MS&E338 Reinforcement Learning Lecture 3 - 4/9/18

Real Time Value Iteration and the State-Action Value Function

Lecturer: Ben Van Roy *Scribe: Apoorva Sharma and Tong Mu*

1 Review

Last time we left off discussing the Asynchronous Value Iteration algorithm:

Algorithm 1: Asynchronous Value Iteration

Start with V_0;
for *k = 0, 1, 2, ...* **do**
  select s_k;
  $V_{k+1}(s) = \begin{cases} (TV_k)(s), & \text{if } s = s_k \\ V_k(s), & \text{otherwise} \end{cases}$;
end

Proposition 1. *If each $s \in S$ is selected infinitely often, then $V_k \to V^*$*

2 Greedy Policy

Definition 1. *π is greedy w.r.t. V if $T_\pi V = TV$.*

Proposition 2. *If π is greedy w.r.t V, then $\|V^* - V^\pi\|_{\infty,\frac{1}{\tau}} \leq \frac{2\alpha}{1-\alpha} \|V^* - V\|_{\infty,\frac{1}{\tau}}$ where $\alpha = \max_s \frac{\tau(s)-1}{\tau(s)}$ as defined in the last lecture.*

This implies that if the value functions is getting close to optimal, then the policy from V is also getting close to optimal.

Proof.

$$\begin{aligned} \|V^* - V^\pi\| &= \|V^* - TV + T_\pi V - V^\pi\| \\ &\leq \|V^* - TV\| + \|T_\pi V - V^\pi\| \quad \text{by the triangle inequality} \\ &\leq \alpha \|V^* - V\| + \alpha \|V - V^\pi\| \quad \text{by contraction} \\ &\leq \alpha \|V^* - V\| + \alpha \|V - V^* + V^* - V^\pi\| \\ &\leq 2\alpha \|V^* - V\| + \alpha \|V^* - V^\pi\| \quad \text{by the triangle inequality} \end{aligned}$$

Shifting and dividing, we get the desired result

$$\|V^* - V^\pi\| \leq \frac{2\alpha}{1-\alpha} \|V^* - V\|.$$

□

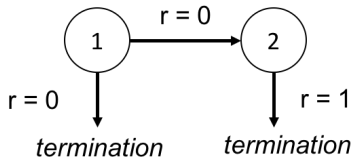

Figure 1: Example MDP

3 Real Time Value Iteration

With this algorithm, we move one step closer to Reinforcement Learning. The agent simulates the MDP and generates actions until termination (episodes). We still assume we know the rewards, $\bar{R}(s,a)$ and the transition dynamics $P(s,a)$. We will index the value function by two indices, the first index will be the episode index ℓ and the second index will be the time index t which represents the timestep within the episode. We define $\pi_{l,t}$ as the greedy policy w.r.t $V_{l,t}$ and $\bar{R}(s,a) = E[r_{t+1}|s_t = s, a_t = a]$.

Algorithm 2: Real Time Value Iteration

Start with $V_{0,0}$;
for $\ell = 0, 1, 2, \ldots$ *(episode)* **do**
  sample $s_0 \sim \rho$;
  for t = *0, 1, 2, .., $\tau - 1$ (time)* **do**
    select $a_t \in \arg\max_{a \in \mathcal{A}}(\bar{R}(s_t, a) + \sum_{s' \in \mathcal{S}} P_{s_t,a}(s') V_{\ell,t}(s'))$;
    sample $s_{t+1} \sim P_{s_t,a_t}$;
    $V_{\ell,t+1}(s) = \begin{cases} (TV_{\ell,t})(s), & \text{if } s = s_{t+1} \\ V_{\ell,t}(s), & \text{otherwise} \end{cases}$;
  end
  $V_{\ell+1,0} = V_{\ell,\tau}$
end

This algorithm is not guarranted to work. To show this, we will provide a negative example where the optimal value function V^* is not learned even as $l \to \infty$.

Consider the MDP in Figure 1 with the available actions represented by arrows. Actions are deterministic. Suppose we always start in state 1 ($\rho = [1, 0]$) and we initialize our values with $V_{0,0}(1) = 0$ and $V_{0,0}(2) = -1$. In this case, starting from state 1, we will always choose the action that leads to termination because a reward of 0 is better than the believed reward of -1 if we transition to state 2, so we are unable to learn the optimal value function.

