\right)\theta \approx \phi(s_k) r(s_k, \pi(r_k)),\ \forall k$$

This becomes a least square problem. We can just choose $A = \sum_k \phi(s_k)\left(\phi(s_k) - \gamma\phi(s_{k+1})\right)$ and $b = \sum_k \phi(s_k) r(s_k, \pi(r_k))$. Then we fit θ as $\theta = A^{-1}b$. When A is not invertible, we can use pseudo inverse. We can add eligibility trace into the algorithm. Details are omitted.

Summary of the key idea in TD learning. The basic idea of TD learning is to fit the value function by enforcing the left and right sides of the Bellman equation to be roughly equal to each other on the observed data. A similar idea can be used to find the optimal Q function. One can try to fit the optimal value function by enforcing the left and right sides of the optimal Bellman equation to be roughly equal to each other on the observed data. This is the idea behind value-based reinforcement learning methods.

15.3 A Jump System Perspective on TD Learning

As discussed previously, TD(0) with linear function approximation can be rewritten as $\theta_{k+1} = \theta_k + \varepsilon(A_{i_k}\theta_k + b_{i_k})$. Suppose θ_π is the solution to the projected Bellman equation[2] for the given policy π. How can we obtain a finite time bound for the mean square TD estimation error $\mathbb{E}\|\theta_k - \theta_\pi\|^2$? Obtaining such a bound under the IID assumption on $\{i_k\}$ is relatively straightforward. However, for TD learning, $\{i_k\}$ is typically Markov. Obtaining a finite sample bound under such a Markov assumption is more technical. There are several different techniques that can be used to tackle this difficulty. We will present one argument based on the jump system theory. Specifically, we can rewrite the TD scheme as

$$\theta_{k+1} - \theta_\pi = \theta_k - \theta_\pi + \varepsilon A_{i_k}(\theta_k - \theta_\pi) + \varepsilon(A_{i_k}\theta_\pi + b_{i_k}),$$

which is equivalent to

$$\theta_{k+1} - \theta_\pi = (I + \varepsilon A_{i_k})(\theta_k - \theta_\pi) + \varepsilon(A_{i_k}\theta_\pi + b_{i_k}). \tag{15.3}$$

Denote $\xi_k = \theta_k - \theta_\pi$, $G_{i_k} = \varepsilon(A_{i_k}\theta_\pi + b_{i_k})$, and $H_{i_k} = I + \varepsilon A_{i_k}$. The above scheme just becomes a jump system:

$$\xi_{k+1} = H_{i_k}\xi_k + G_{i_k}u_k \tag{15.4}$$

where $u_k = 1$ for all k. Notice that the jump system theory established in the control field can be directly applied to obtain closed-form formula for $\mathbb{E}\|\xi_k\|^2$ under the assumption that $\{i_k\}$ is either IID or Markov [1, Proposition 3.35]. As a matter of fact, the jump system theory can actually handle general forms of $\{H_i, G_i\}$. Such an analysis for TD learning was originally developed in [2]. We will present the detailed analysis in the next lecture.

[2]The projected Bellman equation and the equation $\sum_{i=1}^n p_i^\infty b_i = 0$ are actually equivalent, i.e. we have $\sum_i p_i^\infty(A_i\theta_\pi + b_i) = 0$.

Bibliography

[1] O. Costa, M. Fragoso, and R. Marques. *Discrete-time Markov jump linear systems*. Springer Science & Business Media, 2006.

[2] B. Hu and U. Syed. Characterizing the exact behaviors of temporal difference learning algorithms using Markov jump linear system theory. In *Advances in Neural Information Processing Systems*, pages 8479–8490, 2019.

[3] R. S. Sutton, A. R. Mahmood, and M. White. An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*, 17(1):2603–2631, 2016.