ECE 598: Interplay between Control and Machine Learning **Fall 2020**

Lecture 9

Optimal Bellman Equation and Value Iteration

Lecturer: Bin Hu, Date:10/01/2020

In this lecture, we discuss how to design the optimal policy. The key concept is the optimal Bellman equation.

9.1 Discrete Space Case

Recall that 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. Then the goal is to find an optimal policy that maximizes the total accumulated rewards, i.e.

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

Suppose the value of π^* is V^*. Then the optimal Bellman equation states that the optimal value function satisfies

$$V^*(s) = \max_{a \in \mathcal{A}} \left[\bar{R}(s,a) + \gamma \sum_{s'} P_{ss'}^a V^*(s')\right] \tag{9.1}$$

We can compare the above equation with the Bellman equation for a fixed deterministic policy π:

$$V^\pi(s) = \bar{R}(s, \pi(s)) + \gamma \sum_{s'} P_{ss'}^{\pi(s)} V^\pi(s') \tag{9.2}$$

Obviously the optimal Bellman equation is nonlinear since it involves a minimization over $a \in \mathcal{A}$. For each $s \in \mathcal{S}$, one has to choose a such that the term $\left[\bar{R}(s,a) + \gamma \sum_{s'} P_{ss'}^a V^*(s')\right]$ is minimized. Clearly such an optimal action exists since there is only a finite number of possible actions. As a matter of fact, such an optimal action is a function of s and this one-to-one mapping directly gives the optimal policy. Such a policy is deterministic and we have

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \left[\bar{R}(s,a) + \gamma \sum_{s'} P_{ss'}^a V^*(s')\right] \tag{9.3}$$

Now we discuss how to solve the optimal Bellman equation. A fundamental algorithm is the so-called value iteration (VI) method. VI is an iterative algorithm which solves V^*

by recursively applying the Bellman operator T which maps any $V \in \mathbb{R}^n$ to another vector whose i-th element is equal to $\max_{a\in\mathcal{A}} \left[\bar{R}(s,a) + \gamma \sum_{s'} P^a_{ss'} V^*(s')\right]$. The VI algorithm can be compactly written as $V_{k+1} = T(V_k)$. This is equivalent to

$$V_{k+1}(s) = \max_{a\in\mathcal{A}} \left[\bar{R}(s,a) + \gamma \sum_{s'} P^a_{ss'} V_k(s')\right]$$

One can show that the Bellman operator is a contraction mapping $\|T(V) - T(V')\|_\infty \leq \gamma \|V - V'\|_\infty$ for any (V, V'). Therefore, by the famous Banach fixed point theorem, we know V_k converges to V^* at a linear rate γ.

9.2 Continuous Space Case

9.2.1 LQR without Process Noise

For simplicity, again let's first consider the LQR setup:

$$x_{t+1} = Ax_t + Bu_t \tag{9.4}$$

The goal is to design a controller to minimize the quadratic cost.

For the above LQR problem, the optimal Bellman equation becomes the following Riccati equation

$$P = A^\mathsf{T} PA + Q - A^\mathsf{T} PB(B^\mathsf{T} PB + R)^{-1} B^\mathsf{T} PA \tag{9.5}$$

When (A, B) is stabilizable, the above equation has a unique positive definite[1] stabilizing solution P such that $\rho(A - B(B^\mathsf{T}PB + R)^{-1}B^\mathsf{T}PA) < 1$. Then the optimal action is given by a linear policy, i.e. $u_t = -(B^\mathsf{T}PB + R)^{-1}B^\mathsf{T}PAx_t$. This linear policy is optimal among all (potentially nonlinear) policies. This can be shown using completion of square. We can rewrite the cost as follows.

