

Outline. This lecture discuss Bellman Optimality Equation. Properties of Bellman Optimality Operator and its proofs. Banach's fixed point theorem. Methods to solve Bellman Optimality theorem using Policy evaluation and iterative policy evaluation algorithm.

1 Bellman's Equation for state values:

Given a policy π , the state value is defined as:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t | S_t = s] \\ &= \sum_{a \in \mathcal{A}(s)} \pi(a|s) \left(r(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s, a) v_\pi(s') \right) \end{aligned}$$

For an given policy, MDP(Markov Decision Process) becomes MRP(Markov Reward Process) as follows.

$$\begin{aligned} v_\pi(s) &= \sum_{a \in \mathcal{A}(s)} \pi(a|s) \left(r(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s, a) v_\pi(s') \right) \\ &= \sum_{a \in \mathcal{A}(s)} \pi(a|s) r(s, a) + \gamma \sum_{a \in \mathcal{A}(s)} \pi(a|s) \sum_{s' \in \mathcal{S}} p(s'|s, a) v_\pi(s') \\ &= r_s^\pi + \gamma \sum_{s' \in \mathcal{S}} P_{s',s}^\pi v_\pi(s') \end{aligned}$$

where

$$\begin{aligned} r_s^\pi &= \sum_{a \in \mathcal{A}(s)} \pi(a|s) r(s, a) \\ P_{s',s}^\pi &= \sum_{a \in \mathcal{A}(s)} \pi(a|s) p(s'|s, a) \end{aligned}$$

The Bellman expectation equation can be written concisely using the induced MRP as follows.

$$V^\pi = r^\pi + \gamma P^\pi V^\pi$$

Where,

V^π =State values corresponding to π

r^π =Immediate expected reward values corresponding to π

P^π =Probability matrix corresponding to π

$$\begin{aligned}
V^\pi &= [v_\pi(s_1) \dots v_\pi(s_N)]^T \\
r^\pi &= [r_\pi(s_1) \dots r_\pi(s_N)]^T \\
P^\pi &= \begin{pmatrix} P_{s_1,s_1}^\pi & P_{s_2,s_1}^\pi & \dots & P_{s_N,s_1}^\pi \\ P_{s_1,s_2}^\pi & P_{s_2,s_2}^\pi & \dots & P_{s_N,s_2}^\pi \\ \vdots & \vdots & \ddots & \vdots \\ P_{s_1,s_N}^\pi & P_{s_2,s_N}^\pi & \dots & P_{s_N,s_N}^\pi \end{pmatrix}
\end{aligned}$$

Thus, to find the state values under policy π , we have to solve a system of linear equations. The solution of above system of linear equations is as follows.

$$V^\pi = (I - \gamma P^\pi)^{-1} r^\pi$$

It requires computing the inverse of a matrix. Thus, the time complexity of finding the state values for a given policy is $O(N^3)$.

2 Bellman Operators

Bellman operator is a mapping from one value functions to another value functions.

2.1 Bellman Expectation Operator:

Bellman Expectation Operator $F^\pi : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is defined as:

$$F^\pi(V) = r^\pi + P^\pi V$$

2.2 Bellman Optimality Operator

Bellman Optimality Operator $F : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is defined as:

$$F \left(\begin{pmatrix} v(s_1) \\ \vdots \\ v(s_N) \end{pmatrix} \right) = \begin{pmatrix} \max_{a \in \mathcal{A}(s_1)} \left(r(s_1, a) + \gamma \sum_{i=1}^N p(s_i | s_1, a) v(s_i) \right) \\ \vdots \\ \max_{a \in \mathcal{A}(s_N)} \left(r(s_N, a) + \gamma \sum_{i=1}^N p(s_i | s_N, a) v(s_i) \right) \end{pmatrix}$$

3 Properties of Bellman Operators

Theorem 1. Monotonicity For any $V, U \in \mathbb{R}^N$ and $V \leq U$ component-wise then

$$\begin{aligned}
F^\pi(V) &\leq F^\pi(U) \\
F(V) &\leq F(U)
\end{aligned}$$

Proof. 1. **Proof for Bellman's Expectation Operator:**

$$\begin{aligned} F^\pi(V) &= r^\pi + \gamma P^\pi V \\ F^\pi(U) &= r^\pi + \gamma P^\pi U \end{aligned}$$

