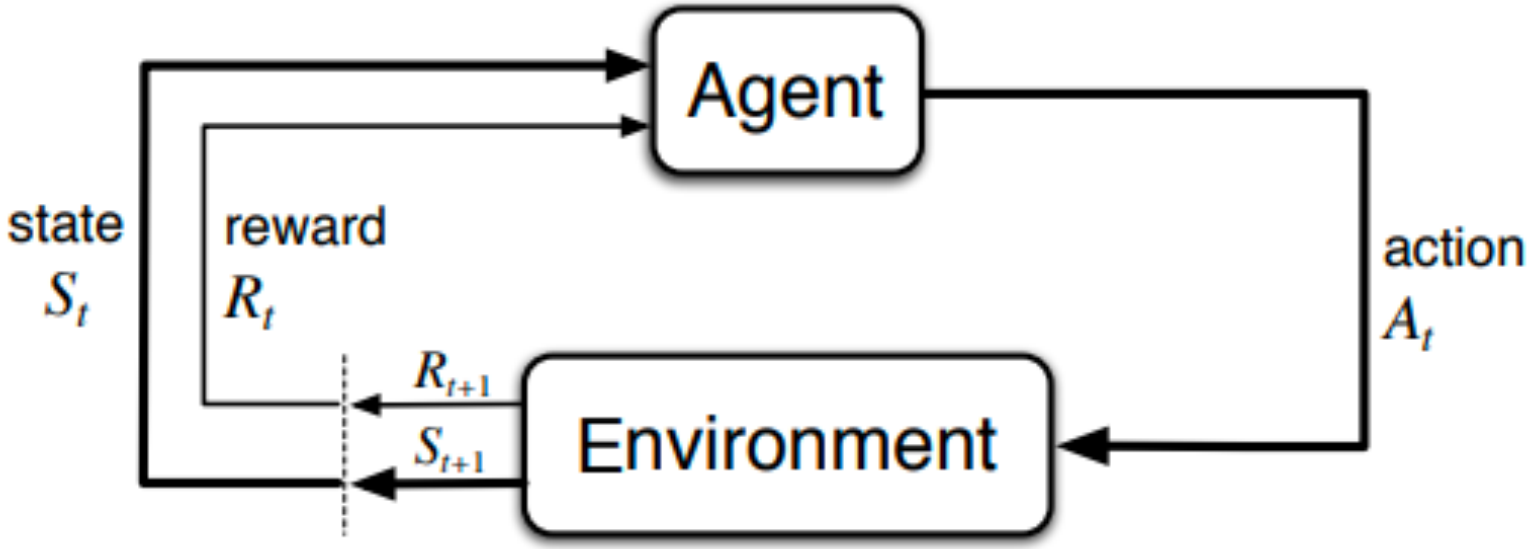
Reinforcement Learning

INSTRUCTOR: HONGJIE CHEN

JUNE 28TH 2022

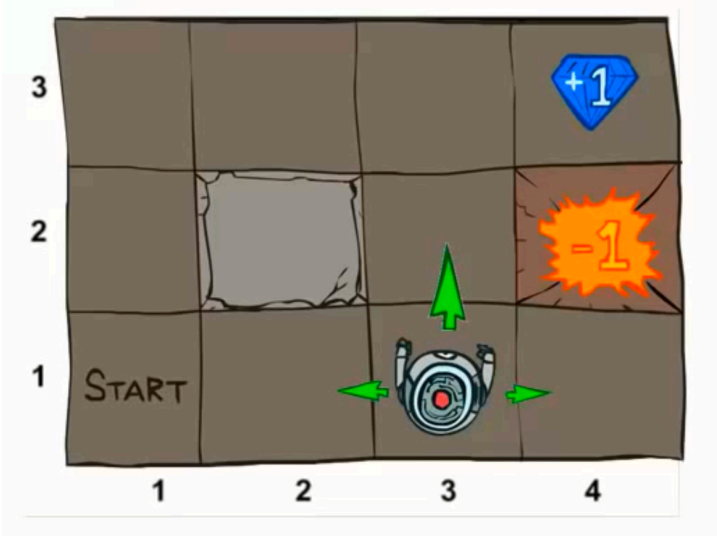
Markov Decision Process (MDP)