$$\begin{aligned}
&\sum_{t=0}^{\infty} (x_t^\mathsf{T} Q x_t + u_t^\mathsf{T} R u_t) \\
=& x_0^\mathsf{T} P x_0 + \sum_{t=0}^{\infty} (x_t^\mathsf{T} Q x_t + u_t^\mathsf{T} R u_t + x_{t+1}^\mathsf{T} P x_{t+1} - x_t^\mathsf{T} P x_t) \\
=& x_0^\mathsf{T} P x_0 + \sum_{t=0}^{\infty} \left(x_t^\mathsf{T} Q x_t + u_t^\mathsf{T} R u_t + (Ax_t + Bu_t)^\mathsf{T} P (Ax_t + Bu_t) - x_t^\mathsf{T} P x_t\right) \\
=& x_0^\mathsf{T} P x_0 + \sum_{t=0}^{\infty} \left(x_t^\mathsf{T} (Q + A^\mathsf{T} PA - P) x_t + u_t^\mathsf{T} (R + B^\mathsf{T} PB) u_t + x_t^\mathsf{T} A^\mathsf{T} PB u_t + u_t^\mathsf{T} B^\mathsf{T} PA x_t\right)
\end{aligned}$$

[1] For simplicity, we assume Q is positive definite in this note. Hence the solution P is positive definite. In general, it is possible to allow Q to be only positive semidefinite and the solution P can also become positive semidefinite. Some extra observability assumption is then needed.

By the Riccati equation, we have $Q + A^\top PA - P = A^\top PB(B^\top PB + R)^{-1}B^\top PA$. Hence we have

$$\begin{aligned}
&\sum_{t=0}^{\infty}(x_t^\top Qx_t + u_t^\top Ru_t)\\
=&x_0^\top Px_0 + \sum_{t=0}^{\infty}\left(x_t^\top A^\top PB(B^\top PB + R)^{-1}B^\top PAx_t + u_t^\top(R + B^\top PB)u_t + x_t^\top A^\top PBu_t + u_t^\top B^\top PAx_t\right)\\
=&x_0^\top Px_0 + \sum_{t=0}^{\infty}\left((B^\top PB + R)u_t + B^\top PAx_t\right)^\top (B^\top PB + R)^{-1}\left((B^\top PB + R)u_t + B^\top PAx_t\right)
\end{aligned}$$

Notice $B^\top PB + R$ is positive definite. Therefore, the cost achieves its minimal value $x_0^\top Px_0$ if the action is chosen to satisfy $(B^\top PB + R)u_t + B^\top PAx_t = 0$. We have skipped a few technical details but the above calculation roughly explain why the optimal policy is given by $K = (B^\top PB + R)^{-1}B^\top PA$.

How does the Riccati equation arise? Usually one first looks at LQR on a finite horizon and then extends the result there to the infinite horizon setup. Instead of doing that, we will provide an alternative informal derivation that starts from the assumption that the optimal policy is linear. Under this assumption, the optimal value function is quadratic and takes the form of $x^\top Px$. The optimal Bellman equation states the following fact.

$$x^\top Px = \min_u(x^\top Qx + u^\top Ru + (Ax + Bu)^\top P(Ax + Bu)) \tag{9.6}$$

Clearly, the right side is a quadratic function in u. Taking the gradient of the right hand side with respect to u leads to the following equation:

$$(R + B^\top PB)u + B^\top PAx = 0$$

The optimal u is given as $u = -(R + B^\top PB)^{-1}B^\top PAx$. We can substitute this relation to the right side of (9.6) and obtain

$$\begin{aligned}
P = &Q + A^\top PB(R + B^\top PB)^{-1}R(R + B^\top PB)^{-1}B^\top PA\\
&+ \left(A - B(B^\top PB + R)^{-1}B^\top PA\right)^\top P\left(A - B(B^\top PB + R)^{-1}B^\top PA\right)
\end{aligned}$$

Simplifying the above equation leads to the Riccati equation (9.5).