However under certain conditions this can perform well:

Proposition 3. *If $V_{0,0} \geq V^*$ (optimism)*
then (1) $\forall s \in S$ visited i.o. $V_{\ell,t}(s) \to V^*(s)$
(2) $\forall s \in S$ visited i.o. $V^{\pi_{\ell,t}}(s) \to V^*(s)$

The second point implies the following as a corollary:

Corollary 2. $\sum_{s \in S} \rho(s) V^{\pi_{\ell,t}}(s) \to \sum_{s \in S} \rho(s) V^*(s)$

We now prove the proposition:

Proof. Let $\bar{S} = \{s \in S : s \text{ is visited i.o}\}$. Let $\bar{A}_s = \{a \in A : P_{s,a}(s') = 0 \ \forall \ s' \notin \bar{S}\}$.

Let l_0 be such that states are in $\bar{S}$ and actions are in $\bar{A}$ for all $l \geq l_0$. Note that this point must exist because we are simulating infinitely many times, if this point does not exist, then we will consistently take actions that lead us to states outside of $\bar{S}$ and visit those states infinitely often as well, resulting in them becoming part of $\bar{S}$ as well.

Consider the surrogate MDP $(\bar{S}, \bar{A}, R, P, \rho)$. Note that this surrogate MDP is a limited version of the original MDP in which all states are visited infinitely often, and thus we know asynchronous value iteration will converge for this altered MDP.

$$\begin{aligned}
V_{\ell,t+1}(s_t) &= (TV_{\ell,t})(s_t) \\
&= \max_{a \in A}(\bar{R}(s_t, a) + \sum_{s' \in S} P_{s_t,a}(s')V_{\ell,t}(s')) \\
&= \bar{R}(s_t, a_t) + \sum_{s' \in S} P_{s_t,a}(s')V_{\ell,t}(s') \quad \text{as } a_t \text{ taken by the algorithm is the argmax} \\
&= \max_{a \in \bar{A}_{s_t}} (\bar{R}(s_t, a) + \sum_{s' \in S} P_{s_t,a}(s')V_{\ell,t}(s')) \\
&= (\bar{T}V_{\ell,t})(s_t)
\end{aligned}$$

Therefore $V_{l,t} \to \bar{V}(s) \ \forall \ s \in \bar{S}$.

If $V_{0,0} \geq V^*$ then $TV_{0,0} \geq TV^* \to TV^* = V^*$ so by monotonicity $V_{l,t} \geq V^* \to \bar{V}(s) \geq V^*(s) \ \forall \ s \in \bar{S}$

Additionally, we can apply this policy from $\bar{V}$ to the original MDP because $\bar{S} \subset \mathcal{S}$ so $\bar{V}(s)$ is attainable in the original MDP and $\bar{V}(s) \leq V^*(s) \ \forall \ s \in \bar{S}$ as we are only looking at a subset of all the possible states.

Combining these results, we have:

attainability $\to \bar{V}(s) \leq V^*(s) \ \forall \ s \in \bar{S}$
monotonicity $\to \bar{V}(s) \geq V^*(s) \ \forall \ s \in \bar{S}$

Combining these we get $\bar{V}(s) = V^*(s) \ \forall \ s \in \bar{S}$.

□

4 The State-Action Value Function

We have up to now worked with the value function $V : \mathcal{S} \to \mathbb{R}$. We can define a similar quantity $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which is the state-action value function, often called a Q function. The optimal state-action value function is defined as:

$$Q^*(s, a) = \bar{R}(s, a) + \sum_{s' \in \mathcal{S}} P_{s,a}(s')V^*(s') \tag{1}$$

The state-action value function for a particular policy π is defined similarly by the expression below:

$$Q^\pi(s, a) = \bar{R}(s, a) + \sum_{s' \in \mathcal{S}} P_{s,a}(s')V^\pi(s') \tag{2}$$

We notice that $V^*(s) = \max_{a \in \mathcal{A}} Q^*(s, a)$, so the optimal state-action value function satisfies the **Bellman Equation:**

$$Q^*(s, a) = \bar{R}(s, a) + \sum_{s' \in \mathcal{S}} P_{s,a}(s') \max_{a' \in \mathcal{A}} Q^*(s', a') \tag{3}$$

