IE 498: Online Learning and Decision Making, Fall 2019 Oct 17, 2019

Lecture 16: Value Iteration, Policy Iteration and Policy Gradient

Lecturer: Tanmay Gangwani *Scribe: Dawei Li, Zikun Ye*

1 Recap and Overview

In the last lecture, we introduced the basic definitions of Markov Decision Process (MDP). In particular, the objective is

$$\eta(\pi) = \mathbb{E}_{\mu_0,\pi,T}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right],$$

where π is the policy, μ_0 is the start state distribution, and T is the transition distribution. In this lecture, we introduce algorithms to find

$$\pi^* = \arg\max_{\pi} \eta(\pi). \tag{1}$$

We will discuss a few methods to obtain π^*. These are value iteration (which uses the Bellman optimality operator to find V^*), policy iteration (which iteratively applies policy evaluation and policy improvement), and policy gradient methods (which directly obtain the gradient of (1) *w.r.t* policy parameters.

2 Dynamic Programming Method

As mentioned in the last lecture, the optimal value functions, V^* or Q^*, must satisfy the Bellman optimality equations:

$$\begin{aligned} V^*(s) &= \max_a \left[r(s,a) + \gamma \mathbb{E}_{s'\sim T(s'|s,a)} V^*(s')\right], \\ Q^*(s,a) &= \mathbb{E}_{s'\sim T(s'|s,a)}\left[r(s,a) + \gamma \max_{a'} Q^*(s',a')\right]. \end{aligned} \tag{2}$$

Therefore, by finding V^* or Q^* through the Bellman optimality equations (2), one can determine an optimal policy with realtive ease. Specifically, dynamic programming (DP) methods are obtained by simply turning (2) into an update rule.

2.1 Value Iteration

As a classical DP method, value iteration finds V^* by solving (2). The update rule can be written as a particularly simple backup operation:

$$V_{k+1}(s) = \max_a \left[r(s,a) + \gamma \mathbb{E}_{s'\sim T(s'|s,a)} V_k(s')\right].$$

For a more compact representation and a more convenient analysis, we introduce the Bellman optimality operator.

Definition 1. *A Bellman optimality operator $\mathcal{T} : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$ is an operator that satisfies: for any $V \in \mathbb{R}^{|S|}$,*

$$(\mathcal{T}V)(s) = \max_a \left[r(s,a) + \gamma \mathbb{E}_{s' \sim T(s'|s,a)} V(s') \right].$$

Value iteration can thus be represented as recursively applying the Bellman optimality operator:

$$V_{k+1} = \mathcal{T} V_k. \tag{3}$$

The Bellman optimality operator $\mathcal{T}$ has several excellent properties. It is easy to verify that V^* is a fixed point of $\mathcal{T}$, i.e., $\mathcal{T}V^* = V^*$. Another important property is that $\mathcal{T}$ is a contraction mapping.

Theorem 2. *$\mathcal{T}$ is a contraction mapping under sup-norm $\|\cdot\|_\infty$, i.e., there exists $\gamma \in [0,1)$ such that*

$$\|\mathcal{T}U - \mathcal{T}V\|_\infty \leq \gamma \|U - V\|_\infty, \forall U, V \in \mathbb{R}^{|S|}.$$

Proof. To prove this property, we need the following lemma:

Lemma 3.

$$\left| \max_a f(a) - \max_a g(a) \right| \leq \max_a |f(a) - g(a)|.$$

Lemma 3 is proved as follows. Assume without loss of generality that $\max_a f(a) \geq \max_a g(a)$, and denote $a^* = \arg\max_a f(a)$. Then,

$$\left| \max_a f(a) - \max_a g(a) \right| = \max_a f(a) - \max_a g(a) = f(a^*) - \max_a g(a) \leq f(a^*) - g(a^*) \leq \max_a |f(a) - g(a)|.$$

We now proceed to prove Theorem 2. For any state s, we have

$$\begin{aligned}
|\mathcal{T}V(s) - \mathcal{T}U(s)| &= \left| \max_a \left[r(s,a) + \gamma \mathbb{E}_{s' \sim T(s'|s,a)} V(s') \right] - \max_a \left[r(s,a) + \gamma \mathbb{E}_{s' \sim T(s'|s,a)} U(s') \right] \right| \\
&\leq \max_a \left| \gamma \mathbb{E}_{s' \sim T(s'|s,a)} \left[V(s') - U(s') \right] \right| \\
&\triangleq \left| \gamma \mathbb{E}_{s' \sim T(s'|s,a^*)} \left[V(s') - U(s') \right] \right| \quad \text{where, } a^* \text{ is the argmax of the RHS above} \\
&\leq \gamma \max_{s'} |V(s') - U(s')| \\
&= \gamma \|V - U\|_\infty
\end{aligned}$$