Key message. When writing out the optimal Bellman equation, we need to know how to parameterize the optimal value function. In the LQR problem, we know we can parameterize the optimal value function as a quadratic function. Therefore, eventually the optimal Bellman equation (9.6) is equivalent to an algebraic Riccati equation on P. If we just use a general value function $V(x)$, then (9.6) becomes

$$V(x) = \min_u(x^\top Qx + u^\top Ru + V(Ax + Bu)) \tag{9.7}$$

How to figure out V becomes a difficult task. For nonlinear control, the situation is even worse. Suppose the dynamics become $x_{t+1} = f(x_t, u_t)$ and the one-stage cost is $c(x, u)$ where c is some complicated function. Then the optimal Bellman equation is in the following nonlinear form:

$$V(x) = \min_u (c(x, u) + V(f(x, u))) \tag{9.8}$$

which is extremely difficult to solve. One has to approximate V using neural networks.

Value Iteration. Suppose we decide to parameterize the value function as $V(x) = x^\mathsf{T} P x$. The Bellman operator T maps P to P' as follows

$$x^\mathsf{T} P' x = \min_u (x^\mathsf{T} Q x + u^\mathsf{T} R u + (Ax + Bu)^\mathsf{T} P (Ax + Bu)) \tag{9.9}$$

Clearly, the optimal Bellman equation (9.6) is equivalent to $P = T(P)$ and just gives the fixed point for the above Bellman operator. Again, the right side of (9.9) is a quadratic function of u, and minimizing over u gives

$$P' = A^\mathsf{T} P A + Q - A^\mathsf{T} P B (B^\mathsf{T} P B + R)^{-1} B^\mathsf{T} P A \tag{9.10}$$

Therefore, we can apply the Bellman operator recursively and obtain a value iteration algorithm:

$$P^{n+1} = A^\mathsf{T} P^n A + Q - A^\mathsf{T} P^n B (B^\mathsf{T} P^n B + R)^{-1} B^\mathsf{T} P^n A \tag{9.11}$$

We can clearly see, for the continuous state space case, we need to parameterize the value function and then iteratively update these parameters.

9.2.2 LQR with Process Noise

In this case, the dynamics become

$$x_{t+1} = Ax_t + Bu_t + w_t \tag{9.12}$$

where w_t is an IID Gaussian noise.

1. Optimal Bellman equation: Suppose the optimal state value function is $x^\mathsf{T} P x + r$. We have

$$\begin{aligned} x^\mathsf{T} P x + r &= \min_u (x^\mathsf{T} Q x + u^\mathsf{T} R u + \gamma \mathbb{E}(Ax + Bu + w)^\mathsf{T} P (Ax + Bu + w) + \gamma r) \\ &= \min_u (x^\mathsf{T} Q x + u^\mathsf{T} R u + \gamma (Ax + Bu)^\mathsf{T} P (Ax + Bu) + \gamma \mathbb{E} w^\mathsf{T} P w + \gamma r) \end{aligned}$$

Taking gradient of the function on the right side with respect to u leads to

$$u = -\gamma (R + \gamma B^\mathsf{T} P B)^{-1} B^\mathsf{T} P A x.$$

which can be substituted b back to get the following optimal Bellman equation:

$$P = Q + \gamma A^{\mathsf{T}} P A - \gamma^2 A^{\mathsf{T}} P B (R + \gamma B^{\mathsf{T}} P B)^{-1} B^{\mathsf{T}} P A$$

Then the optimal state-action value function can be calculated as

$$\mathcal{Q}^*(x,u) = \begin{bmatrix} x \\ u \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} Q + \gamma A^{\mathsf{T}} P A & \gamma A^{\mathsf{T}} P B \\ \gamma B^{\mathsf{T}} P A & R + \gamma B^{\mathsf{T}} P B \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \frac{\gamma}{1-\gamma} \operatorname{trace}(PW)$$

2. Value iteration: Algorithm (9.11) should be modified as

$$P^{n+1} = \gamma A^{\mathsf{T}} P^n A + Q - \gamma^2 A^{\mathsf{T}} P^n B (\gamma B^{\mathsf{T}} P^n B + R)^{-1} B^{\mathsf{T}} P^n A \tag{9.13}$$