EE 290 Theory of Multi-armed Bandits and Reinforcement Learning Lecture 21 - 4/6/2021

Lecture 21: TD Learning with Linear Function Approximation

Lecturer: Jiantao Jiao *Scribe: Joey Hejna*

The majority of this lecture's content is from Bhandaru et al. [1]. This lecture presents a proof for analyzing Temporal difference (TD) Learning that is relatively simple, but was not discovered for a long time due to misplaced focus on using contractions in analysis. First we will introduce the different data collection assumptions one can make, then proceed to describe the TD-learning algorithm. Finally, we will conclude by providing a basica analysis of the TD learning algorithm assuming linear function approximation. The jamboard for this lecture can be found here.

Temporal Difference (TD) Learning is a method of policy evaluation with on-policy data. That means given a policy $\mu(a|s)$, we need to run it in the environment to collect data and determine V_μ. Note that to be consistent with [1] we denote policies as μ instead π. π will instead refer to the stationary distribution of the markov chain induced by running the policy μ in the environment MDP.

1 Sampling

In order to run the TD-learning algorithm, we need to obtain data. That means that the means of collecting data from the environment directly impacts the performance and bounds of our algorithm. There are usually two different types of assumptions one can make about the data, which make sense in different application settings. Assume that the reward is given as a deterministic function of the state transition: $r = R(s, s')$.

1. **i.i.d. data assumption**: We have iid samples of state, reward, next state tuples (s, r, s') sampled from the stationary distribution of the markov chain induced by running the policy in the MDP. In particular $P(((s, r, s') = (s, R(s, s'), s')) = \pi(s)p(s'|s)$. One may ask where the action went. They are in fact implicitly buried or absorbed in the markov reward process induced by running the policy in the MDP. As we always $\mu(a|s)$, we can determine the transition probabilities as $P(s'|s, \mu(a|s))$. We assume that this induced markov chain is ergodic with a unique stationary distribution, $\pi(s)$. After assembling the entire joint distribution of actions, states, and next states, we sample from it independently. $pi(s)$ can also be thought of as the marginal empirical distribution of the replay buffer.
2. **Markov Sampling**: Rather than sampling iid from the stationary distribution, we simply collect rollouts in the environment by choosing actions according to $\mu(a|s)$. This results in a sequence of states $(s_0, s_1, s_2, s_3, ...)$. Between each of the states, we get a reward $r = R(s_t, s_{t+1})$ just as before. Note that in this case, the tuples $(s, r(s, s'), s')$ are not independent, as they were sampled in a sequence. Some people think this is a more accurate assumption.

In this lecture, we won't have time to cover Markov Sampling. The proof for the Markov sampling case involves some theory tricks that can be found in the references. We will focus on the iid case.

2 Notation

The description and proof of convergence for the TD learning algorithm under linear function approximation requires some new notation that we will cover in this section.

As we stated above μ will denote the policy and π the stationary distribution induced by it. As is standard, we will define the value function induced by the policy as $V_\mu(s) = \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, s_{t+1})|s_0 = s]$.

The Bellman operator $\mathcal{T}$ is a special function that associates a value function at the current state to the value functions of the next state.

$$(\mathcal{T}_\mu V)(s) = \mathrm{E}_{s' \sim P(\cdot|s)}[R(s, s')] + \gamma \, \mathrm{E}_{s' \sim p(\cdot|s)}[V(s')]$$

The value function induced by μ gives a fixed point for the bellman operator $\mathcal{T}_\mu$: $\mathcal{T}_\mu V_\mu = V_\mu$. Note that we also assume that rewards are bounded: $|R(s, s')| \leq r_{\max}$

Next we will discuss the linear function approximation assumption. Rather than taking on any form, we assume that the value function can be modeled by a linear function of features of the state. Concretely, $V_\mu(s) \approx V_\theta(s) = \phi(s)^\top \theta$ where $\phi(s)$ are known features of the state and θ are the weights of the function approximator. θ and $\phi(s)$ are in $\mathbb{R}^d$, the feature space. Fixing the policy, we refer to this linear approximation as V_θ.

