Model-Based Solutions for MDPs[1]

[1] Sections 2.4-2.6: Yang&Ying

Value iteration

- Recall the Bellman equation for infinite horizon discounted RL

$$V^*(x) = \max_a E\left[r(x,a) + \alpha V^*(x')\right].$$

Value iteration

- Recall the Bellman equation for infinite horizon discounted RL

$$V^*(x) = \max_a E\left[r(x,a) + \alpha V^*(x')\right].$$

- Suppose that both the state space $\mathcal{S}$ and action space $\mathcal{A}$ are finite, $P_{x,x'}$ and $\bar{r}(x,a) = E[r(x,a)]$ are known (model-based).

Value iteration

- Recall the Bellman equation for infinite horizon discounted RL

$$V^*(x) = \max_a E\left[r(x,a) + \alpha V^*(x')\right].$$

- Suppose that both the state space $\mathcal{S}$ and action space $\mathcal{A}$ are finite, $P_{x,x'}$ and $\bar{r}(x,a) = E[r(x,a)]$ are known (model-based).
- Starting from V_0, we can iteratively compute

$$V_{k+1} \leftarrow \max_a \left(\bar{r}(x,a) + \alpha \sum_{x'} P_{x,x'}(a) V_k(x')\right).$$

Value iteration

Starting from V_0, we can iteratively compute

$$V_{k+1} \leftarrow \max_a \left(\bar{r}(x,a) + \alpha \sum_{x'} P_{x,x'}(a) V_k(x') \right).$$

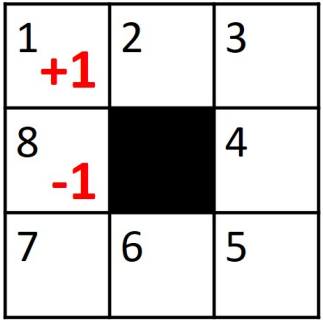
$$V_k \to V^*$$

After finding V^*, we can find the optimal policy by solving

$$\arg\max_a \left(\bar{r}(x,a) + \alpha \sum_{x'} P_{x,x'}(a) V^*(x') \right).$$

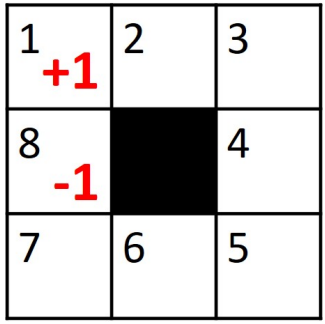
Value iteration: Grid world example

- Two actions: clockwise (c) or counter-clockwise (cc).



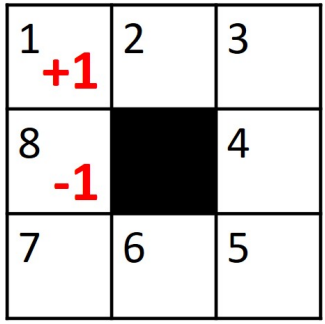
Value iteration: Grid world example

- Two actions: clockwise (c) or counter-clockwise (cc).
- The agent follows the action with probability 0.8 and moves to the opposite direction with probability 0.2.



Value iteration: Grid world example

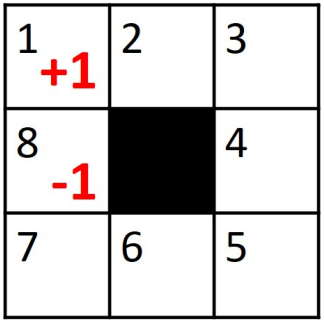
- Two states with nonzero rewards: $+1$ in state 1, -1 in state 8. So for example, $r(1, *) = 1$ and $r(8, *) = -1$.



Value iteration: Grid world example

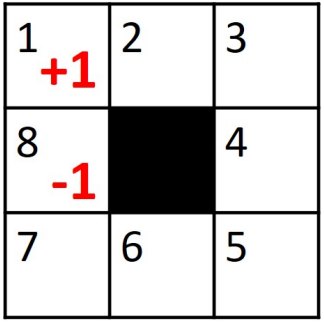
Define

$$\mathbf{V}_k = \begin{pmatrix} V_k(1) \\ V_k(2) \\ V_k(3) \\ V_k(4) \\ V_k(5) \\ V_k(6) \\ V_k(7) \\ V_k(8) \end{pmatrix} \text{ and assume } V_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$



