Lecture 7: Value Functions

1 Plan for today

Admin:

- HW2 deadline extended until Feb 19.
- HW3 is still released today.
- When possible, please post on Slack non-anonymously. This is useful so that we can follow up offline (e.g., in office hours). Note that a tiny portion of your grade is based on (non-anonymous) participation.

1.1 Review

The policy gradient:

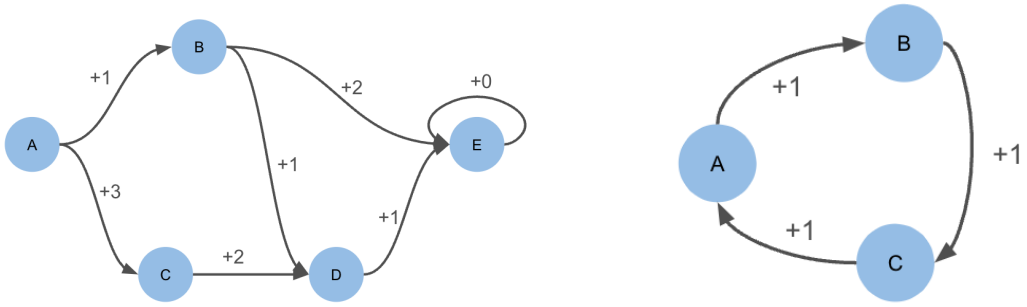
$$\nabla_\theta \mathbb{E}_{\pi_\theta(\tau)}[\sum_t \gamma^t r(s_t, a_t)] = \mathbb{E}_{s_t, a_t \sim \pi_\theta}\left[\Big(\underbrace{\sum_{t'=t}^{T} \gamma^{t'-t} r(s_t, a_t)}_{\text{prediction of future rewards}}\Big) \log \pi_\theta(a_t \mid s_t)\right]. \quad (1)$$

The quantity highlighted above is a prediction of the future rewards if you start at state s_t and take action a_t. Today, we're going to give a name to this sort of prediction. We'll see that there are ways of efficient ways of estimating this quantity, which is known as a value function. These improved estimators will give rise to much more effective RL algorithms.

Learning objectives By the end of today's lecture, you will be able to

1. explain what a value function is and why it might be useful.
2. derive common identities relating value functions, Q functions, and rewards.
3. compute the value function for tabular problems.
4. explain how the value function depends on the policy.

2 Computing the Future Expected Returns



(a) MDP 1 (b) MDP 2

Figure 1: Computing value functions.

1. In MDP 1, what is the future expected discounted return given that your current state is **E**?

2. In MDP 1, what is the future expected discounted return given that your current state is **C**?

3. In MDP 1, what is the future expected discounted return given that your current state is **B** and your current action is "Move Down"?

4. In MDP 1, what is the future expected discounted return given that your current state is **A** and your current action is "Move Up"?

5. In MDP 2 (right figure), what is the future expected discounted return given that your current state is **A**?

2.1 Considerations.

- How do you decide the order in which to update the nodes?
- So, it seems like there's some sort of recursive relationship. This should remind you of dynamic programming, and things like Dijkstra's algorithm.
- The last question highlights how you prediction for future values depends on what actions you'll take in the future, as well as on what action you take now.
- How do you do this when you don't have the graph? How to extend these ideas of dynamic programming to continuous or high-dim settings? This is nice on the graph we presented, but think about what the graph for Tetris or Go looks like.
- Moving forward, we will use a function to represent these Q-values and value functions, just like we did above.
- In the same way that we used the value function at one state to figure out the value function at a different state, we will use the *function* evaluated at one state to figure out the function evaluated at a different state.

3 Value Functions

In class so far, we've primarily been thinking about the *policy*. Today's class will introduce another key object, the value function. The policy and value function are very closely linked together; there's a formal sense in which one is the dual of the other [1]; they are two sides of the same coin. We'll start by defining terms and then explain some of the nice properties of these value functions

The Q-function $Q(s,a)$, also known as a Q-value, is a state-action-conditioned version of the RL objective. The value function $V(s)$ is a state-conditioned version of the RL objective.

