2025科研方法与论文写作大作业-PPT

1024040807-顾许磊

南京邮电大学计算机学院、软件学院、网络空间安全学院

Table of contents

Table of contents

Value iteration algorithm

- 如何求解 Bellman optimality equation
  - 回顾一下 BOE

$$v = f(v) = \max_{\pi} \pi(r_{\pi} + \gamma P_{\pi} v)$$

  - 实际上我们可以通过迭代的方式来进行求解

$$v_{k+1} = f(v_k) = \max_{\pi} \pi(r_{\pi} + \gamma P_{\pi} v_k)$$

  - 其中 v_0 可以任意初始化，通过迭代我们最终可以得到最优策略，这个算法就称为 value iteration

Value iteration algorithm

$$v_{k+1} = f(v_k) = \max_{\pi} \pi(r_\pi + \gamma P_\pi v_k)$$

算法可以分两步

- policy update, 对于一个给定的 v_k, 我们希望能够找到一个最优的 π

$$\pi_{k+1} = arg\max_{\pi} \pi(r_\pi + \gamma P_\pi v_k)$$

- value update, 将我们得到的 π 代入, 求解 v_{k+1}

$$v_{k+1} = r_{\pi_{k+1}} + \gamma P_{\pi_{k+1}} v_k$$

Value iteration algorithm

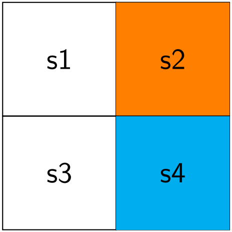
Pseudocode: Value iteration algorithm

Initialization: The probability model $p(r|s,a)$ and $p(s'|s,a)$ for all (s,a) are known. Initial guess v_0.

Aim: Search the optimal state value and an optimal policy solving the Bellman optimality equation. While v_k has not converged in the sense that $\|v_k - v_{k-1}\|$ is greater than a predefined small threshold, for the kth iteration, do

- For every state $s \in \mathcal{S}$, do
  - For every action $a \in \mathcal{A}(s)$, do
    - q-value: $q_k(s,a) = \sum_r p(r|s,a) r + \gamma \sum_{s'} p(s'|s,a) v_k(s')$
  - Maximum action value: $a_k^*(s) = \arg\max_a q_k(a,s)$
  - Policy update: $\pi_{k+1}(a|s) = 1$ if $a = a_k^*$, and $\pi_{k+1}(a|s) = 0$ otherwise
  - Value update: $v_{k+1}(s) = \max_a q_k(a,s)$

环境的奖励设置为 $r_{boundary} = r_{forbidden} = -1, r_{target} = 1$,discount rate $\gamma = 0.9$，我们把所有状态的 v 值都初始化为 0



q-value	$a_1 \uparrow$	$a_2 \rightarrow$	$a_3 \downarrow$	$a_4 \leftarrow$	$a_5 (stay)$
s_1	-1	-1	0	-1	0
s_2	-1	-1	1	0	-1
s_3	0	1	-1	-1	0
s_4	-1	-1	-1	0	1

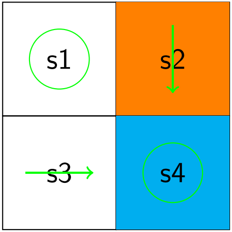
Value iteration algorithm

- Step 1: Policy update:

$$\pi_1(a_5|s_1) = 1, \pi_1(a_3|s_2) = 1, \pi_1(a_2|s_3) = 1, \pi_1(a_5|s_4) = 1$$

- Step 2: Value update:

$$v_1(s_1) = 0, v_1(s_2) = 1, v_1(s_3) = 1, v_1(s_4) = 1$$



q-value	$a_1 \uparrow$	$a_2 \rightarrow$	$a_3 \downarrow$	$a_4 \leftarrow$	$a_5 (stay)$
s_1	-1	-1	0	-1	0
s_2	-1	-1	1	0	-1
s_3	0	1	-1	-1	0
s_4	-1	-1	-1	0	1

Value iteration algorithm

- 根据新的 v 值继续进行计算,$v_1(s_1)=0, v_1(s_2)=1, v_1(s_3)=1, v_1(s_4)=1$

q-table	a_1	a_2	a_3	a_4	a_5
s_1	$-1+\gamma 0$	$-1+\gamma 1$	$0+\gamma 1$	$-1+\gamma 0$	$0+\gamma 0$
s_2	$-1+\gamma 1$	$-1+\gamma 1$	$1+\gamma 1$	$0+\gamma 0$	$-1+\gamma 1$
s_3	$0+\gamma 0$	$1+\gamma 1$	$-1+\gamma 1$	$-1+\gamma 1$	$0+\gamma 1$
s_4	$-1+\gamma 1$	$-1+\gamma 1$	$-1+\gamma 1$	$0+\gamma 1$	$1+\gamma 1$

