Introduction to Reinforcement Learning: Session II

Dr. Mayank S JHA

Maitre de Conférences (Associate Professor),
CRAN,
UMR 7039, CNRS,
Polytech Nancy,
Office: C 225,
Université de Lorraine
France.

Table of Contents

Table of Contents

1. Dynamic Programming
2. Bellman Equation and Bellman Optimality Equation
3. Discrete Time Linear System

Optimal Cost

The optimal cost can be written as

$$\begin{aligned} V_k^*(x) &= \min_{\pi} V_k^{\pi}(x) \\ &= \min_{\pi} \sum_{u} \pi(x,u) \sum_{x'} P_{xx'}^{u} \left[R_{xx'}^{u} + \gamma V_{k+1}^{\pi}\left(x'\right) \right] \end{aligned}$$

Bellman Optimality Principle

Bellman Principle

”An optimal policy has the property that no matter what the previous control actions have been, the remaining controls constitute an optimal policy with regard to the state resulting from those previous controls.”

Bellman Optimality Equation

This principle implies **Optimal Cost** can be written as

Bellman Optimality Equation

$$V_k{}^*(x) = \min_{\pi} \sum_{u} \pi(x, u) \sum_{x'} P^{u}_{xx'} \left[R^{u}_{xx'} + \gamma V^{*}_{k+1} \left(x' \right) \right]$$

Suppose an arbitrary control u is now applied at time k, and the optimal policy is applied from time $k + 1 \mathrm{on}$. Then Bellman's optimality principle indicates that the optimal control policy at time k is given by

$$\pi^*(x, u) = \arg\min_{\pi} \sum_{u} \pi(x, u) \sum_{x'} P^{u}_{xx'} \left[R^{u}_{xx'} + \gamma V^{*}_{k+1} \left(x' \right) \right]$$

Optimal Cost

Assumptions:

- Markov chain corresponding to each policy, with transition probabilities, is ergodic,
- every MDP has a stationary deterministic optimal policy. Minimize the conditional expectation over all actions u in state x.

$$V_k^*(x) = \min_u \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma V_{k+1}^* \left(x' \right) \right]$$

$$u_k^* = \arg \min_u \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma V_{k+1}^* \left(x' \right) \right]$$

Dynamic Programming: Key Points

- The backward recursion forms the basis for dynamic programming.
- **Offline methods** for working backward in time to determine optimal policies.
- DP is an o**ffline procedure for finding the optimal value and optimal policies t**hat requires knowledge of the complete system dynamics in the form of transition probabilities $P^{u}_{x,x'} = \Pr\{x' \mid x, u\}$ and expected costs $R^{u}_{xx'} = E\{r_k \mid x_k = x, u_k = u, x_{k+1} = x'\}$.

Table of Contents

Need for *Forward-in-Time* procedures

- Dynamic programming is a backward-in-time method for finding the optimal value and policy.
- By contrast, reinforcement learning is concerned with finding optimal policies based on causal experience by executing sequential decisions that improve control actions based on the observed results of using a current policy.
- This procedure requires the derivation of methods for finding optimal values and optimal policies that can be executed forward in time.

Infinite Horizon cost

Set the time horizon T to infinity and define the infinite-horizon cost

$$J_k = \sum_{i=0}^{\infty} \gamma^i r_{k+i} = \sum_{i=k}^{\infty} \gamma^{i-k} r_i$$

The associated infinite-horizon value function for the policy $\pi(x, u)$ is

$$V^\pi(x) = E_\pi \{J_k \mid x_k = x\} = E_\pi \left\{ \sum_{i=k}^{\infty} \gamma^{i-k} r_i \mid x_k = x \right\}$$

Bellman Equation

With $T = \infty$, it is seen that the value function for the policy $\pi(x, u)$ satisfies the Bellman equation

$$V^{\pi}(x) = \sum_{u} \pi(x, u) \sum_{x'} P_{xx'}^{u} \left[R_{xx'}^{u} + \gamma V^{\pi}\left(x'\right) \right]$$

Bellman Eq. Observations

- Consistency equation that must be satisfied by the value function at each time stage.
- It expresses a relation between the current value of being in state x and the value of being in next state x' given that policy $\pi(x,u)$ is used.
- The solution to the Bellman equation is the value given by the infinite sum (seen earlier).

$$V^{\pi}(x) = E_{\pi}\left\{J_k \mid x_k = x\right\} = E_{\pi}\left\{\sum_{i=k}^{\infty} \gamma^{i-k} r_i \mid x_k = x\right\}$$

Bellman Equation vs Bellman Optimality Equation

Bellman Optimality Equation:

$$V_k{}^*(x) = \min_{\pi} \sum_u \pi(x,u) \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V^*_{k+1}\left(x'\right)\right]$$

The Bellman optimality equation involves the "minimum" operator and so does not contain any specific policy $\pi(x,u)$. Its solution relies on knowing the dynamics, in the form of transition probabilities.
By contrast........

Bellman Equation vs Bellman Optimality Equation

Bellman Equation:

$$V^{\pi}(x) = \sum_{u} \pi(x,u) \sum_{x'} P^{u}_{xx'} \left[R^{u}_{xx'} + \gamma V^{\pi}\left(x'\right) \right]$$

Bellman equation is simpler than that of the optimality equation, and it is easier to solve. The solution to the Bellman equation yields the value function of a specific policy $\pi(x,u)$.

- $V^{\pi}(x)$ may be considered as a predicted performance,
- $\sum_{u} \pi(x,u) \sum_{x'} P^{u}_{xx'} R^{u}_{xx'}$ the observed one-step reward,
- and $V^{\pi}\left(x'\right)$ as a current estimate of future behavior.