MDP Example: GridWorld

- An MDP is defined by
  - Set of states S
  - Set of actions A
  - Transition function $P(s' \mid s, a)$
  - Reward function $R(s, a, s')$
  - Beginning state s_0
  - Discount factor γ
  - Time horizon H

Figure: credit

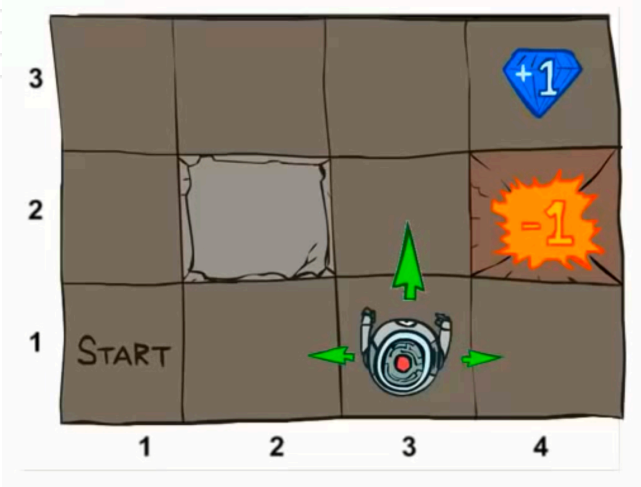


Objective:

$$\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t R(S_t, A_t, S_{t+1}) \mid \pi\right]$$

Optimal Value Function

$$V^*(s) = \max_{\pi} \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t R(s_t, a_t, s_{t+1}) \,|\, \pi, s_0 = s\right]$$



- Sum of discounted rewards when starting at state s
- Assume $\gamma = 1, H = 4$
- Let's calcualte

$V^*(4,3) =$

$V^*(3,3) =$

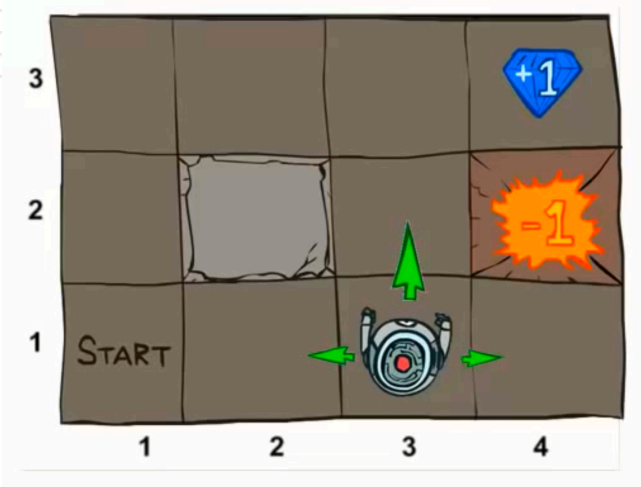
$V^*(2,3) =$

$V^*(1,1) =$

$V^*(4,2) =$

Optimal Value Function

$$V^*(s) = \max_{\pi} \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t R(s_t, a_t, s_{t+1}) \,|\, \pi, s_0 = s\right]$$



- Sum of discounted rewards when starting at state s
- Assume $\gamma = 0.9, H = 4$

$V^*(4,3) =$

$V^*(3,3) =$

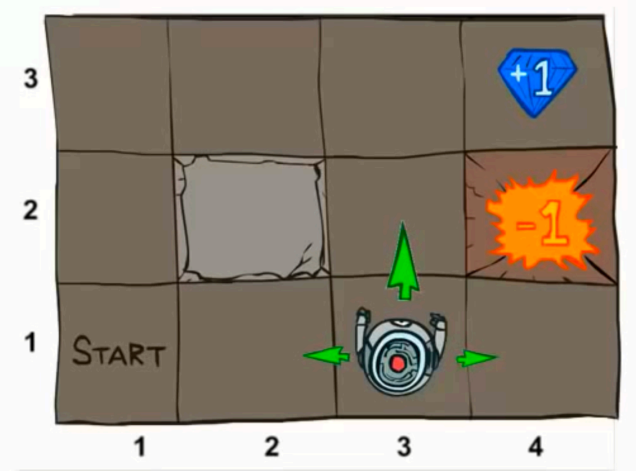
$V^*(2,3) =$

$V^*(1,1) =$

$V^*(4,2) =$

Optimal Value Function

- $$V^*(s) = \max_{\pi} \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t R(s_t, a_t, s_{t+1}) \,|\, \pi, s_0 = s\right]$$