When the state space is finite, we can write everything in matrix form as $V_\theta \in \mathbb{R}^S$. We get the following:

$$V_\theta = \begin{bmatrix} \phi(s_1)^\top \\ \vdots \\ \phi(s_n)^\top \end{bmatrix} \theta = \begin{bmatrix} \phi_1(s_1) & \phi_k(s_1) & \phi_d(s_1) \\ \vdots & \vdots & \vdots \\ \phi_1(s_n) & \phi_k(s_n) & \phi_d(s_n) \end{bmatrix} \theta = \Phi\theta,$$

where $\Phi \in \mathbb{R}^{n \times d}$ and $\theta \in \mathbb{R}^d$.

To complete our analysis, we will need a new definition of distance with respect to value functions. This will let us determine if our value function is in fact converging. We will do this using the matrix representation of the value function and the following inner product defined for Positive semi-definite matrices A: $\langle x, y \rangle_A = x^\top A y$. The induced norm by this inner product is $||x||_A = \sqrt{x^\top A x}$. Let D be the diagonal matrix with the stationary distribution $\pi(s)$ along the diagonal of size $|S|$: $D = \text{diag}(\pi(s_1), \ldots, \pi(s_n)) \in \mathbb{R}^{|S| \times |S|}$. We use D to define a distance between value functions:

$$\|V_\theta - V_{\theta'}\|_D = \sqrt{\sum_{s \in \mathcal{S}} \pi(s) \left(\phi(s)^\top(\theta - \theta')\right)^2} = \sqrt{(\theta - \theta')^\top \Sigma (\theta - \theta')} = \|\theta - \theta'\|_\Sigma$$

where

$$\Sigma := \Phi^\top D \Phi = \sum_{s \in S} \pi(s) \phi(s) \phi(s)^\top$$

Notice that we are weighting the distance in value function by the stationary distribution, or how often we see that state. In all updates, we update the weights θ as it defines the value function.

3 TD Learning Algorithm

Now that we have set up sufficient notation, we can give the TD learning algorithm. Specifically, we will cover the TD(0) algorithm with linear function approximation. See Algorithm 1.

Algorithm 1: TD(0) with linear function approximation

Input : initial guess θ_0, step-size sequence $\{\alpha_t\}_{t \in \mathbb{N}}$.
Initialize: $\bar{\theta}_0 \leftarrow \theta_0$.
for $t = 0, 1, \ldots$ **do**
- Observe tuple: $O_t = (s_t, r_t = r(s_t, s'_t), s'_t)$;
- Define target: $y_t = r(s_t, s'_t) + \gamma V_{\theta_t}(s'_t)$; `/* sample Bellman operator */`
- Define loss function: $\frac{1}{2}(y_t - V_\theta(s_t))^2$; `/* sample Bellman error squared */`
- Compute negative gradient: $g_t(\theta_t) = -\frac{\partial}{\partial \theta} \frac{1}{2}(y_t - V_\theta(s_t))^2 \big|_{\theta = \theta_t}$;
- Take a gradient step: $\theta_{t+1} = \theta_t + \alpha_t g_t(\theta_t)$; `/* `α_t`:step-size */`
- Update averaged iterate: $\bar{\theta}_{t+1} \leftarrow \left(\frac{t}{t+1}\right) \bar{\theta}_t + \left(\frac{1}{t+1}\right) \theta_{t+1}$; `/* `$\bar{\theta}_{t+1} = \frac{1}{t+1} \sum_{\ell=0}^{t} \theta_\ell$ `*/`

end

This algorithm can be interpreted as randomly sampling a target from the next state, then minimizing the L2 loss between the current value function and the target. We proceed as follows. First, we sample the

observation tuple, then define the target via a random sample instead of an expectation. Then, we perform a step of gradient descent. $g_t(\theta_t)$ is defined to be the negative gradient of the loss at θ_t:. We then update the weights. Note that the TD target is a random variable that depends on θ_t, which is the real target in expectation. When we use the TD algorithm, we hope the gradient from the sample is good and that it well let us converge.