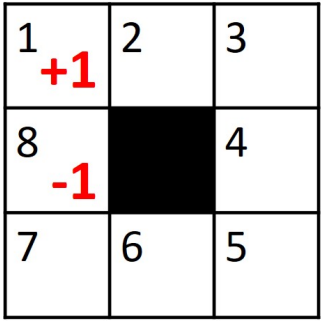
Value iteration: Grid world example

$$
\begin{aligned}
V_1(2) &= \max_{a \in \{c, cc\}} E\left[r(2,a) + 0.9 V_0(x')\right] \\
&= \max_{a \in \{c, cc\}} E\left[0.9 V_0(x')\right] \\
&= \max\{0, 0\} = 0.
\end{aligned}
$$



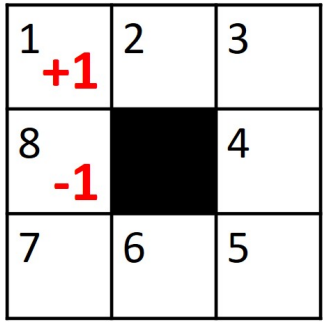
Value iteration: Grid world example

$$
\begin{aligned}
V_1(1) &= \max_{a \in \{c, cc\}} E\left[r(1,a) + 0.9 V_0(x')\right] \\
&= \max_{a \in \{c, cc\}} E\left[1 + 0.9 V_0(x')\right] \\
&= 1.
\end{aligned}
$$



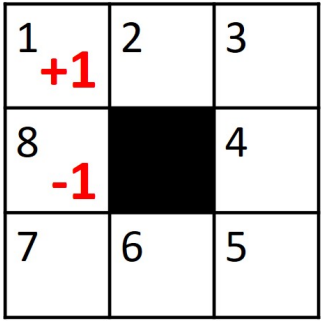
Value iteration: Grid world example

$$V_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \rightarrow V_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$



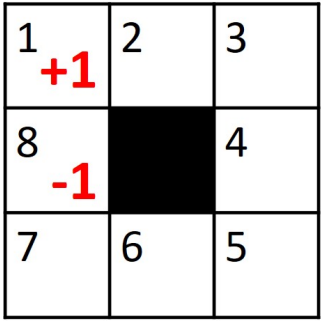
Value iteration: Grid world example

$$
\begin{aligned}
V_2(2) &= \max_{a \in \{c, cc\}} E\left[r(2,a) + 0.9 V_1(x')\right] \\
&= \max_{a \in \{c, cc\}} E\left[0.9 V_1(x')\right] \\
&= \max\{0.9 \times 0.2 \times 1, 0.9 \times 0.8 \times 1\} \\
&= 0.72.
\end{aligned}
$$



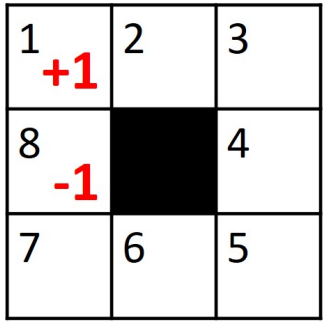
Value iteration: Grid world example

$$
\begin{aligned}
V_2(1) &= \max_{a \in \{c, cc\}} E\left[r(1,a) + 0.9 V_1(x')\right] \\
&= \max_{a \in \{c, cc\}} E\left[1 + 0.9 V_1(x')\right] \\
&= \max\left\{1 + 0.9 \times (-0.2), 1 + 0.9 \times (-0.8)\right\} \\
&= 0.82.
\end{aligned}
$$



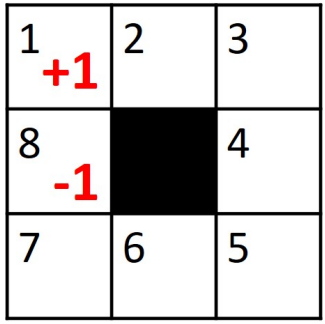
Value iteration: Grid world example

$$V_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \rightarrow V_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{pmatrix} \rightarrow V_2 = \begin{pmatrix} 0.82 \\ 0.72 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.18 \\ -0.28 \end{pmatrix}$$