- Sum of discounted rewards when starting at state s
- Assume action can fail, successful probability 0.8, $\gamma = 0.9$, $H = 4$

$V^*(4,3) =$

$V^*(3,3) =$

$V^*(2,3) =$

$V^*(1,1) =$

$V^*(4,2) =$

Value Iteration

- $V_0^*(s) =$
- $V_1^*(s) =$
- $V_2^*(s) =$

- $V_k^*(s) =$

Representing Value Iteration

- $V_0^*(s) = 0, \forall s$
- $V_1^*(s) = \max_a \sum_{s'} P(s' \mid s, a)(R(s, a, s') + \gamma V_0^*(s'))$
- $V_2^*(s) = \max_a \sum_{s'} P(s' \mid s, a)(R(s, a, s') + \gamma V_1^*(s'))$

- $V_k^*(s) = \max_a \sum_{s'} P(s' \mid s, a)(R(s, a, s') + \gamma V_{k-1}^*(s'))$

Learning Distribution from Value Iteration

- Initialize with $V_0^*(s) = 0, \forall s$
- For each horizon step $k = 1,2,\ldots,H$
  - For all states s

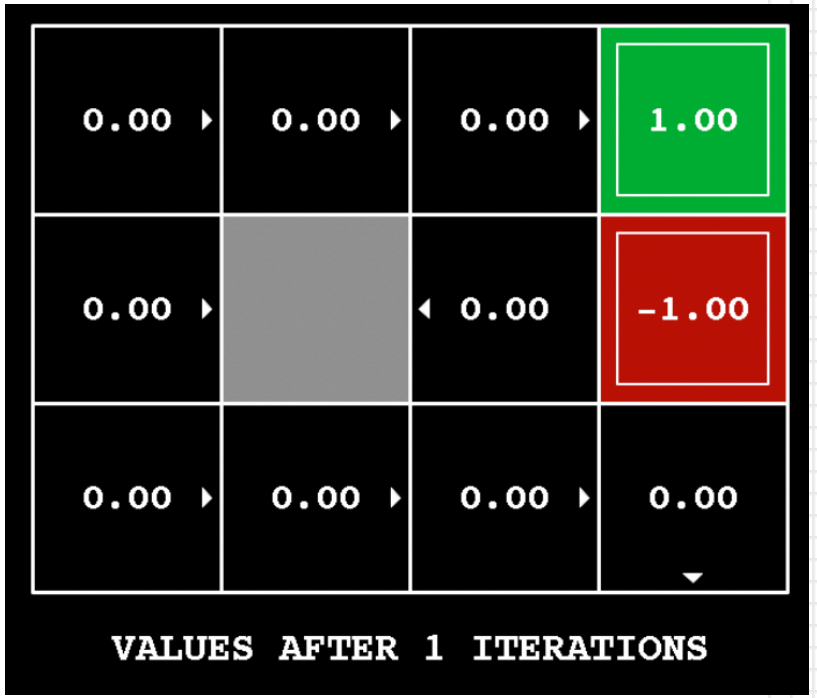
Bellman update

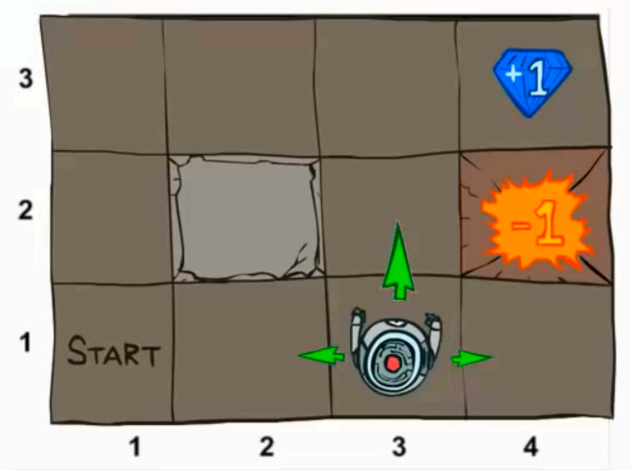
$$V_k^*(s) \leftarrow \max_a \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma V_{k-1}^*(s'))$$

$$\pi_k^*(s) \leftarrow \text{argmax}_a \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma V_{k-1}^*(s'))$$

