

**Outline.** Description of **Value Iteration** algorithm with an example, its correctness and convergence. Comparison with **Policy Iteration** algorithm.

## 1 Effect of $\gamma$ on Optimal policy Calculation

Consider a finite MDP as described in Figure 1:

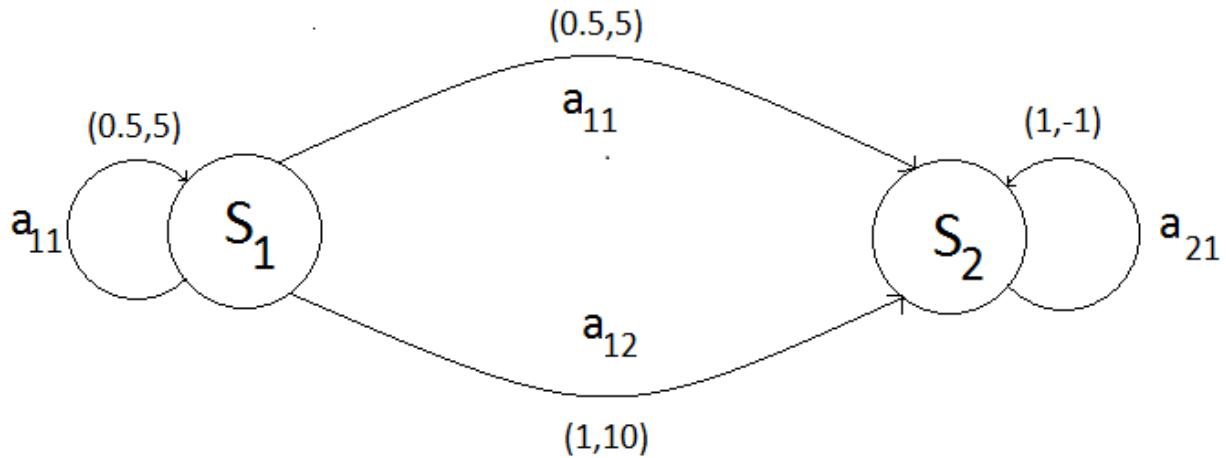


Figure 1: Example of MDP with two states,  $s_1$  and  $s_2$  i.e.  $\mathcal{S} = \{s_1, s_2\}$ . The action set is  $\mathcal{A} = \{a_{11}, a_{12}, a_{21}\}$ . Only transitions with non-zero probabilities are represented. Each transition is labeled with the action taken followed by a pair  $(p(s'|s, a), r(s, a, s'))$ .

$$v_\star(s_1) = \max \left\{ \begin{array}{l} 5 + 0.5\gamma v_\star(s_1) + 0.5\gamma v_\star(s_1) \\ 10 + \gamma v_\star(s_2) \end{array} \right\}$$

$$v_\star(s_2) = -1 + \gamma v_\star(s_2)$$

### 1.1 Case I : $\gamma = 0$

we are looking for immediate reward.

$$v_\star(s_1) = \max(5, 10) = 10$$

$$v_\star(s_2) = -1$$

$$\pi_\star(s_1) = a_{12}$$

$$\pi_\star(s_2) = a_{21}$$

## 1.2 Case II : $\gamma = 0.5$

$$\begin{aligned} v_\star(s_2) &= -1 + 0.5v_\star(s_2) = -2 \\ v_\star(s_1) &= \max \left\{ \begin{array}{l} 5 + 0.5 \times 0.5v_\star(s_1) + 0.5 \times 0.5v_\star(s_2) \\ 10 + 0.5v_\star(s_2) \end{array} \right\} \\ &= \max \{4.5 + 0.25v_\star(s_1), 9\} \end{aligned}$$

Thus,  $v_\star(s_1) \geq 4.5 + 0.25v_\star(s_1)$ , that is  $v_\star(s_1) \geq \frac{4.5}{0.75} = 6$ . And  $v_\star(s_1) \geq 9$ . Thus,  $v_\star(s_1) = \max(6, 9) = 9$ . Hence,

$$\begin{aligned} \pi_\star(s_1) &= a_{12} \\ \pi_\star(s_2) &= a_{21} \end{aligned}$$