4 Analysis

First we can begin by determining the gradient as a general function of θ.

$$\begin{aligned} g_t(\theta) &= -\frac{\partial}{\partial\theta}\frac{1}{2}(y_t - V_\theta(s_t))^2 = -\frac{\partial}{\partial\theta}\frac{1}{2}(y_t - \phi(s_t)^\top\theta)^2 \\ &= (y_t - \phi(s)^\top\theta)\phi(s_t) \\ &= (r_t + \gamma\phi(s_t')^\top\theta - \phi(s_t)^\top\theta)\phi(s_t) \end{aligned}$$

Note that the first term is a scalar and the second term is a vector in $\mathbb{R}^d$. It's also important to realize that this update gradient does not yield gradient descent. The algorithm we have does not exactly reduce to gradient descent because the targets move at each step. Thus, we are not running gradient descent on a convex function. However, optimization theorists have realized that the key analysis techniques from gradient descent.

We characterize the dynamics of the system by the expected negative gradient step. We define $\bar{g}(\theta)$ as follows:

$$\begin{aligned} \bar{g}(\theta) &= \mathrm{E}_{(s_t,r_t,s_t')\sim\pi(s_t)1\{r_t=R(s_t,s_t')\}P(s_t'|s_t)}[g_t(\theta)] \\ &= \sum_{s,s'\in S}\pi(s)\pi(s'|s)\left(R(s,s') + \gamma\phi(s')^\top\theta - \phi(s)^\top\theta\right)\phi(s) \end{aligned}$$

Now we get to the magic part. Suppose the true value function is $V_\mu^*(s) = \phi(s)^\top\theta^*$. We can then analyze the dynamics of the weights using approaches from gradient descent that are gone over in more detail in optimization focused courses, like 227C. We do this by looking at how the distance to the optimal weights evolves over training.

$$\begin{aligned} ||\theta^* - \theta_{t+1}||_2^2 &= ||\theta^* - (\theta_t + \alpha_t\bar{g}(\theta_t))||_2^2 \\ &= ||\theta^* - \theta_t||_2^2 - 2\alpha_t(\theta^* - \theta_t)^\top\bar{g}(\theta_t) + \alpha_t^2||\bar{g}(\theta_t)||_2^2 \end{aligned} \tag{1}$$

This step is completed by expanding out the square of the L2 norm by $||x||_2^2 = x^\top x$. In order for this to converge, we need to show that the subtracted second term is big while the added third term is small. This way with each step, we subtract more from our distance to the optimal weight vector than we add. There are typical methods for doing this in convex optimization that do not apply here because our optimization function isn't fixed or convex. However, you can replicate analysis from other areas to get a two line proof!

What we are going to show can be summarized in Figure 1. At the poin θ_t, there are multiple directions we could move to in optimization space. g_t points in one direction, the vector to θ^* or $\theta^* - \theta$ is in another direction. We cannot directly move in this direction because we do not have true information because of the target being not an expectation. However, we can get closer to θ^*. The second term in Equation 1 being big means the vector to the optimal value is close to the estimate of the gradient, as their dot product is large. Meanwhile as we converge, the average gradient size decreases or the third term in Equation 1 gets smaller.

Now we will begin formal analysis on the distance to the optimal solution.

In order to show this convergence, we thus derive two lemmas to show that the latter two terms in Equation 1 are indeed "small" and "big". Note that we will use the following claim in showing these Lemmas.

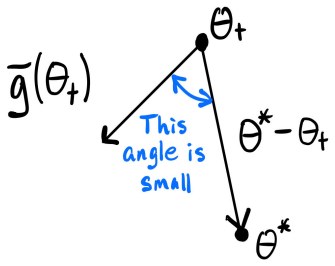

Figure 1: Diagram of weight space of the optimization.

Claim 1. $\bar{g}(\theta^*) = 0$.

This is true as at the optimal point, the gradient of the expected squared loss ought to be zero.

Lemma 2. *For any $\theta \in \mathbb{R}^d$. We have that $(\theta^* - \theta)^\top \bar{g}(\theta) \geq (1-\gamma)||V_{\theta^*} - V_\theta||_D^2$*

This means that at each step we make progress of at least $1-\gamma$ towards reducing the distance between optimal value functions.