Demo

$$V_k^*(s) \leftarrow \max_a \sum_{s'} P(s' \mid s, a)(R(s, a, s') + \gamma V_{k-1}^*(s'))$$

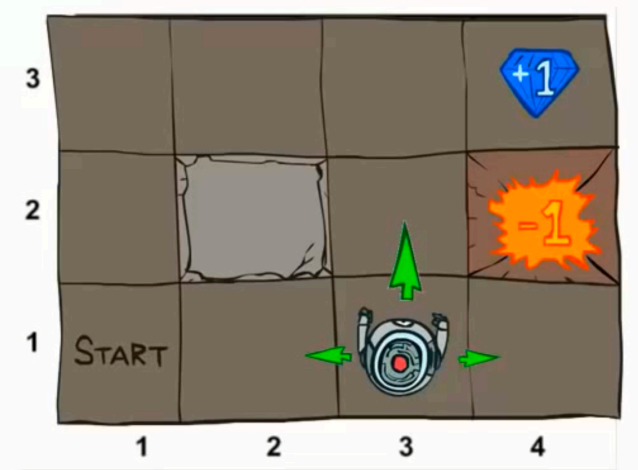


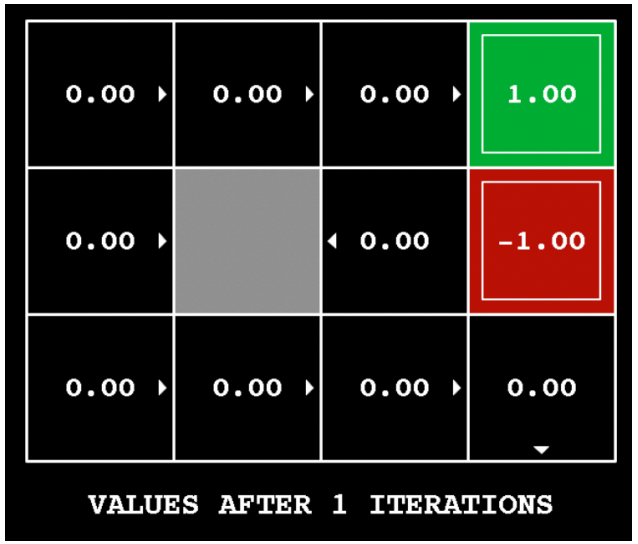


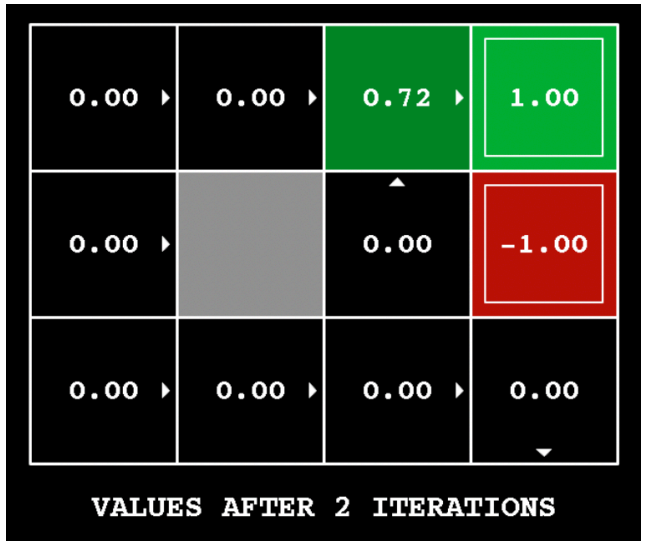
Demo

$$V_k^*(s) \leftarrow \max_a \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma V_{k-1}^*(s'))$$