## 1.3 Case III : $\gamma = 0.95$

$$\begin{aligned} v_\star(s_2) &= -1 + 0.95 \times v_\star(s_2) = -20 \\ v_\star(s_1) &= \max \left\{ \begin{array}{l} 5 + 0.5 \times 0.95 \times v_\star(s_1) + 0.5 \times 0.95 \times v_\star(s_2) \\ 10 + 0.95 \times v_\star(s_2) \end{array} \right\} \\ &= \max \left\{ \begin{array}{l} 5 + 0.475 \times v_\star(s_1) - 9.5 \\ 10 - 19 \end{array} \right\} \\ &= \max \left\{ \begin{array}{l} 5 + 0.475 \times v_\star(s_1) - 9.5 \\ -9 \end{array} \right\} \\ &= \max \left\{ \begin{array}{l} -8.57 \\ -9 \end{array} \right\} \\ &= -8.57 \end{aligned}$$

Thus,

$$\begin{aligned} \pi_\star(s_1) &= a_{11} \\ \pi_\star(s_2) &= a_{21} \end{aligned}$$

Thus optimum policy changes with different values of  $\gamma$ .

## 2 Value Iteration Algorithm

Value iteration algorithm finds optimal policy by first finding the fixed point of Bellman optimality operator. So, using Value Iteration, our approximate value function converges to the true value function of that policy. More formal description of **Value Iteration** approach is provided in *Algorithm 1*.

$$v_{n+1}(s) = \max_{a \in \mathcal{A}(s)} \left[ r(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s, a) \star v_n(s') \right] \quad \forall s \in \mathcal{S}$$

---

**Algorithm 1** Value Iteration Algorithm

---

- 1: **Input:** MDP and parameters  $\epsilon$  and  $\gamma$
  - 2: **Initialize:** Choose an initial return value function  $v_0(s)$ ,  $\forall s \in S$ ,  $n = 0$
  - 3:  $\forall s \in S$ , assign the next return value function as
- $$V_{n+1}(s) = \max_{a \in A(s)} [r(s, a) + \gamma \sum_{s' \in S} p(s'|s, a)v_n(s')]$$
- 4: **if**  $\| V_{n+1} - V_n \| \leq \epsilon \star \frac{1-\gamma}{2\gamma}$  **then**
  - 5:     Stop
  - 6: **else**
  - 7:      $n = n + 1$
  - 8:     Return to Step (3)
  - 9: **end if**
  - 10: Choose the output policy such that

$$\pi_\epsilon(s) \in \arg \max_{a \in A(s)} [r(s, a) + \gamma \sum_{s' \in S} p(s'|s, a)v_{n+1}(s')]$$


---

Let  $N$  be the number of states, then the update equation can be written in vector form as follows.

$$\begin{aligned} V_{n+1} &= F(V_n) \\ &= F \begin{pmatrix} v_n(s_1) \\ v_n(s_2) \\ \vdots \\ v_n(s_N) \end{pmatrix} \\ &= \begin{pmatrix} \max_{a \in \mathcal{A}(s_1)} \left( r(s_1, a) + \gamma \sum_{i=1}^N p(s_i|s_1, a)v_n(s_i) \right) \\ \max_{a \in \mathcal{A}(s_2)} \left( r(s_2, a) + \gamma \sum_{i=1}^N p(s_i|s_2, a)v_n(s_i) \right) \\ \vdots \\ \max_{a \in \mathcal{A}(s_N)} \left( r(s_N, a) + \gamma \sum_{i=1}^N p(s_i|s_N, a)v_n(s_i) \right) \end{pmatrix} \end{aligned}$$

### 3 Example : Value Iteration

Consider a finite MDP as described in Figure 1.