Thus,

$$F^\pi(V) - F^\pi(U) = \gamma P^\pi(V - U)$$

But $V - U \leq \mathbf{0}$ and all elements of P^π are non-negative. Thus $P^\pi(V - U) \leq \mathbf{0}$. Hence

$$F^\pi(V) \leq F^\pi(U)$$

2. **Proof for Bellman's Optimality Operator:**

$$\begin{aligned} F(V) - F(U) &= F\left(\begin{pmatrix} v(s_1) \\ \vdots \\ v(s_N) \end{pmatrix}\right) - F\left(\begin{pmatrix} u(s_1) \\ \vdots \\ u(s_N) \end{pmatrix}\right) \\ &= \left(\begin{array}{c} \max_{a \in \mathcal{A}(s_1)} \left(r(s_1, a) + \gamma \sum_{i=1}^N p(s_i|s_1, a)v(s_i) \right) - \max_{a \in \mathcal{A}(s_1)} \left(r(s_1, a) + \gamma \sum_{i=1}^N p(s_i|s_1, a)u(s_i) \right) \\ \vdots \\ \max_{a \in \mathcal{A}(s_N)} \left(r(s_N, a) + \gamma \sum_{i=1}^N p(s_i|s_N, a)v(s_i) \right) - \max_{a \in \mathcal{A}(s_N)} \left(r(s_N, a) + \gamma \sum_{i=1}^N p(s_i|s_N, a)u(s_i) \right) \end{array} \right) \\ &\leq \left(\begin{array}{c} \max_{a \in \mathcal{A}(s_1)} \left(r(s_1, a) + \gamma \sum_{i=1}^N p(s_i|s_1, a)v(s_i) - r(s_1, a) - \gamma \sum_{i=1}^N p(s_i|s_1, a)u(s_i) \right) \\ \vdots \\ \max_{a \in \mathcal{A}(s_N)} \left(r(s_N, a) + \gamma \sum_{i=1}^N p(s_i|s_N, a)v(s_i) - r(s_N, a) - \gamma \sum_{i=1}^N p(s_i|s_N, a)u(s_i) \right) \end{array} \right) \\ &= \left(\begin{array}{c} \max_{a \in \mathcal{A}(s_1)} \gamma \sum_{i=1}^N p(s_i|s_1, a) (v(s_i) - u(s_i)) \\ \vdots \\ \max_{a \in \mathcal{A}(s_N)} \gamma \sum_{i=1}^N p(s_i|s_N, a) (v(s_i) - u(s_i)) \end{array} \right) \end{aligned}$$

But $v(s_i) - u(s_i) \leq 0, i = 1 \dots N$ and $p(s_i|s_j, a) \geq 0, \forall s_i, s_j \in \mathcal{S}, \forall a \in \mathcal{A}(s_j)$. Thus, we get,

$$\sum_{i=1}^N p(s_i|s_j, a) (v(s_i) - u(s_i)) \leq 0, \forall a \in \mathcal{A}(s_j), j = 1 \dots N$$

Thus,

$$F(V) \leq F(U)$$

□

Theorem 2. Offset: Let $\mathbf{1}$ be an N -dimensional vector whose all elements are 1. Then, for any $c \in \mathbb{R}$,

$$\begin{aligned} F^\pi(V + c\mathbf{1}) &= F^\pi(V) + \gamma c\mathbf{1} \\ F(V + c\mathbf{1}) &= F(V) + \gamma c\mathbf{1} \end{aligned}$$

Proof. 1. **Proof for Bellman's Expectation Operator:**

$$\begin{aligned} F^\pi(V + c\mathbf{1}) &= r^\pi + \gamma P^\pi(V + c\mathbf{1}) \\ &= F^\pi(V) + \gamma c \begin{pmatrix} \sum_{i=1}^N P_{s_i, s_1}^\pi \\ \vdots \\ \sum_{i=1}^N P_{s_i, s_N}^\pi \end{pmatrix} \\ &= F^\pi(V) + \gamma c\mathbf{1} \end{aligned}$$