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] \tag{2}$$

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a\right] \tag{3}$$

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s\right] \tag{4}$$

$$\tag{5}$$

Note that we've used the superscript "π" to label the Q functions and value functions. We've done this because these quantities depend on the policy. If you have a bad policy, you expect to get few rewards in the future, and your Q function and value function will be small. Conversely, if you have a great policy and your future returns will be high, and so your value function and Q function will also be high.

A note about names. "Value function" is sometimes used to refer to $V(s)$, or to either $V(s)$ or $Q(s,a)$. Sometimes the Q function is called the state-action value function.

Exercise: Using iterated expectation prove the following:

$$V^{\pi}(s) = \mathbb{E}_{\pi(a|s)}[Q^{\pi}(s,a)]. \tag{6}$$

Proof.

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s\right] \tag{7}$$

$$= \mathbb{E}_{\pi(a_0|s_0)}\left[\mathbb{E}_{p(s_1|s_0,a_0),\pi(a_1|s_1),\cdots}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s\right]\right] \tag{8}$$

$$= \mathbb{E}_{\pi(a_0|s_0)}\left[Q^{\pi}(s_0, a_0)\right]. \tag{9}$$

□

Note that when we use the superscript π on the left hand side, we end up seeing that same policy *(1)* for defining the expectation on the right hand side, and *(2)* in defining the Q value used on the right hand side.

Optimal policies. We will use $\pi^\star(a \mid s)$ to denote the optimal (i.e., reward-maximizing) policy. We will use $Q^\star(s,a)$ and $V^\star(s)$ to denote the corresponding Q function and value functions.

Identities:

$$V^\star(s) = \max_a Q^\star(s,a). \tag{10}$$

Bellman equation The Q function and value function satisfy some identities. These are pretty important, so it's worth memorizing them. There are a couple of variations of the Bellman equations, so I've put a few versions below.

$$Q^\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)\pi(a'|s')}\left[Q^\pi(s',a')\right] \tag{11}$$

$$Q^\star(s,a) = r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)}\left[\max_{a'} Q^\star(s',a')\right] \tag{12}$$

$$V^\pi(s) = \mathbb{E}_{\pi(a|s),p(s'|s,a)}\left[r(s,a) + \gamma V^\pi(s')\right] \tag{13}$$

$$\tag{14}$$

It is straightforward to proof these identities using the law of iterated expectations.

The Bellman equations are important because they will enable us to develop better ways of learning the value function. The key idea is that we'll be able to estimate the value at one time step by just looking one step into the future. This is as opposed to having to looking far into the future and counting up all the future rewards. Rather, we'll be able to just look at the reward at the current time step plus the predicted reward at the next time step.

4 Using the value function

In the next four lectures, we'll talk about how you can *learn* the value function. It will be a neural network that takes as input s or (s,a) and outputs a prediction (a scalar). For now, I want to preview how this function can be used, once learned.

The policy gradient Value functions give us a new way of thinking about the policy gradient.

$$\max_\pi \mathbb{E}_{\pi(a|s)}[Q(s,a)]. \tag{15}$$

We can optimize this using the tools that we saw last time. Namely, we can take the gradient of this objective w.r.t. the policy:

$$\nabla_\theta \mathbb{E}_{\pi_\theta(a|s)}[Q(s,a)] = \mathbb{E}_{\pi(a|s)}[Q(s,a)\nabla_\theta \log \pi_\theta(a \mid s)]. \tag{16}$$

Note that this looks very similar to policy gradient that we saw last time.