where the first inequality comes from Lemma 3. Since the above holds for any state s, it also holds for the state maximizing the LHS, such that:

$$\max_s |\mathcal{T}V(s) - \mathcal{T}U(s)| \leq \gamma \|V - U\|_\infty,$$

which means

$$\|\mathcal{T}U - \mathcal{T}V\|_\infty \leq \gamma \|V - U\|_\infty.$$

□

Given $\mathcal{T}$ is a contraction mapping, it is easy to prove the convergence of value iteration.

Theorem 4. *Value iteration (3) converges to V^*, i.e.,*

$$\lim_{k \to \infty} V_k = V^*,$$

where $V_k = \mathcal{T}^{k-1} V_0$.

Proof. Note that V^* is a fixed point of $\mathcal{T}$. In addition, according to Theorem 2, $\mathcal{T}$ is a contraction mapping. Therefore,

$$\|V_k - V^*\|_\infty = \|\mathcal{T}V_{k-1} - \mathcal{T}V^*\|_\infty \leq \gamma\|V_{k-1} - V^*\|_\infty \leq \cdots \leq \gamma^k\|V_0 - V^*\|_\infty.$$

Let $k \to \infty$, and we have $\|V_k - V^*\|_\infty \to 0$. Thus $\lim_{k\to\infty} V_k = V^*$. □

Furthermore, we can show that $\mathcal{T}$ has a *unique* fixed point. This can be proved by assuming two different fixed points, and using the contraction property to arrive at a contradiction *(left as exercise)*.

Other Bellman operators can be defined in an analogous manner. For instance, if we fix the policy π in the operator definition, we can construct the Bellman expectation operator $\mathcal{T}^\pi$.

Definition 5. *A Bellman expectation operator for policy π, or $\mathcal{T}^\pi : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$ is an operator that satisfies: for any $V \in \mathbb{R}^{|S|}$,*

$$(\mathcal{T}^\pi V)(s) = \mathbb{E}_{a\sim\pi(a|s),s'\sim T(s'|s,a)}\left[r(s,a) + \gamma V^\pi(s')\right].$$

$\mathcal{T}^\pi$ is also a contraction mapping, and the unique fixed point of $\mathcal{T}^\pi$ is V^π. The proof technique is very similar and is thus omitted. Bellman operators that act on the Q function (instead of V) are similarly constructed, and also have the contraction and convergence properties.

2.2 Policy Iteration

Another method to solve (2) is policy iteration, which iteratively applies policy evaluation and policy improvement, and converges to the optimal policy. Compared to value-iteration that finds V^*, policy iteration finds Q^* instead. A detailed algorithm is given below.

Algorithm 1 Policy Iteration

```
Randomly initialize policy $\pi_0$
for each $k = 0, 1, 2, ..., \infty$ do
    $Q^{\pi_k} \leftarrow$ Policy evaluation with $\pi_k$;
    Policy improvement: $\pi_{k+1} = G(Q^{\pi_k})$;
end for
```

Here G represents the operator that carries out policy improvement. Examples of G include:

1. The greedy method, defined as

$$\pi_{k+1}(a|s) = \begin{cases} 1, & a = \arg\max Q^{\pi_k}(s,a), \\ 0, & o.w. \end{cases} \tag{4}$$

2. The ϵ-greedy method, defined as

$$\pi_{k+1}(a|s) = \begin{cases} \frac{\epsilon}{|A|} + 1 - \epsilon, & a = \arg\max Q^{\pi_k}(s,a), \\ \frac{\epsilon}{|A|}, & o.w. \end{cases} \tag{5}$$

   where $|A|$ refers to the number of actions in the action space. Compared to the greedy method, the ϵ-greedy method helps in exploring the action space.

We now show that policy iteration (Algorithm 1) with greedy policy improvement converges to Q^* and π^*. To that end, we first prove an important theorem known as the Policy Improvement Theorem.

Theorem 6. ***Policy Improvement Theorem:*** *Consider two policy $\pi(a|s)$, $\pi'(a|s)$, and define*

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

If $\forall s \in S$, we have that $Q^\pi(s,\pi') \geq V^\pi(s)$, then it holds that $V^{\pi'}(s) \geq V^\pi(s)$, $\forall s \in S$. This means that π' is atleast as good a policy as π.