Noise=0.2, Discount=0.9



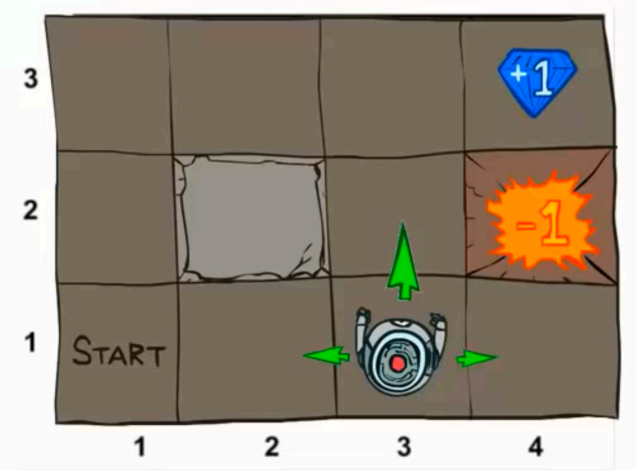


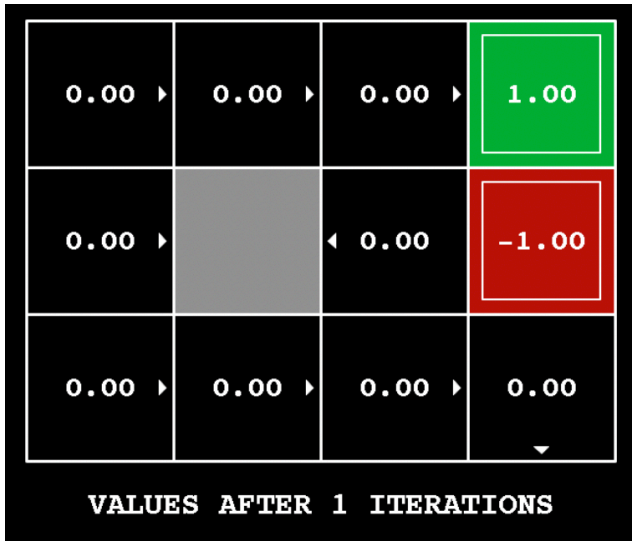


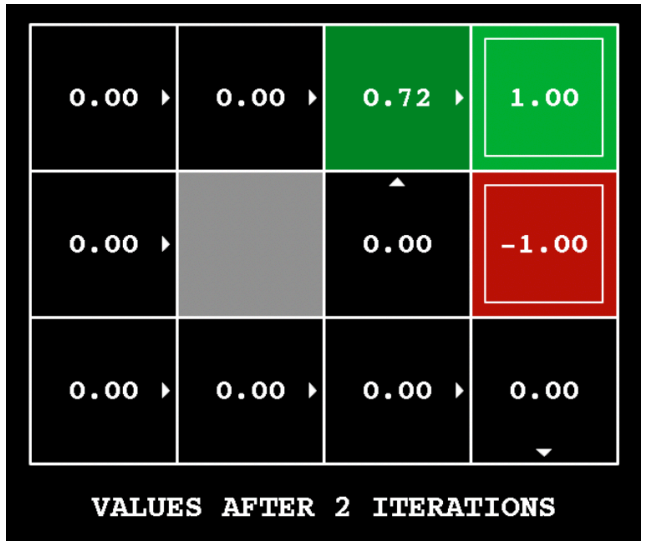
Demo

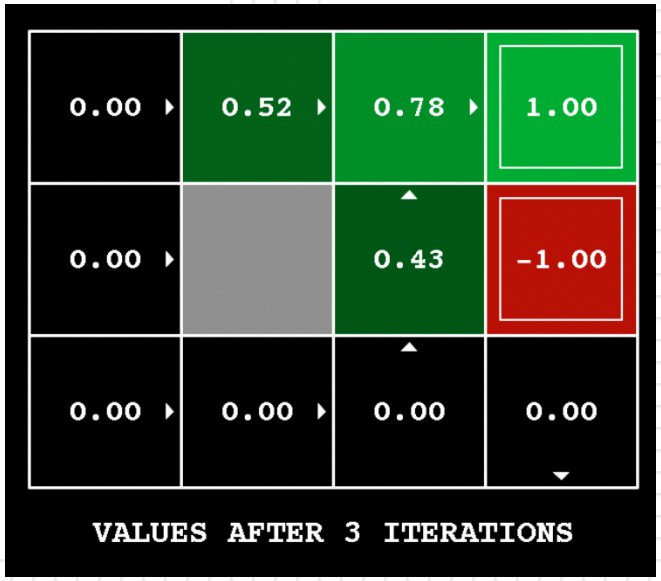
$$V_k^*(s) \leftarrow \max_a \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma V_{k-1}^*(s'))$$