We can thus also define the **Bellman Operator** for the state-action value function as F where

$$(FQ)(s,a) = \bar{R}(s,a) + \sum_{s' \in \mathcal{S}} P_{s,a}(s') \max_{a' \in \mathcal{A}} Q(s',a') \tag{4}$$

Just like the Bellman operator for the value function T, this operator F has the monotonicity and contraction properties.

$$\begin{aligned}
\textbf{Monotonicity:} \quad & \forall Q, Q', \quad \text{if} \;\; Q \leq Q', \;\; FQ \leq FQ' \\
\textbf{Contraction:} \quad & \forall Q, Q', \quad ||FQ - FQ'||_{\infty,1/\tau} \leq \alpha ||Q - Q'||_{\infty,1/\tau} \\
& \text{where} \;\; ||Q||_{\infty,1/\tau} = \max_{s \in \mathcal{S}, a \in \mathcal{A}} \frac{|Q(s,a)|}{\tau(s)} \\
& \alpha = \max_{s \in \mathcal{S}} \frac{\tau(s) - 1}{\tau(s)}
\end{aligned}$$

While the definition of the state-action value function $Q(s,a)$ the associated bellman equation and operator follow relatively straightforwardly from their counterparts for the value function $V(s)$, we have not yet addressed why we might care about this function. There are a few reasons why using the state-action value function is convenient:

- **Greedy action selection is simple.**
  For a policy to be greedy w.r.t. a state-action value function Q, the policy simply becomes the maximization:
  $$\pi(s) = \arg\max_{a \in \mathcal{A}} Q(s,a) \tag{5}$$

- **Obtaining unbiased samples of FQ is easier than obtaining unbiased samples of TV.**
  Notice that we can write the Bellman Operator F as as expectation:

  $$FQ(s,a) = \mathbb{E}[r_{t+1} + \max_{a' \in \mathcal{A}} Q(s_{t+1}, a') \mid s_t = s, a_t = a].$$

  Thus, the quantity

  $$r_{t+1} + \max_{a' \in \mathcal{A}} Q(s_{t+1}, a')$$

  is an unbiased sample of $FQ(s,a)$ if $s_{t+1} \sim P_{s,a}$, and $r_{t+1} \sim R_{s,a,s_{t+1}}$.

  In contrast, recall that

  $$TV = \max_{a \in \mathcal{A}} \mathbb{E}[r_{t+1} + V(s_{t+1}) | s_t = s, a_t = a],$$

  which has the maximization outside the expectation. Therefore, the quantity

  $$\max_{a \in \mathcal{A}} (r_{t+1} + V(s_{t+1}))$$

  where $s_{t+1} \sim P_{s,a}$ and $r_{t+1} \sim R_{s,a,s_{t+1}}$ *does not* yield an unbiased sample.

4.1 Asynchronous State-Action Value Iteration

Similar to Asynchronous Value Iteration, we can perform updates to an arbitrary function Q by sampling (s,a) and then updating the function's value at that point by the rule:

$$Q(s,a) = (FQ)(s,a). \tag{6}$$

If every $(s,a) \in \mathcal{S} \times \mathcal{A}$ is selected infinitely often, then $Q \to Q^*$.

4.2 Q-learning

Rather than computing the expectation in F exactly, we can approximate it with a single unbiased sample, as described earlier. This yields what is known as the Q learning update:

$$Q_{k+1}(s,a) := (1-\gamma_k)Q_k(s,a) + \gamma_k(r + \max_{a' \in \mathcal{A}} Q_k(s',a')) \tag{7}$$

$$s' \sim P_{s,a} \tag{8}$$

$$r \sim R_{s,a,s'} \tag{9}$$

This update is sometimes equivalently written as:

$$Q_{k+1}(s,a) := Q_k(s,a) + \gamma_k(r + \max_{a' \in \mathcal{A}} Q_k(s',a') - Q_k(s,a)) \tag{10}$$

If each $(s,a) \in \mathcal{S} \times \mathcal{A}$ are updated infinitely often, $\sum_{k=1}^{\infty} \gamma_k = \infty$, and $\sum_{k=1}^{\infty} \gamma_k^2 < \infty$, then $Q \to Q^*$.