Value Function Evaluation and Policy Update Example

Xiuwen Liu, Department of Computer Science, Florida State University

Here we use the following example from the book, where there are 14 nonterminal states and one terminal state. The value for the terminal state is fixed at 0. For simplicity, we will state 0 to represent the terminal state.

	1	2	3
4	5	6	7
8	9	10	11
12	13	14	

$R_t = -1$
on all transitions

As shown here, all the transitions have an immediate reward of -1. Here we use the iterative evaluation algorithm where the initial values are 0 for the nonterminal states. The policy is the equiprobable random policy; in other words, the probability of taking each of the four actions is 0.25. The reward in the example is undiscounted; in other words, $\gamma = 1$.

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
  $\Delta \leftarrow 0$
  Loop for each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\left[r + \gamma V(s')\right]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

The pseudocode gives an in-place version of iterative policy evaluation. In other words, the updated values are used immediately in new updates. As a result, the intermediate results depend on the order of updates.

For simplicity, we update all the state values using the values from the previous iteration. In other words, the updated values are not available until all the values are updated, known as synchronous evaluation. The equation used is given by:

$$\begin{aligned} v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\Big[r + \gamma v_k(s')\Big], \end{aligned}$$

According to the algorithm, at each iteration, we need to update the value for each state. All the values are initialized to be zero. For the next iteration (k=1), for state 1, we have

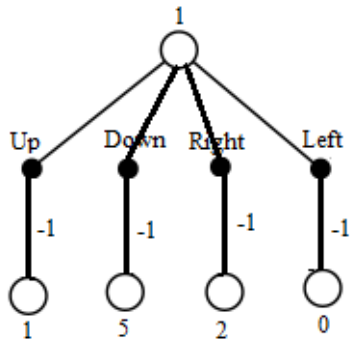
$V_1(1) = 0.25 \times (1 \times [-1 + \gamma \times V_0(1)]) + 0.25 \times (1 \times [-1 + \gamma \times V_0(5)]) + 0.25 \times (1 \times [-1 + \gamma \times V_0(2)]) + 0.25 \times (1 \times [-1 + \gamma \times V_0(0)]) = -1 + \gamma \frac{V_0(1)+V_0(5)+V_0(2)+V_0(0)}{4} = -1.$

Similarly, for all the states, the updated values are -1.

For the next iteration (k=2), again for state 1, we have

$V_2(1) = 0.25 \times (1 \times [-1 + \gamma \times V_1(1)]) + 0.25 \times (1 \times [-1 + \gamma \times V_1(5)]) + 0.25 \times (1 \times [-1 + \gamma \times V_1(2)]) + 0.25 \times (1 \times [-1 + \gamma \times V_1(0)]) = -1 + \gamma \frac{V_1(1)+V_1(5)+V_1(2)+V_1(0)}{4} = -1.75 \approx -1.7.$

The corresponding backup diagram is given as



Similarly, for state 2, we have

$V_2(2) = 0.25 \times (1 \times [-1 + \gamma \times V_1(2)]) + 0.25 \times (1 \times [-1 + \gamma \times V_1(6)]) + 0.25 \times (1 \times [-1 + \gamma \times V_1(3)]) + 0.25 \times (1 \times [-1 + \gamma \times V_1(1)]) = -1 + \gamma \frac{V_1(2)+V_1(6)+V_1(3)+V_1(1)}{4} = -2.$

For the next iteration (k=3), again for state 1, we have

$V_3(1) = 0.25 \times (1 \times [-1 + \gamma \times V_2(1)]) + 0.25 \times (1 \times [-1 + \gamma \times V_2(5)]) + 0.25 \times (1 \times [-1 + \gamma \times V_2(2)]) + 0.25 \times (1 \times [-1 + \gamma \times V_2(0)]) = -1 + \gamma \frac{V_2(1)+V_2(5)+V_2(2)+V_2(0)}{4} = -1 + 1.0 \times \frac{-1.7-2.0-2.0-0}{4} = -2.425 \approx -2.4.$

For this problem, if we continue to update the state values, the values will converge. Again for state 1, we have

$V_\infty(1) = 0.25 \times (1 \times [-1 + \gamma \times V_\infty(1)]) + 0.25 \times (1 \times [-1 + \gamma \times V_\infty(5)]) + 0.25 \times (1 \times [-1 + \gamma \times V_\infty(2)]) + 0.25 \times (1 \times [-1 + \gamma \times V_\infty(0)]) = -1 + \gamma \frac{V_\infty(1)+V_\infty(5)+V_\infty(2)+V_\infty(0)}{4} = -1 + 1.0 \times \frac{-14-20-18-0}{4} = -14.$ In other words, the value will remain the same. You can also verify all the values remain the same and therefore the algorithm has converged. For this problem, you can also find the solution by solving 14 linear equations with 14 unknowns. For example, the equation for state 1 is:

$V(1) = 0.25 \times (1 \times [-1 + \gamma \times V(1)]) + 0.25 \times (1 \times [-1 + \gamma \times V(5)]) + 0.25 \times (1 \times [-1 + \gamma \times V(2)]) + 0.25 \times (1 \times [-1 + \gamma \times V(0)]) = -1 + \frac{V(1)+V(5)+V(2)}{4},$

which is equivalent to

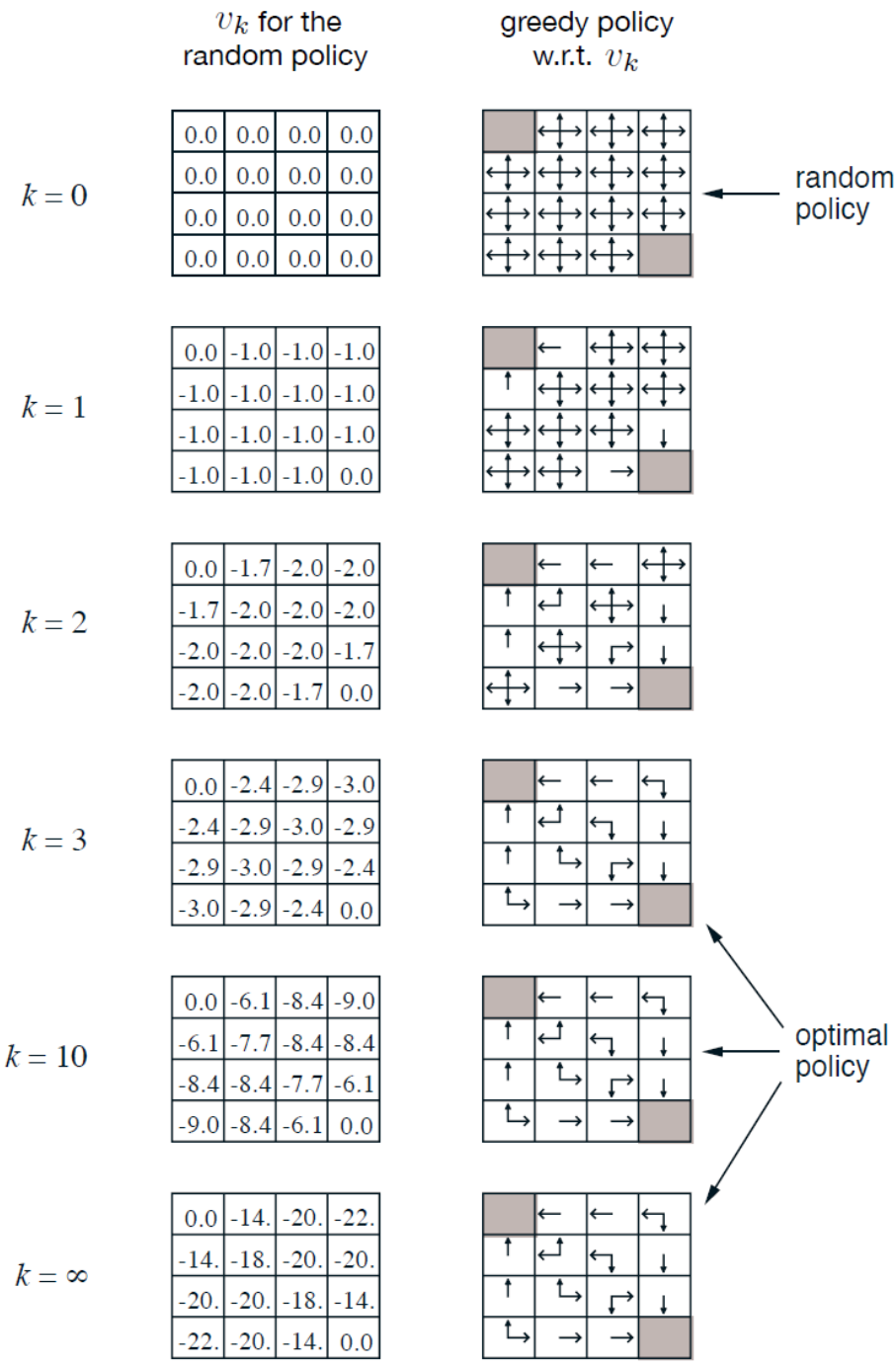
$3V(1) - V(2) - V(5) = -4.$

Similarly you can write down the equations for all the other states. You can also verify the solution given in the book at k=∞ satisfy the equations.

Please make sure that you know how to update the value for any state.

Given a stat-value function, to improve the policy, we will take the action that we will give us the maximal reward. For example, if we use the state value function for k=2, for state 1, the resulting states and the rewards are 1, -2.7 (up), 5, -3.0 (down), 2, -3.0 (right), and 0, -1 (left) respectively. Therefore, we can improve the current policy by going left always if we are in state 1. The equation for policy improvement is

$$\begin{aligned}\pi'(s) &= \arg\max_a Q^\pi(s,a) \\ &= \arg\max_a E\left\{r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s, a_t = a\right\} \\ &= \arg\max_a \sum_{s'} \mathcal{P}^a_{ss'}\left[\mathcal{R}^a_{ss'} + \gamma V^\pi(s')\right],\end{aligned}$$


v_k for the random policy
greedy policy w.r.t. v_k
$k = 0$
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
random policy
$k = 1$
0.0 -1.0 -1.0 -1.0
-1.0 -1.0 -1.0 -1.0
-1.0 -1.0 -1.0 -1.0
-1.0 -1.0 -1.0 0.0
$k = 2$
0.0 -1.7 -2.0 -2.0
-1.7 -2.0 -2.0 -2.0
-2.0 -2.0 -2.0 -1.7
-2.0 -2.0 -1.7 0.0
$k = 3$
0.0 -2.4 -2.9 -3.0
-2.4 -2.9 -3.0 -2.9
-2.9 -3.0 -2.9 -2.4
-3.0 -2.9 -2.4 0.0
$k = 10$
0.0 -6.1 -8.4 -9.0
-6.1 -7.7 -8.4 -8.4
-8.4 -8.4 -7.7 -6.1
-9.0 -8.4 -6.1 0.0
optimal policy
$k = \infty$
0.0 -14. -20. -22.
-14. -18. -20. -20.
-20. -20. -18. -14.
-22. -20. -14. 0.0