Thinking about the optimal policy

- Assume that your problem discrete actions (say, 5), and that each has a unique value (i.e., no repeated values). What does the optimal policy looks like?
- Assume that your actions are continuous, and that there is a unique maximum. What does the optimal policy look like?
- If you parametrize your policy as a Gaussian, $\pi_\theta(a \mid s) = \mathcal{N}(a; \mu_\theta(s), \sigma_\theta(s))$. When you do REINFORCE, what do you expect to happen to σ?

5 Outlook

The first several lectures in this class were focused on learning the policy. Today's lecture we shifted gears to start looking at the value function. These two objects are closely related to one another, with many practical algorithms alternating between updating a value function ("policy evaluation") and updating a policy ("policy improvement").

6 Solutions to Exercises

No actions (Fig. 1a) To start, let's assume that there are no actions, so we're just looking at the transitions $p(s' \mid s)$. When there are multiple paths, let's assume that we choose randomly.

$$V(E) = 0 \tag{17}$$
$$V(D) = 1 + \gamma V(E) = 1 \tag{18}$$
$$V(C) = 2 + \gamma V(D) = 2 + \gamma \tag{19}$$
$$V(B) = \tfrac{1}{2}(2 + \gamma V(E)) + \tfrac{1}{2}(1 + \gamma V(D)) \tag{20}$$
$$= \tfrac{1}{2}(2 + \gamma 0) + \tfrac{1}{2}(1 + \gamma 1) \tag{21}$$
$$= \tfrac{3}{2} + \tfrac{1}{2}\gamma \tag{22}$$
$$V(A) = \tfrac{1}{2}(1 + \gamma V(B)) + \tfrac{1}{2}(3 + \gamma V(C)) \tag{23}$$
$$= \tfrac{1}{2}(1 + \gamma(\tfrac{3}{2} + \tfrac{1}{2}\gamma)) + \tfrac{1}{2}(3 + \gamma(2 + \gamma)) \tag{24}$$
$$= \tfrac{1}{2} + \tfrac{3}{2} + (\tfrac{3}{4} + 1)\gamma + (\tfrac{1}{4} + \tfrac{1}{2})\gamma^2 \tag{25}$$
$$= 2 + \tfrac{7}{4}\gamma + \tfrac{3}{4}\gamma^2. \tag{26}$$

With actions (Fig 1b) Now, let's assume that actions allow the agent to choose between the outgoing edges. In this setting, let's figure out what the state-action value function looks like:

$$Q(E, \emptyset) = 0 \tag{27}$$
$$Q(D, \emptyset) = 1 + \gamma V(E) = 1 \tag{28}$$
$$Q(B, \text{right}) = 2 + \gamma V(E) = 2 \tag{29}$$
$$Q(B, \text{down}) = 1 + \gamma V(D) = 1 + \gamma \qquad \text{(Note that discount means we prefer "right.")}$$
$$Q(A, \text{right}) = 1 + \gamma V(B) \qquad \text{(But this requires knowing actions for B!)}$$
$$= 1 + \gamma Q(B, \text{right}) = 1 + 2\gamma. \tag{30}$$

6.1 Circle MDP (Fig. 1b)

We consider one more example. In the example above, it was clear where to start estimating the values, with $V(E)$. As shown in Fig. 1b, in some MDPs this isn't clear. In this setting, we can write down the identities from above:

$$V(A) = 1 + \gamma V(B) \tag{31}$$
$$V(B) = 1 + \gamma V(C) \tag{32}$$
$$V(C) = 1 + \gamma V(A). \tag{33}$$

Overloading notation to use $V = (V(A) \quad V(B) \quad V(C))$, we can write this as a system of linear equations:

$$V = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \gamma \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}}_{\triangleq P} V, \tag{34}$$

where we've introduced the matrix P to denote the transitions. We can solve this system of equations using linear algebra:

$$V = (I - \gamma P)^{-1} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{1-\gamma} \\ \frac{1}{1-\gamma} \\ \frac{1}{1-\gamma} \end{pmatrix}. \tag{35}$$

This example highlights the connections between value functions, linear algebra, and graphs.

References

[1] Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.