- Possible Actions at states  $s_1$  and  $s_2$  are given by

$$\begin{aligned} \mathcal{A}(s_1) &= \{a_{11}, a_{12}\} \\ \mathcal{A}(s_2) &= \{a_{21}\} \end{aligned}$$

- Transition probabilities are

$$\begin{aligned} p(s_1|s_1, a_{11}) &= 0.5 \\ p(s_2|s_1, a_{11}) &= 0.5 \\ p(s_1|s_1, a_{12}) &= 1 \\ p(s_1|s_2, a_{21}) &= 1 \end{aligned}$$

- Rewards  $r(s, a, s')$  are given by

$$\begin{aligned} r(s_1, a_{11}, s_1) &= 5 \\ r(s_2, a_{11}, s_2) &= 5 \\ r(s_1, a_{12}, s_2) &= 10 \\ r(s_1, a_{21}, s_2) &= -1 \end{aligned}$$

- Let  $\gamma = 0.95$  and the initial state values are:

$$\begin{aligned} v_0(s_1) &= 0 \\ v_0(s_2) &= 0 \end{aligned}$$

### Iteration 1

*Step 1:*

$$\begin{aligned} v_1(s_1) &= \max \left\{ \begin{array}{l} 5 + 0.5 \times 0.95 \times v_0(s_1) + 0.5 \times 0.95 \times v_0(s_2) \\ 10 + 0.95 \times v_0(s_2) \end{array} \right\} \\ &= \max(5, 10) \\ &= 10 \\ v_1(s_2) &= -1 + 0.95 \times v_0(s_2) \\ &= -1 \end{aligned}$$

Iterating the step 1 until the change in state values is not significant. And system will stop after 162 iterations and using the state values optimal policy is obtained:

$$\begin{aligned} \pi_\epsilon(s_1) &= a_{11} \\ \pi_\epsilon(s_2) &= a_{21} \end{aligned}$$

## 4 Correctness Of Value Iteration

**Theorem 4.1.** *For the series  $V_n$  and policy  $\pi_\epsilon$  computed by value iteration following will hold:*

1. *Let  $T$  be a  $\gamma$ -Contraction Mapping  $T : \mathbb{R}^N \rightarrow \mathbb{R}^N$  and  $V_{n+1} = T(V_n)$ , then,  $\lim_{n \rightarrow \infty} V_n = V_\gamma^*$  where  $V_\gamma^*$  is unique fixed point of  $T$ , which means  $T(V_\gamma^*) = V_\gamma^*$ .*
2. *Given  $\epsilon > 0$ ,  $\exists n_0 \in \mathbb{N}$  such that  $\|V_{n+1} - V_n\|_\infty \leq \frac{\epsilon(1-\gamma)}{2\gamma}$ ,  $\forall n \geq n_0$ . This ensures that the value iteration algorithm indeed reaches the stopping condition.*

3. if  $\| V_{n+1} - V_n \|_\infty \leq \frac{\epsilon(1-\gamma)}{2\gamma}$ , then

$$\| V_{n+1} - V_\gamma^* \|_\infty \leq \frac{\epsilon}{2}$$

i.e. state values obtained after the value iteration algorithm stops, are close to the optimal state values.

4. The policy  $\pi_\epsilon$  is  $\epsilon$ -optimal, which means

$$\| V_\gamma^* - V_\gamma^{\pi_\epsilon} \|_\infty \leq \epsilon$$

i.e. state values corresponding to  $\pi_\epsilon$  are close to the optimal state values.

*Proof.* 1. The proof is discussed in previous lectures.

2. As the sequence  $V_{n+1} = T(V_n)$  is a Cauchy Sequence, thus for a given  $\epsilon_1 > 0$ ,  $\exists n_0 \in \mathbb{N}$  s.t.  $\forall n > n_0$ ,  $\| V_{n+1} - V_n \|_\infty \leq \epsilon_1$ . We can choose  $\epsilon_1 = \frac{\epsilon(1-\gamma)}{2\gamma}$  for a given  $\epsilon > 0$ . Thus,  $\| V_{n+1} - V_n \|_\infty \leq \frac{\epsilon(1-\gamma)}{2\gamma}$ ,  $\forall n \geq n_0$ .