Proof As we are in the iid case, consider a tuple of starts starting at $s \sim \pi(s)$ and $s' \sim P(s'|s)$ or $(s, s') \sim P(s'|s)\pi(s)$. For purposes of this proof, set $\phi \leftarrow \phi(s)$, $\phi' \leftarrow \phi(s')$ and $r \leftarrow R(s, s')$. Define $\xi = V_{\theta^*}(s) - V_\theta(s) = (\theta^* - \theta)^\top \phi$ and $\xi' = V_{\theta^*}(s') - V_\theta(s') = (\theta^* - \theta)^\top \phi'$. Notice that ξ and ξ' are both random variables the represent the observed difference between the value function estimates. As they are both over states, they have the same marginal distribution. This will be important later. We can start with the left hand side of what we want to prove.

$$\begin{aligned}
(\theta^* - \theta)^\top \bar{g}(\theta) &= (\theta^* - \theta)^\top (\bar{g}(\theta) - \bar{g}(\theta^*)) \\
&= (\theta^* - \theta)^\top \mathrm{E}[(r + \gamma \phi'^\top \theta - \phi^\top \theta)\phi - (r + \gamma \phi'^\top \theta^* - \phi^\top \theta^*)\phi] \\
&= \mathrm{E}\left[(\theta^* - \theta)^\top \phi(\gamma\phi' - \phi)^\top (\theta - \theta^*)\right] \\
&= \mathrm{E}[\xi(\xi - \gamma\xi')]
\end{aligned}$$

We can now expand the expectation and apply Cauchy-Schwartz to the second term.

$$\mathrm{E}\left[\xi(\xi - \gamma\xi')\right] = \mathrm{E}[\xi^2] - \gamma\,\mathrm{E}[\xi'\xi] \geq (1-\gamma)\,\mathrm{E}[\xi^2] = (1-\gamma)\,\|V_{\theta^*} - V_\theta\|_D^2$$

Why is this exactly true? Cauchy-Schwartz tells us that $|\,\mathrm{E}[\xi\xi']| \leq \sqrt{\mathrm{E}[\xi^2]\,\mathrm{E}[\xi'^2]} = \mathrm{E}[\xi^2]$ because as we mentioned before ξ and ξ' have the same marginal distributions. In the last step, we apply the fact that $\mathrm{E}[\xi^2] = \mathrm{E}[((\theta^* - \theta)^\top \phi(s))^2] = ||V_{\theta^*} - V_\theta||_D^2$ □

This lemma tells us that the first term that we subtract is at least proportional to the current distance minus some factor. We know just need to prove that the second term is smaller than this term. Notice the factor of two in front of this term in Equation 1. This means that we have that the second term subtracts at least $2\alpha||V_{\theta^*} - V_\theta||_D^2$. We thus need the third term to add at most this amount. Thus, we have to upper bound $||\bar{g}(\theta_t)||_2$ We do this using the following lemma, that effectively shows that the third term is small.

Lemma 3. *For any $\theta \in \mathbb{R}^d$. We have that $||\bar{g}(\theta)||_2 \leq 2||V_\theta - V_{\theta^*}||_D$*

Proof We can borrow some of the ideas and notation from the last Lemma.

$$\bar{g}(\theta) = \bar{g}(\theta) - \bar{g}(\theta^*) = \mathrm{E}[\phi(\gamma\phi' - \phi)^\top (\theta - \theta^*)] = \mathrm{E}[\phi(\xi - \gamma\xi')]$$

We now consider the norm of $\bar{g}(\theta)$.

$$||\bar{g}(\theta)||_2 = ||\,\mathrm{E}[\phi(\xi - \gamma\xi')]||_2 \leq \sqrt{\mathrm{E}\left[\|\phi\|_2^2\right]}\sqrt{\mathrm{E}\left[(\xi - \gamma\xi')^2\right]}$$

Note that $||\phi||_2 \leq 1$ by assumption, and thus $\|\phi\|_2^2 \leq 1$. Now we have to deal with the second term. We maximize the second term in the case when $\xi = -\xi'$ and consequently $\xi - \gamma\xi' = (1+\gamma)\xi$. We get the following.