Value iteration: Grid world example

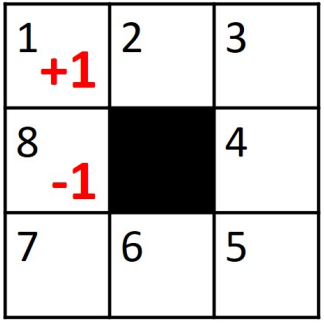
$$V^* = \begin{pmatrix} 3.36 \\ 2.86 \\ 2.43 \\ 2.07 \\ 1.77 \\ 1.54 \\ 1.49 \\ 1.69 \end{pmatrix}$$



Value iteration: Grid world example

$$\begin{aligned}\mu^*(x) &\in \arg \max_{a \in \{c, cc\}} E\left[r(x,a) + 0.9 V^*(x')\right] \\ &= \arg \max_{a \in \{c, cc\}} E\left[0.9 V^*(x')\right] \\ &= \arg\max \left\{V^*(x+c), V^*(x+cc)\right\}.\end{aligned}$$

$$\mu^* = \begin{pmatrix} c \\ cc \\ cc \\ cc \\ cc \\ cc \\ c \\ c \end{pmatrix}$$



Policy iteration

- Choose an initial policy μ, set $k = 0$
- Compute the value function of the policy V_{μ_k} by solving the Bellman equation.

$$V_{\mu_k}(i) = E[r(i, \mu_k(i))] + \alpha \sum_j P_{ij}(\mu_k(i)) V_{\mu_k}(j)$$

- Compute a new policy μ_{k+1} as

$$\mu_{k+1}(i) \leftarrow \arg\max_u E[r(i, u)] + \alpha \sum_j P_{ij}(u) V_{\mu_k}(j)$$

- Repeat if $V_{\mu_{k+1}} \neq V_{\mu_k}$

Policy iteration

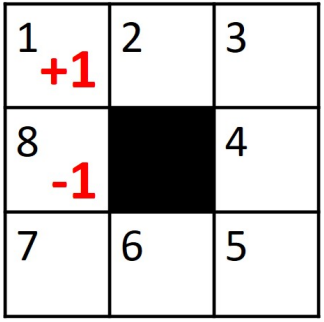
$V_{\mu_{k+1}} \geq V_{\mu_k}$, i.e. the value function improves at each step before the algorithm terminates.

Policy iteration: Grid world example

$\mu_0(i) = c$, i.e. the initial policy is to move clockwise in any state.

$$V_{\mu_0}(i) = E[r(i,c)] + 0.9\,(0.8\,V_{\mu_0}(i+1) + 0.2\,V_{\mu_0}(i-1))$$

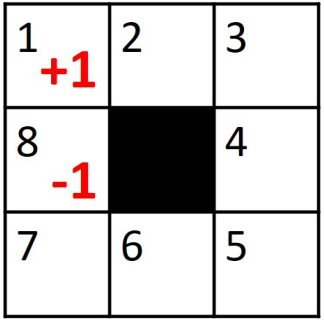
$$V_{\mu_0} = \begin{pmatrix} 1.04 \\ 0.13 \\ -0.08 \\ -0.14 \\ -0.18 \\ -0.21 \\ -0.25 \\ -0.30 \end{pmatrix}$$



Policy iteration: Grid world example

The improved policy is

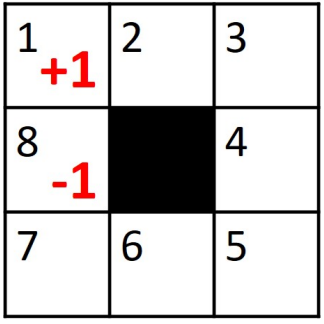
$$\mu_1 = \begin{pmatrix} c \\ cc \\ cc \\ cc \\ cc \\ cc \\ cc \\ c \end{pmatrix}$$



Policy iteration: Grid world example

The improved policy is

$$\mu_1 = \begin{pmatrix} c \\ cc \\ cc \\ cc \\ cc \\ cc \\ cc \\ c \end{pmatrix} \quad \text{recall that } \mu^* = \begin{pmatrix} c \\ cc \\ cc \\ cc \\ cc \\ cc \\ c \\ c \end{pmatrix}$$



Challenges of Applying These Algorithms

Challenges

Model-based algorithms: need to know $\bar{r}(i, u)$ and $P_{ij}(u)$

Challenges of Applying These Algorithms

Challenges

Model-based algorithms: need to know $\bar{r}(i, u)$ and $P_{ij}(u)$

Scalability: only work for finite state space and finite action space

Learning is needed when the system model is unknown.