3.

$$\begin{aligned} \| V_\gamma^* - V_{n+1} \|_\infty &= \| V_\gamma^* - FV_{n+1} + FV_{n+1} - V_{n+1} \|_\infty && \text{where } F \text{ is Bellman Optimality Operator} \\ &\leq \| V_\gamma^* - FV_{n+1} \|_\infty + \| FV_{n+1} - V_{n+1} \|_\infty \\ &= \| FV_\gamma^* - FV_{n+1} \|_\infty + \| FV_{n+1} - FV_n \|_\infty && \because V_\gamma^* \text{ is fixed point} \\ &\leq \gamma \| V_\gamma^* - V_{n+1} \|_\infty + \gamma \| V_{n+1} - V_n \|_\infty && \text{using } \gamma - \text{contraction} \end{aligned}$$

Thus,

$$\begin{aligned} \| V_\gamma^* - V_{n+1} \|_\infty &\leq \frac{\gamma}{1-\gamma} \| V_{n+1} - V_n \|_\infty \\ &\leq \frac{\gamma}{1-\gamma} \frac{\epsilon(1-\gamma)}{2\gamma} && \text{using property 2 of the theorem} \\ &= \frac{\epsilon}{2} \end{aligned}$$

4. Assuming the stopping criterion is met, which means  $\| V_{n+1} - V_n \|_\infty \leq \frac{\epsilon(1-\gamma)}{2\gamma}$ . Thus,

$$\begin{aligned} \| V_\gamma^* - V_\gamma^{\pi_\epsilon} \|_\infty &= \| V_\gamma^* - V_{n+1} + V_{n+1} - V_\gamma^{\pi_\epsilon} \|_\infty \\ &\leq \underbrace{\| V_\gamma^* - V_{n+1} \|_\infty}_{\text{Part 1}} + \underbrace{\| V_{n+1} - V_\gamma^{\pi_\epsilon} \|_\infty}_{\text{Part 2}} && \text{using triangle inequality} \\ &\leq \frac{\epsilon}{2} + \| V_{n+1} - V_\gamma^{\pi_\epsilon} \|_\infty && \text{using part 3 of the theorem} \end{aligned}$$

But, we see that

$$\begin{aligned} \| V_{n+1} - V_\gamma^{\pi_\epsilon} \|_\infty &= \| V_\gamma^{\pi_\epsilon} - FV_{n+1} + FV_{n+1} - V_{n+1} \|_\infty \\ &\leq \| V_\gamma^{\pi_\epsilon} - FV_{n+1} \|_\infty + \| FV_{n+1} - V_{n+1} \|_\infty \\ &= \| F^{\pi_\epsilon} V_\gamma^{\pi_\epsilon} - FV_{n+1} \|_\infty + \| FV_{n+1} - FV_n \|_\infty \\ &= \| F^{\pi_\epsilon} V_\gamma^{\pi_\epsilon} - F^{\pi_\epsilon} V_{n+1} \|_\infty + \| FV_{n+1} - FV_n \|_\infty && \because \pi_\epsilon \text{ is found using } V_{n+1} \\ &\leq \gamma \| V_\gamma^{\pi_\epsilon} - V_{n+1} \|_\infty + \gamma \| V_{n+1} - V_n \|_\infty && \text{using } \gamma - \text{contraction} \end{aligned}$$

Thus,

$$\begin{aligned}
\| V_\gamma^{\pi_\epsilon} - V_{n+1} \|_\infty &\leq \gamma \| V_{n+1} - V_n \|_\infty \\
&\leq \frac{\gamma}{1-\gamma} \frac{\epsilon(1-\gamma)}{2\gamma} && \text{using property 2 of the theorem} \\
&= \frac{\epsilon}{2}
\end{aligned}$$

Thus,

$$\| V_\gamma^* - V_\gamma^{\pi_\epsilon} \|_\infty \leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon$$

□

## 5 Convergence Rate Of Value Iteration

We show that the convergence of the value iteration algorithm to the optimal policy is exponentially fast in the parameter  $\lambda$ .