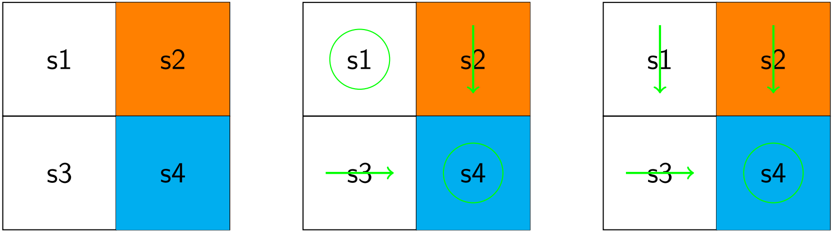
- Step 1: Policy update:

$$\pi_1(a_3|s_1)=1, \pi_1(a_3|s_2)=1, \pi_1(a_2|s_3)=1, \pi_1(a_5|s_4)=1$$

- Step 2: Value update:

$$v_2(s_1)=\gamma 1, v_2(s_2)=1+\gamma 1, v_2(s_3)=1+\gamma 1, v_2(s_4)=1+\gamma 1$$

Value iteration algorithm



Policy iteration algorithm

给定一个随机初始化的策略 π_0，

- Step 1: policy evaluation (PE)
  这一步是为了计算在策略 π_k 下的 state value:
$$v_{\pi_k} = r_{\pi_k} + \gamma P_{\pi_k} v_{\pi_k}$$
- Step 2: policy improvement (PI)
  根据 v_{π_k}，我们可以得到一个新的策略
$$\pi_{k+1} = \arg\max_{\pi} (r_\pi + \gamma P_\pi v_{\pi_k})$$

Policy iteration algorithm

Pseudocode: Policy iteration algorithm

Initialization: The probability model $p(r|s,a)$ and $p(s'|s,a)$ for all (s,a) are known. Initial guess π_0.

Aim: Search for the optimal state value and an optimal policy.

While the policy has not converged, for the kth iteration, do

1. Policy evaluation:
    1. Initialization: an arbitrary initial guess $v_{\pi_k}^{(0)}$
    2. While $v_{\pi_k}^{(j)}$ has not converged, for the jth iteration, do
        1. For every state $s \in \mathcal{S}$, do

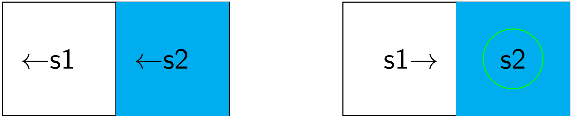
$$v_{\pi_k}^{(j+1)}(s) = \sum_a \pi_k(a|s) \left[\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a) v_{\pi_k}^{(j)}(s') \right]$$

2. Policy improvement:
    1. For every state $s \in \mathcal{S}$, do
        1. For every action $a \in \mathcal{A}(s)$, do

$$q_{\pi_k}(s,a) = \sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a) v_{\pi_k}(s')$$

        2. $a_k^*(s) = \arg\max_a q_{\pi_k}(s,a)$
        3. $\pi_{k+1}(a|s) = 1$ if $a = a_k^*$, and $\pi_{k+1}(a|s) = 0$ otherwise

Policy iteration algorithm



(a) initial policy (b) optimal policy

- The reward setting is $r_{\text{boundary}} = -1$ and $r_{\text{target}} = 1$. The discount rate is $\gamma = 0.9$.
- Actions: a_ℓ, a_0, a_r represent go left, stay unchanged, and go right.
- Aim: use policy iteration to find out the optimal policy.

Iteration $k = 0$: Step 1: policy evaluation

π_0 is selected as the policy in Figure (a). The Bellman equation is

$$v_{\pi_0}(s_1) = -1 + \gamma v_{\pi_0}(s_1), v_{\pi_0}(s_2) = 0 + \gamma v_{\pi_0}(s_1)$$

- Solve the equations directly:

$$v_{\pi_k} = (I - \gamma P_{\pi_k})^{-1} r_{\pi_k}$$

$$v_{\pi_0}(s_1) = -10, \quad v_{\pi_0}(s_2) = -9$$

$$\mathbf{I} - \gamma \mathbf{P}_{\pi_0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - 0.9 \times \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0.1 & 0 \\ -0.9 & 1 \end{bmatrix}$$

$$(\mathbf{I} - \gamma \mathbf{P}_{\pi_0})^{-1} = \begin{bmatrix} 10 & 0 \\ 9 & 1 \end{bmatrix}$$

$$\mathbf{v}_{\pi_0} = \begin{bmatrix} 10 & 0 \\ 9 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} -10 \\ -9 \end{bmatrix}$$