Bellman Eq to Bellman Optimality

Given:

- a current policy $\pi(x, u)$,
- MDP is finite and has N states,
- **Bellman equation** is a system of N simultaneous linear equations for the value $V^{\pi}(x)$ of being in each state x.

The optimal value satisfies

$$\begin{aligned} V^*(x) &= \min_{\pi} V^{\pi}(x) \\ &= \min_{\pi} \sum_{u} \pi(x, u) \sum_{x'} P^{u}_{xx'} \left[R^{u}_{xx'} + \gamma V^{\pi}\left(x'\right) \right] \end{aligned}$$

Bellman's optimality principle then yields the **Bellman optimality equation**

$$V^*(x) = \min V^{\pi}(x)$$

Bellman Eq to Bellman Optimality

Bellman optimality equation can be written as

$$V^*(x) = \min_u \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V^* \left(x' \right) \right]$$

This equation is known as the Hamilton-Jacobi-Bellman equation in control systems. If the MDP is finite and has N states, then the Bellman optimality equation is a system of N nonlinear equations for the optimal value $V^*(x)$ of being in each state. The optimal control is given by

$$u^* = \arg\min_u \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V^* \left(x' \right) \right]$$

Relation to Feedback Control of Dynamical systems

- For the Discrete-Time the linear quadratic regulator (LQR), the Bellman equation becomes a Lyapunov equation.
- The Bellman Optimality Equation for Discrete-Time LQR Is an Algebraic Riccati Equation"

Table of Contents

System

MDP is deterministic and satisfies the state transition equation

$$x_{k+1} = Ax_k + Bu_k,$$

with the discrete time index k. The associated infinite-horizon performance index has deterministic stage costs and is

$$J_k = \frac{1}{2}\sum_{i=k}^{\infty} r_i = \frac{1}{2}\sum_{i=k}^{\infty}\left(x_i^T Q x_i + u_i^T R u_i\right)$$

Here: state space $X = R^n$ and action space $U = R^m$ are infinite and continuous.

Select a policy $u_k = \mu(x_k)$ and write the associated value function as

$$V(x_k) = \frac{1}{2}\sum_{i=k}^{\infty} r_i = \frac{1}{2}\sum_{i=k}^{\infty}\left(x_i^T Q x_i + u_i^T R u_i\right)$$

An equivalent difference equation is

$$\begin{aligned} V(x_k) &= \frac{1}{2}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}\sum_{i=k+1}^{\infty}\left(x_i^T Q x_i + u_i^T R u_i\right) \\ &= \frac{1}{2}\left(x_k^T Q x_k + u_k^T R u_k\right) + V(x_{k+1}). \end{aligned}$$

- The solution $V(x_k)$ to this equation that satisfies $V(0) = 0$, is the value given above.
- **This is exactly the Bellman equation for the LQR.**

Assuming that the value is quadratic in the state so that

$$V_k(x_k) = \frac{1}{2} x_k^T P x_k$$

for some kernel matrix P, yields the Bellman equation form

$$2V(x_k) = x_k^T P x_k = x_k^T Q x_k + u_k^T R u_k + x_{k+1}^T P x_{k+1}$$

which, using the state equation, can be written

$$2V(x_k) = x_k^T Q x_k + u_k^T R u_k + (Ax_k + Bu_k)^T P (Ax_k + Bu_k)$$

Assuming a constant, that is, stationary, state feedback policy $u_k = \mu(x_k) = -Kx_k$ for some stabilizing gain K, write

$$\begin{aligned} 2V(x_k) =& x_k^T P x_k \\ =& x_k^T Q x_k + x_k^T K^T R K x_k \\ &+ x_k^T (A - BK)^T P (A - BK) x_k. \end{aligned}$$

Since this equation holds for all state trajectories, we have

$$(A - BK)^T P (A - BK) - P + Q + K^T R K = 0,$$

which is a Lyapunov equation.

Observations

- That is, the Bellman equation for the discrete-time LQR is equivalent to a Lyapunov equation.
- Value recursion equations do not depend on model (A, B) (see last to last slide).
- But Lyapunov equation can only be used if the state dynamics (A, B) are known (see Last slide).
- Reinforcement learning algorithms for learning optimal solutions online can be devised by using temporal difference methods. That is, reinforcement learning allows the Lyapunov equation to be solved online without knowing A or B.

Bellman Optimality Equation for Discrete-Time LQR

The discrete-time LQR Hamiltonian function is

$$\begin{aligned} H\left(x_k, u_k\right) = & x_k^T Q x_k + u_k^T R u_k + \left(A x_k + B u_k\right)^T \\ & \times P\left(A x_k + B u_k\right) - x_k^T P x_k . \end{aligned}$$

A necessary condition for optimality is the stationarity condition $\partial H\left(x_k, u_k\right) / \partial u_k = 0$,
which is equivalent to finding the minimum control using first partial derivative.

ARE

Solving this equation yields the optimal control

$$u_k = -Kx_k = -\left(B^T PB + R\right)^{-1} B^T PAx_k.$$

Inserting this equation in above, yields the discrete-time **algebraic Riccati equation (ARE)**

$$A^T PA - P + Q - A^T PB\left(B^T PB + R\right)^{-1} B^T PA = 0.$$

The ARE is exactly the Bellman optimality equation for the discrete-time LQR.

References I

Lewis, F. L., Vrabie, D., & Vamvoudakis, K. G. (2012). Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine*, *32*(6), 76-105.