Proof. Note that $Q^\pi(s,\pi') \geq V^\pi(s)$. By expanding Q^π, we can get that $\forall s \in S$,

$$\begin{aligned}
V^\pi(s) &\leq Q^\pi(s,\pi') \\
&= \mathbb{E}_{a\sim\pi'(a|s), s'\sim\mathcal{T}'(s'|s,a)}[r(s,a) + \gamma V^\pi(s')] \\
&\leq \mathbb{E}_{a\sim\pi'(a|s), s'\sim\mathcal{T}'(s'|s,a)}[r(s,a) + \gamma Q^\pi(s',\pi')] \\
&= \mathbb{E}_{a,a'\sim\pi'}[r(s,a) + \gamma r(s',a') + \gamma^2 V^\pi(s'')] \\
&\leq ... \\
&\leq \mathbb{E}_{a,a',a''...\sim\pi'}[r(s,a) + \gamma r(s',a') + \gamma^2 r(s'',a'') + ...] \\
&= V^{\pi'}(s)
\end{aligned}$$

□

Now, we prove monotone improvement with the policy-iteration (Algorithm 1) using the greedy policy improvement. Note that by definition:

$$Q^{\pi_k}(s,\pi_{k+1}) = \max_a Q^{\pi_k}(s,a)$$
$$V^{\pi_k}(s) = \mathbb{E}_{a\sim\pi_k(a|s)}[Q^{\pi_k}(s,a)]$$

Therefore, $Q^{\pi_k}(s,\pi_{k+1}) \geq V^{\pi_k}(s), \forall s \in S$. This satisfies the requirement of the policy improvement theorem, and hence, $V^{\pi_{k+1}}(s) \geq V^{\pi_k}(s), \forall s \in S$. To show convergence to the optimal policy, along with monotone improvement, we need to show that if there is no improvement in the value function at any state, then we are at optimality. The proof sketch is as follows. We consider k such that $V^{\pi_{k+1}}(s) = V^{\pi_k}(s), \forall s \in S$. We can show that such V^{π_k} satisfies the Bellman optimality equation, and hence $V^{\pi_k} = V^*$ *(left as exercise).*

3 Policy Gradient

Policy gradient methods aim at modeling and optimizing the policy directly, but estimating the gradient of the expected cumulative return. For example, in applications with neural networks, policy is typically parameterized by θ, $\pi_\theta(a|s)$. Then the objective function can be rewritten as,

$$\eta(\pi_\theta) = \mathbb{E}_{\mu_0,\pi_\theta,T}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right]$$

To illustrate the idea of policy gradients, we rewrite the objective in terms of trajectories generated using π_θ. A trajectory is defined as $\tau = (s_0, a_0, r_0, s_1, ...)$. $P_\theta(\tau) = \mu_0(s_0)\prod_t \pi_\theta(a_t|s_t)T(s_{t+1}|s_t,a_t)$ is the probability of the trajectory decomposed using the MDP assumptions, and $R(\tau) = \sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)$ is the discounted return with given trajectory τ. Then,

$$
\begin{aligned}
\eta(\pi_\theta) &= \mathbb{E}_{\mu_0,\pi,T}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] \\
&= \mathbb{E}_{\tau\sim P_\theta(\tau)}\left[R(\tau)\right] \\
\nabla_\theta\eta(\pi_\theta) &= \sum_\tau \nabla_\theta P_\theta(\tau)R(\tau) \\
&= \mathbb{E}_{\tau\sim P_\theta(\tau)}[\nabla_\theta \log P_\theta(\tau)R(\tau)] \\
&= \mathbb{E}_{\tau\sim P_\theta(\tau)}[\sum_{t=0}^{\infty}\nabla_\theta \log \pi_\theta(a_t|s_t)R(\tau)] \\
&= \mathbb{E}_{\tau\sim P_\theta(\tau)}[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta \log \pi_\theta(a_t|s_t)R_t]
\end{aligned}
$$

where R_t is the discounted sum of rewards from time-step t onwards. The policy gradient $\nabla_\theta\eta(\pi_\theta)$ can also be obtained in terms of the unnormalized discounted occupancy measure $\hat{\rho}(s,a)$ (defined in last lecture).

Theorem 7. ***Policy Gradient Theorem:*** *For a differentiable policy π_θ, the policy gradient is*

$$\nabla_\theta\eta(\pi_\theta) = \mathbb{E}_{s,a\sim\hat{\rho}_\theta(s,a)}[\nabla_\theta \log \pi_\theta(a|s)Q^{\pi_\theta}(s,a)]$$