# 1 Algorithms for finding the optimal policy

## 1.1 Optimal policies and the utilities of states

Suppose that we enter state $s$ and follow the optimal policy from $s$ onwards, what is our total expected utility? This is a way to measure the "true utility" of each state $s$, denoted by $U(s)$.

Note that $U(s)$ is very different from $R(s)$. $R(s)$ is the "one-time" "short-term" reward of reaching state $s$. $U(s)$ is the "long term" total reward from $s$ onwards.

The following figure shows $U(s)$ for every state (with discount factor $\gamma = 1$ and $R(s) = -0.04$). The utilities are higher for states closer to the +1 exit because fewer steps are required to reach the exit.

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0.705 | 0.655 | 0.611 | 0.388 |
| 2 | 0.762 | X | 0.660 | -1 |
| 3 | 0.812 | 0.868 | 0.918 | +1 |

Given $U(s)$, it's straightforward to determine the optimal action in each state.

What is my expected utility if I'm in state $s$ and take action $a$?

$$U(s,a) = \sum_{s'} P(s'|s,a)U(s')$$

Given this, I would like to choose the action that maximizes my expected utility.

$$\pi^*(s) = \arg\max_a U(s,a).$$

What is the optimal policy in $s_{13}$?

- $U(s_{13}, down) = 0.8 * 0.660 + 0.1 * 0.655 + 0.1 * 0.388 = 0.6323$
- $U(s_{13}, left) = 0.8 * 0.655 + 0.1 * 0.660 + 0.1 * 0.611 = 0.6511$
- $U(s_{13}, right) = 0.8 * 0.388 + 0.1 * 0.611 + 0.1 * 0.660 = 0.4375$
- $U(s_{13}, up) = 0.8 * 0.611 + 0.1 * 0.655 + 0.1 * 0.388 = 0.5931$

Therefore, $\pi^*(s_{13}) = left$.

How do we determine $U(s)$?

## 1.2 Value iteration

Basic idea:

- Calculate the true utility $U(s)$ of each state $s$.
- Choose the optimal action in each state based on $U(s)$.

$U(s)$ are the unique solutions of the Bellman equations.

$$U(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s,a)U(s')$$

$R(s)$ is the immediate reward for reaching state $s$. $\gamma \max_a \sum_{s'} P(s'|s,a)U(s')$ is the discounted expected utility of the next state, assuming that the agent chooses the optimal action.

The Bellman equation for $U(s_{11})$:

$$\begin{aligned} U(s_{11}) = -0.04 + \gamma \max[&0.8U(s_{12}) + 0.1U(s_{21}) + 0.1U(s_{11}), \\ &0.9U(s_{11}) + 0.1U(s_{12}), \\ &0.9U(s_{11}) + 0.1U(s_{21}), \\ &0.8U(s_{21}) + 0.1U(s_{12}) + 0.1U(s_{11})]. \end{aligned}$$

For our example, there are 9 Bellman equations, one for each state. We can solve these 9 equations to find the 9 unknowns $U(s)$.

Can we solve this system of equations efficiently?

- No. These equations are non-linear since "max" is non-linear.
- We can solve linear equations efficiently using linear algebra techniques.

Instead, we will solve for $U(s)$ using an iterative approach.

1. Start with arbitrary initial values for the utilities.
2. Let $U_i(s)$ be the utility value for state $s$ at the $i$th iteration.

   At every iteration, update all $U(s)$ simultaneously using the following update rule.

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) U_i(s')$$

3. Terminate when the maximum change from $U_i(s)$ to $U_{i+1}(s)$ for all $s$ is small enough.

If we apply the Bellman update infinitely often, we are guaranteed to converge to the optimal $U(s)$.

A few notes about the update rule in step 2:

- We apply the update rule from right to left. Plug in the old utility values on the right. The result from the right becomes the new utility value.
- Updating all values "simultaneously" means that, we will use the values of the utilities from the previous iteration to calculate the values of the utilities for the current iteration, and then we will replace all the utility values all at once.

$U_0(s)$:

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | X | 0 | -1 |
| 3 | 0 | 0 | 0 | +1 |

$$U_1(s_{33}) = -0.04 + 1 * max[0.8 * 1, 0.1 * 1, 0, 0.1 * 1] = -0.04 + 0.8 = 0.76$$
$$U_1(s_{23}) = -0.04 + 1 * max[0.1 * (-1), 0.8 * (-1), 0.1 * (-1), 0] = -0.04 + 0 = -0.04$$

$U_1(s)$:

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | $-0.04$ | $-0.04$ | $-0.04$ | $-0.04$ |
| 2 | $-0.04$ | X | $-0.04$ | -1 |
| 3 | $-0.04$ | $-0.04$ | 0.76 | +1 |

$$U_2(s_{33}) = -0.04 + 0.8 * 1 + 0.1 * (-0.04) + 0.1 * 0.76 = 0.832$$
$$U_2(s_{23}) = -0.04 + 0.8 * 0.76 + 0.1(-0.04) + 0.1(-1) = 0.464$$
$$U_2(s_{32}) = -0.04 + 0.8 * 0.76 + 0.2(-0.04) = 0.56$$

$U_2(s)$:

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | $-0.08$ | $-0.08$ | $-0.08$ | $-0.08$ |
| 2 | $-0.08$ | X | 0.464 | -1 |
| 3 | $-0.08$ | 0.56 | 0.832 | +1 |

$$U_3(s_{33}) = -0.04 + 0.8 * 1 + 0.1 * 0.464 + 0.1 * 0.832 = 0.890$$
$$U_3(s_{23}) = -0.04 + 0.8 * 0.832 + 0.1(0.464) + 0.1(-1) = 0.572$$
$$U_3(s_{32}) = -0.04 + 0.8 * 0.832 + 0.1(0.56) + 0.1(0.56) = 0.738$$
$$U_3(s_{31}) = -0.04 + 0.8 * 0.56 + 0.1(-0.08) + 0.1(-0.08) = 0.392$$
$$U_3(s_{13}) = -0.04 + 0.8 * 0.464 + 0.1(-0.08) + 0.1(-0.08) = 0.315$$

$U_3(s)$:

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | −0.12 | −0.12 | 0.315 | −0.12 |
| 2 | −0.12 | X | 0.572 | -1 |
| 3 | 0.392 | 0.738 | 0.890 | +1 |

$U_4(s)$:

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | −0.16 | 0.188 | 0.394 | 0.100 |
| 2 | 0.250 | X | 0.628 | -1 |
| 3 | 0.577 | 0.819 | 0.906 | +1 |

$U_5(s)$:

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0.162 | 0.312 | 0.492 | 0.185 |
| 2 | 0.471 | X | 0.648 | -1 |
| 3 | 0.698 | 0.849 | 0.914 | +1 |

The states accumulate negative rewards until a path is found to $s_{34}$.

After