$$\sqrt{\mathrm{E}\left[\|\phi\|_2^2\right]}\sqrt{\mathrm{E}\left[(\xi - \gamma\xi')^2\right]} \leq \sqrt{\mathrm{E}\left[(1+\gamma)^2(\xi^2)\right]} \leq (1+\gamma)\sqrt{\mathrm{E}[\xi^2]} = (1+\gamma)||V_{\theta^*} - V_\theta||_D$$

This last line comes from the same equality of $\mathrm{E}[\xi^2]$ shown in Lemma 1. Makes sense because if $\theta = \theta^*$ then the LHS is zero. Then the RHS is the same thing that we use the bound the other big part. So the the same bound on both. Finally, we know that $\gamma \leq 1$ by definition, so $1 + \gamma \leq 2$. This lets us conclude that $||\bar{g}(\theta)||_2 \leq ||V_{\theta^*} - V_\theta||_D$. □

Now that we have derived these two Lemmas, we can use them to prove the convergence of TD learning.

Theorem 2. *Choosing $\alpha = (1-\gamma)/4$, we have that*

$$||\theta^* - \theta_{t+1}||_2^2 \leq \exp\left\{-\frac{\omega(1-\gamma)^2}{4}\right\}\|\theta^* - \theta_t\|_2^2$$

Where ω is the minimum eigenvalue of Σ.

Proof Let's start with Equation 1 and apply both Lemmas, then substitute in the learning rate.

$$\begin{aligned}\|\theta^* - \theta_{t+1}\|_2^2 &\leq \|\theta^* - \theta_t\|_2^2 - \left(2\alpha(1-\gamma) - 4\alpha^2\right)\|V_{\theta^*} - V_{\theta_t}\|_D^2 \\ &= \|\theta^* - \theta_t\|_2^2 - \left(\frac{(1-\gamma)^2}{4}\right)\|V_{\theta^*} - V_{\theta_t}\|_D^2\end{aligned}$$

Now we need to relate the second term above back to $\|\theta - \theta_t\|_2^2$. We can use the definition of the linear approximator.

$$\|V_{\theta^*} - V_{\theta_t}\|_D^2 = \|\theta^* - \theta\|_\Sigma^2 = (\theta^* - \theta)^\top\Sigma(\theta^* - \theta) = \|\Sigma^{1/2}(\theta^* - \theta)\|_2^2 \geq \omega\|\theta^* - \theta\|_2^2$$

Now we can return back to our original proof. We get the following:

$$\begin{aligned}\|\theta^* - \theta_{t+1}\|_2^2 &\leq \|\theta^* - \theta_t\|_2^2 - \left(\frac{(1-\gamma)^2}{4}\right)\omega\|\theta^* - \theta_t\|_2^2 \\ &= \left(1 - \frac{\omega(1-\gamma)^2}{4}\right)\|\theta^* - \theta_t\|_2^2 \\ &\leq \exp\left\{-\frac{\omega(1-\gamma)^2}{4}\right\}\|\theta^* - \theta_t\|_2^2\end{aligned}$$

The last step is from using the inequality $\left(1 - \frac{\omega(1-\gamma)^2}{4}\right) \leq e^{\frac{-\omega(1-\gamma)^2}{4}}$. □

By repeatedly applying the theorem, we can get exponential convergence in the number of time-steps that we run TD-learning. Assume we run the algorithm for T steps. Inductively we get:

$$\|\theta^* - \theta_T\|_2^2 \leq \exp\left\{-\left(\frac{(1-\gamma)^2\omega}{4}\right)T\right\}\|\theta^* - \theta_0\|_2^2.$$

Note that there were many things we have not shown in this note that can be found in [1], including how people used to try to analyze TD-learning via contractions and how to extend to the Markovian sampling case.

References

[1] J. Bhandari, D. Russo, and R. Singal, "A finite time analysis of temporal difference learning with linear function approximation," in *Conference On Learning Theory*. PMLR, 2018, pp. 1691–1692.