Policy iteration algorithm

- Solve the equations iteratively. Select the initial guess as $v_{\pi_0}^{(0)}(s_1) = v_{\pi_0}^{(0)}(s_2) = 0$:

$$\begin{cases} v_{\pi_0}^{(1)}(s_1) = -1 + \gamma v_{\pi_0}^{(0)}(s_1) = -1, \\ v_{\pi_0}^{(1)}(s_2) = 0 + \gamma v_{\pi_0}^{(0)}(s_1) = 0, \end{cases}$$

$$\begin{cases} v_{\pi_0}^{(2)}(s_1) = -1 + \gamma v_{\pi_0}^{(1)}(s_1) = -1.9, \\ v_{\pi_0}^{(2)}(s_2) = 0 + \gamma v_{\pi_0}^{(1)}(s_1) = -0.9, \end{cases}$$

$$\begin{cases} v_{\pi_0}^{(3)}(s_1) = -1 + \gamma v_{\pi_0}^{(2)}(s_1) = -2.71, \\ v_{\pi_0}^{(3)}(s_2) = 0 + \gamma v_{\pi_0}^{(2)}(s_1) = -1.71, \end{cases}$$

$$\begin{cases} \dots \end{cases}$$

Policy iteration algorithm

Iteration $k = 0$: Step 2: policy improvement

$q_{\pi_k}(s, a)$:

$q_{\pi_k}(s, a)$	a_ℓ	a_0	a_r
s_1	$-1 + \gamma v_{\pi_k}(s_1)$	$0 + \gamma v_{\pi_k}(s_1)$	$1 + \gamma v_{\pi_k}(s_2)$
s_2	$0 + \gamma v_{\pi_k}(s_1)$	$1 + \gamma v_{\pi_k}(s_2)$	$-1 + \gamma v_{\pi_k}(s_2)$

Substituting $v_{\pi_0}(s_1) = -10$, $v_{\pi_0}(s_2) = -9$ and $\gamma = 0.9$ gives

$q_{\pi_0}(s, a)$	a_ℓ	a_0	a_r
s_1	-10	-9	-7.1
s_2	-9	-7.1	-9.1

By seeking the greatest value of q_{π_0}, the improved policy is:

$$\pi_1(a_r|s_1) = 1, \quad \pi_1(a_0|s_2) = 1.$$

Truncated policy iteration algorithm

The two algorithms are very similar:

$$\text{Policy iteration:} \quad \pi_0 \xrightarrow{PE} v_{\pi_0} \xrightarrow{PI} \pi_1 \xrightarrow{PE} v_{\pi_1} \xrightarrow{PI} \pi_2 \xrightarrow{PE} v_{\pi_2} \xrightarrow{PI} \cdots$$

$$\text{Value iteration:} \quad u_0 \xrightarrow{PU} \pi_1' \xrightarrow{VU} u_1 \xrightarrow{PU} \pi_2' \xrightarrow{VU} u_2 \xrightarrow{PU} \cdots$$

PE=policy evaluation. PI=policy improvement.
PU=policy update. VU=value update.

Truncated policy iteration algorithm

▷ Let's compare the steps:

	Policy iteration algorithm	Value iteration algorithm	Comments
1) Policy:	π_0	N/A	
2) Value:	$v_{\pi_0} = r_{\pi_0} + \gamma P_{\pi_0} v_{\pi_0}$	$v_0 := v_{\pi_0}$	
3) Policy:	$\pi_1 = \arg\max_\pi (r_\pi + \gamma P_\pi v_{\pi_0})$	$\pi_1 = \arg\max_\pi (r_\pi + \gamma P_\pi v_0)$	The two policies are the same
4) Value:	$v_{\pi_1} = r_{\pi_1} + \gamma P_{\pi_1} v_{\pi_1}$	$v_1 = r_{\pi_1} + \gamma P_{\pi_1} v_0$	$v_{\pi_1} \geq v_1$ since $v_{\pi_1} \geq v_{\pi_0}$
5) Policy:	$\pi_2 = \arg\max_\pi (r_\pi + \gamma P_\pi v_{\pi_1})$	$\pi_2' = \arg\max_\pi (r_\pi + \gamma P_\pi v_1)$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$

- They start from the same initial condition.
- The first three steps are the same.
- The fourth step becomes different:
  - In policy iteration, solving $v_{\pi_1} = r_{\pi_1} + \gamma P_{\pi_1} v_{\pi_1}$ requires an iterative algorithm (an infinite number of iterations)
  - In value iteration, $v_1 = r_{\pi_1} + \gamma P_{\pi_1} v_0$ is a one-step iteration

Truncated policy iteration algorithm

Consider the step of solving $v_{\pi_1} = r_{\pi_1} + \gamma P_{\pi_1} v_{\pi_1}$:

$$
\begin{aligned}
v_{\pi_1}^{(0)} &= v_0 \\
\text{value iteration} \leftarrow v_1 \leftarrow v_{\pi_1}^{(1)} &= r_{\pi_1} + \gamma P_{\pi_1} v_{\pi_1}^{(0)} \\
v_{\pi_1}^{(2)} &= r_{\pi_1} + \gamma P_{\pi_1} v_{\pi_1}^{(1)} \\
&\vdots \\
\text{truncated policy iteration} \leftarrow \bar{v}_1 \leftarrow v_{\pi_1}^{(j)} &= r_{\pi_1} + \gamma P_{\pi_1} v_{\pi_1}^{(j-1)} \\
&\vdots \\
\text{policy iteration} \leftarrow v_{\pi_1} \leftarrow v_{\pi_1}^{(\infty)} &= r_{\pi_1} + \gamma P_{\pi_1} v_{\pi_1}^{(\infty)}
\end{aligned}
$$