**Theorem 5.1.** *Let  $V_n$  be the sequence of state values found by **Value Iteration** then:*

$$1. \| V_n - V_\gamma^* \| \leq \frac{\gamma^n}{1-\gamma} \| V_1 - V_0 \|$$

$$2. \| V_\gamma^{\pi_\epsilon} - V_\gamma^* \| \leq \frac{2\gamma^n}{1-\gamma} \| V_1 - V_0 \|$$

*Proof.*

$$\begin{aligned}
\| V_n - V_\gamma^* \| &= \| V_\gamma^* - V_n \| \\
&\leq \frac{\gamma}{1-\gamma} \| V_n - V_{n-1} \| && \text{as seen in the previous theorem} \\
&\leq \frac{\gamma^n}{1-\gamma} \| V_1 - V_0 \| && \text{using } \gamma - \text{contraction property}
\end{aligned}$$

Similarly,

$$\begin{aligned}
\| V_\gamma^{\pi_\epsilon} - V_\gamma^* \| &\leq \| V_\gamma^* - V_n \| + \| V_n - V_\gamma^{\pi_\epsilon} \| \\
&\leq \frac{\gamma^n}{1-\gamma} \| V_1 - V_0 \| + \frac{\gamma^n}{1-\gamma} \| V_1 - V_0 \| \\
&\leq \frac{2\gamma^n}{1-\gamma} \| V_1 - V_0 \|
\end{aligned}$$

□

Thus, after each iteration of the value iteration algorithm, the state value vector becomes closer to the optimal state value vector by a factor of  $\lambda$ . The rate of convergence decreases as  $\lambda$  becomes closer to 1.

## 6 Policy Iteration versus Value Iteration

Let  $|S|$  be the cardinality of state space and  $|\mathcal{A}|$  be the cardinality of the action space. Policy iteration algorithm converges in finite number of iterations, at worst  $O(|\mathcal{A}|^{|S|})$ . In each iteration, it goes for two steps, namely, policy evaluation and policy improvement. Policy evaluation step takes  $O(|S|^3)$  time and policy improvement takes  $O(|S| \cdot |\mathcal{A}|)$  time. Thus, the overall time complexity of policy iteration is  $O(|S|^3 \cdot |\mathcal{A}|^{|S|})$ . On the other hand, convergence of value iteration algorithm is asymptotic. Each iteration of value iteration algorithm takes  $O(|S| \cdot |\mathcal{A}|)$  time.

Now, we will show the relationship between the state values generated by value iteration and policy iteration algorithms.

**Theorem 6.1.** *Let  $(U_n)_{n \in \mathbb{N}}$  be the sequence of policy values generated by the value iteration algorithm, and  $(V_n)_{n \in \mathbb{N}}$  the one generated by the policy iteration algorithm. If  $U_0 = V_0$  then,*

$$\forall n \in \mathbb{N}, \quad U_n \leq V_n \leq V^*$$

*Proof.* The proof is by induction. Assume that  $U_n \leq V_n$ , then by the monotonicity of  $F$ , we have

$$\begin{aligned} U_{n+1} &= F(U_n) \\ &\leq F(V_n) \\ &= \left( \begin{array}{l} \max_{a \in \mathcal{A}(s_1)} \left( r(s_1, a) + \gamma \sum_{i=1}^N p(s_i | s_1, a) v_n(s_i) \right) \\ \max_{a \in \mathcal{A}(s_2)} \left( r(s_2, a) + \gamma \sum_{i=1}^N p(s_i | s_2, a) v_n(s_i) \right) \\ \vdots \\ \max_{a \in \mathcal{A}(s_N)} \left( r(s_N, a) + \gamma \sum_{i=1}^N p(s_i | s_N, a) v_n(s_i) \right) \end{array} \right) \end{aligned}$$

Let  $\pi_{n+1}$  be the maximizing policy, that is,

$$\pi_{n+1}(s_i) = \arg \max_{a \in \mathcal{A}(s_i)} \left( r(s_i, a) + \gamma \sum_{j=1}^N p(s_j | s_i, a) v_n(s_j) \right), \quad i = 1 \dots N$$

Then,

$$\begin{aligned} F(V_n) &= F^{\pi_{n+1}}(V_n) \\ &\leq F^{\pi_{n+1}}(V_{n+1}) = V_{n+1} \quad \because V_{n+1} \text{ is the fixed point of } F^{\pi_{n+1}} \end{aligned}$$

$$\therefore U_{n+1} \leq V_{n+1}.$$

□

## References

- [1] Richard S. Sutton and Andrew Barto, *Reinforcement Learning: An Introduction, Second Edition*, The MIT Press, 2012.