2. **Proof for Bellman's Optimality Operator:**

$$\begin{aligned} F(V + c\mathbf{1}) &= \begin{pmatrix} \max_{a \in \mathcal{A}(s_1)} \left(r(s_1, a) + \gamma \sum_{i=1}^N p(s_i | s_1, a)(v(s_i) + c) \right) \\ \vdots \\ \max_{a \in \mathcal{A}(s_N)} \left(r(s_N, a) + \gamma \sum_{i=1}^N p(s_i | s_N, a)(v(s_i) + c) \right) \end{pmatrix} \\ &= \begin{pmatrix} \max_{a \in \mathcal{A}(s_1)} \left(r(s_1, a) + \gamma c + \gamma \sum_{i=1}^N p(s_i | s_1, a)v(s_i) \right) \\ \vdots \\ \max_{a \in \mathcal{A}(s_N)} \left(r(s_N, a) + \gamma c + \gamma \sum_{i=1}^N p(s_i | s_N, a)v(s_i) \right) \end{pmatrix} \\ &= \begin{pmatrix} \gamma c + \max_{a \in \mathcal{A}(s_1)} \left(r(s_1, a) + \gamma \sum_{i=1}^N p(s_i | s_1, a)v(s_i) \right) \\ \vdots \\ \gamma c + \max_{a \in \mathcal{A}(s_N)} \left(r(s_N, a) + \gamma \sum_{i=1}^N p(s_i | s_N, a)v(s_i) \right) \end{pmatrix} \\ &= \begin{pmatrix} \max_{a \in \mathcal{A}(s_1)} \left(r(s_1, a) + \gamma \sum_{i=1}^N p(s_i | s_1, a)v(s_i) \right) \\ \vdots \\ \max_{a \in \mathcal{A}(s_N)} \left(r(s_N, a) + \gamma \sum_{i=1}^N p(s_i | s_N, a)v(s_i) \right) \end{pmatrix} + \gamma c\mathbf{1} \\ &= F(V) + \gamma c\mathbf{1} \end{aligned}$$

□

Theorem 3. γ Contraction in L_∞ Norm : For any $V, U \in \mathbb{R}^N$,

$$\begin{aligned} \|F^\pi(V) - F^\pi(U)\|_\infty &\leq \gamma \|V - U\|_\infty \\ \|F(V) - F(U)\|_\infty &\leq \gamma \|V - U\|_\infty \end{aligned}$$

Proof. 1. **Proof for Bellman's Expectation Operator:** Given any policy π we have,

$$\|F^\pi(V) - F^\pi(U)\|_\infty = \gamma \|P^\pi(V - U)\|_\infty$$

we know that, $\|v\|_\infty = \max_{i \in \{1, \dots, N\}} |v_i|$. Thus,

$$\begin{aligned} \gamma \|P^\pi(V - U)\|_\infty &= \gamma \max_{j \in \{1, \dots, N\}} \left| \sum_i P_{i,j}^\pi (v(s_i) - u(s_i)) \right| \\ &\leq \gamma \max_{j \in \{1, \dots, N\}} \sum_i P_{i,j}^\pi |v(s_i) - u(s_i)| \quad \text{using triangle inequality} \\ &\leq \gamma \max_{j \in \{1, \dots, N\}} \max_{i \in \{1, \dots, N\}} |v(s_i) - u(s_i)| \sum_i P_{i,j}^\pi = \gamma \max_{i \in \{1, \dots, N\}} |v(s_i) - u(s_i)| \\ &= \gamma \|V - U\|_\infty \end{aligned}$$

Hence,

$$\|F^\pi(V) - F^\pi(U)\| \leq \gamma \|V - U\|_\infty$$

2. **Proof for Bellman's Optimality Operator:** For any $U, V \in \mathbb{R}^N$, we get

$$\begin{aligned} \|F(V) - F(U)\|_\infty &= \max_{j \in \{1, \dots, N\}} \left| \max_{a \in \mathcal{A}(s_j)} \left(r(s_j, a) + \gamma \sum_{i=1}^N P(s_i|s_j, a)v(s_i) \right) \right. \\ &\quad \left. - \max_{a \in \mathcal{A}(s_j)} \left(r(s_j, a) + \gamma \sum_{i=1}^N P(s_i|s_j, a)u(s_i) \right) \right| \\ &\leq \max_{j \in \{1, \dots, N\}} \left| \max_{a \in \mathcal{A}(s_j)} \gamma \sum_{i=1}^N P(s_i|s_j, a) (v(s_i) - u(s_i)) \right| \\ &\leq \max_{j \in \{1, \dots, N\}} \max_{a \in \mathcal{A}(s_j)} \gamma \sum_{i=1}^N P(s_i|s_j, a) |v(s_i) - u(s_i)| \quad \text{using triangle inequality} \\ &\leq \max_{j \in \{1, \dots, N\}} \max_{a \in \mathcal{A}(s_j)} \max_{i \in \{1, \dots, N\}} \gamma |v(s_i) - u(s_i)| \sum_{i=1}^N P(s_i|s_j, a) \\ &= \max_{i \in \{1, \dots, N\}} \gamma |v(s_i) - u(s_i)| \\ &= \gamma \|V - U\|_\infty \end{aligned}$$

□

4 Banach's Fixed Point Theorem

Let T be a contraction mapping from a closed subset \mathcal{X} of a Banach space E into \mathcal{X} . Then there exists a unique $z \in \mathcal{X}$ such that $T(z) = z$. Banach's fixed point theorem ensures the existence of unique fixed point for Bellman operators F^π and F .

- Let V^π is the unique fixed point of F^π for a given policy π .

- V^* is the unique fixed point of F

Also for any $V \in \mathbb{R}^N$ and any stationary policy π

$$\lim_{k \rightarrow \infty} (F^\pi)^k(V) = V^\pi$$

$$\lim_{k \rightarrow \infty} (F)^k(V) = V$$

5 Methods to Solve Bellman's Optimality Equations

We want to find the optimal policy which simultaneously maximizes the state values $v(s), \forall s$ simultaneously. Following are the different methods to solve Bellman's Optimality Equation.

1. **Dynamic Programming:** It requires complete and accurate model of the environment. Which means, we need to know

$$Pr\{R_{t+1} = r, S_{t+1} = s' | S_t = s, A_t = a\} \quad \forall s, s' \in \mathcal{S}, a \in \mathcal{A}(s), r$$

2. **Monte Carlo:** It is conceptually simple. No model is required but its unsuitable for incremental computation.
3. **Temporal difference methods:** No model required. Suitable for incremental computation but mathematically complex to analyse.

6 Dynamic Programming

The term dynamic programming (DP) refers to a collection of algorithms that can be used to compute optimal policies given a perfect model of the environment as a Markov decision process (MDP). We usually assume that the environment is a finite MDP with states S , action $\mathcal{A}(s), \forall s \in \mathcal{S}$ are finite, and that its dynamics are given by a set of probabilities $p(s', r | s, a), \forall s, s' \in \mathcal{S}, a \in \mathcal{A}(s)$.

The key idea of DP, and of reinforcement learning generally, is the use of value functions to organize and structure the search for good policies. We will show how DP can be used to compute the value functions. Dynamic Programming can solved using two methods:

1. **Value Iteration:** It includes finding optimal value function and one policy extraction.
2. **Policy Iteration:** It includes policy evaluation and policy improvement, and the two are repeated iteratively until policy converges.

7 Policy Iteration

Policy iteration starts with a policy. Then it evaluates the state values for each state. Then it finds a new policy which improves the state value for at least one state. Thus, it follows two steps repeatedly.

1. **Policy Evaluation:** Compute $v_\pi(s), \forall s \in \mathcal{S}$ for given π . As we have seen, a state-value can be found as

$$v_\pi(s) = \sum_{a \in \mathcal{A}(s)} \pi(a|s) \sum_{s',r} p(s',r|s,a) [r(s,a,s') + \gamma v_\pi(s')]$$

where $r(s,a,s') = \mathbb{E}_\pi [R_{t+1}|s_t = s, A_t = a, s_{t+1} = s']$. For a given π , one can find the state values by solving the following set of linear equations.

$$V^\pi = r^\pi + P^\pi V^\pi \quad (1)$$

An alternative method of finding the state values for a given π is described in Algorithm 1

2. **Policy Improvement:** Finds a new policy which improves the state values for at least one state as follows. For each $s \in \mathcal{S}$,

$$\pi'(s) = \arg \max_{a \in \mathcal{A}(s)} \left[r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) v_\pi(s') \right]$$

7.1 Iterative Policy Evaluation

Algorithm 1 Policy Evaluation

Input: π , the policy to be evaluated

Initialization: Set $v_0(s) = 0, \forall s \in \mathcal{S}$, set $k \leftarrow 0, \Delta \leftarrow 0$

while $\Delta < \Theta$ **do**

for $s \in S$ **do**

$$v_{k+1}(s) = \sum_{a \in \mathcal{A}(s)} \pi(a|s) \sum_{s',r} p(s',r|s,a) [r(s,a,s') + \gamma v_k(s')] \\ \Delta \leftarrow \max(\Delta, |v_{k+1}(s) - v_k(s)|)$$

end for

$$k \leftarrow k + 1$$

end while

Output: $v_{k+1}(s), s \in \mathcal{S}$

The policy evaluation algorithm works as follows. The initial approximation, V_0 , is chosen as zero vector. In general, it can be initialized arbitrarily. A sequence of approximate values V_0, V_1, V_2, \dots by using the Bellman's expectation operator successively. Thus $V_{k+1} = F^\pi(V_k)$. Which means

$$v_{k+1}(s) = \sum_{a \in \mathcal{A}(s)} \pi(a|s) \sum_{s',r} p(s',r|s,a) [r(s,a,s') + \gamma v_k(s')], \quad \forall s \in \mathcal{S}$$

Let V_π is the fixed point of F^π (F^π is a γ -contraction operator). The sequence V_k can be shown to converge to V_π as $k \rightarrow \infty$.

References

- [1] Sutton, Richard S and Barto, Andrew G *Reinforcement learning: An introduction*, MIT press Cambridge 1998.