Noise=0.2, Discount=0.9





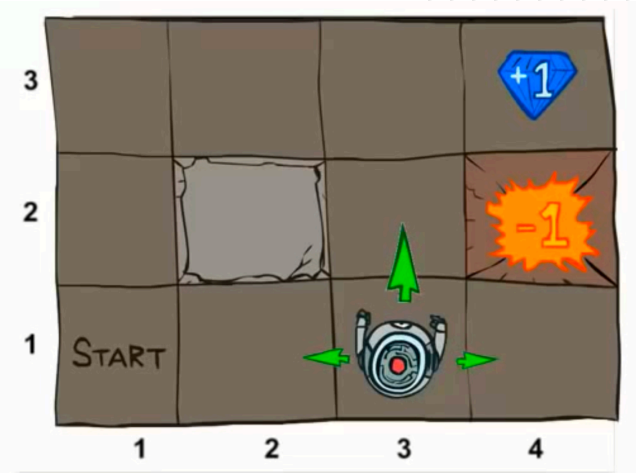


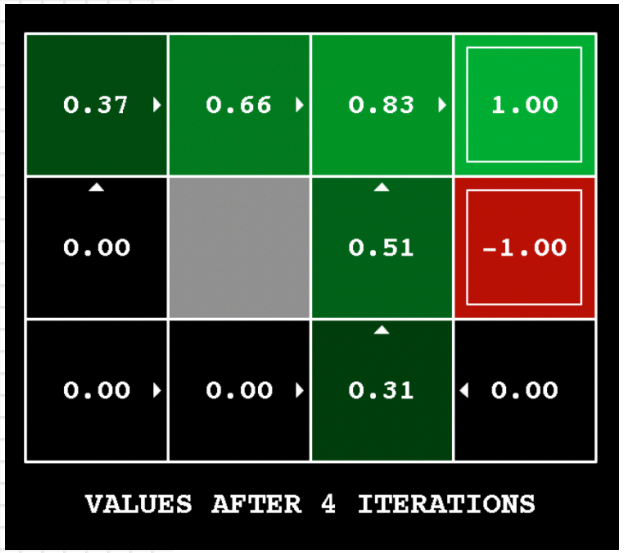


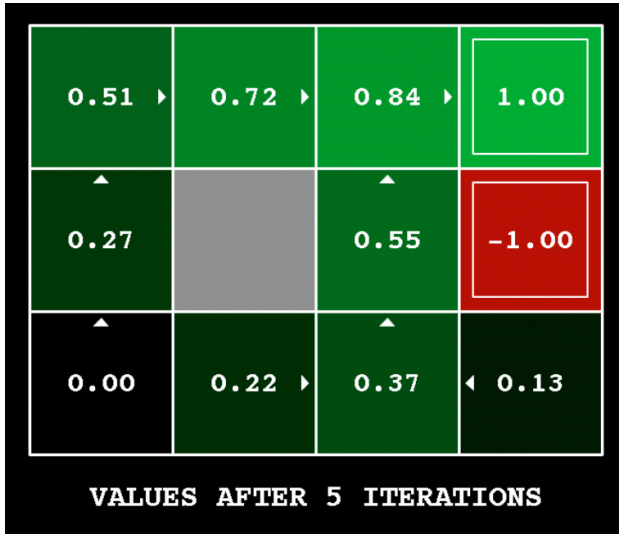
Demo

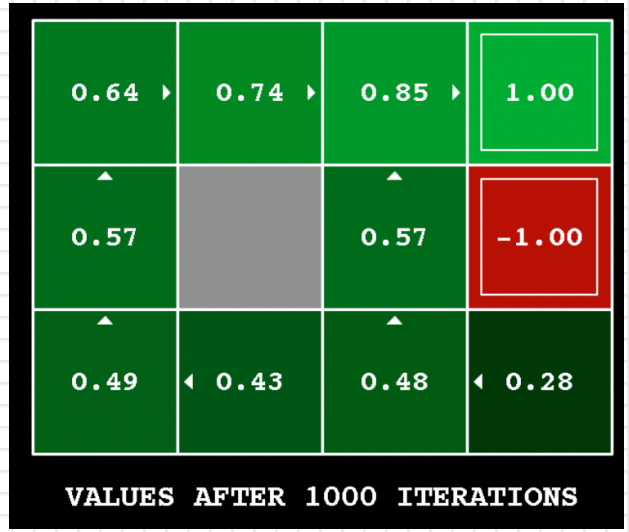
$$V_k^*(s) \leftarrow \max_a \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma V_{k-1}^*(s'))$$

Noise=0.2, Discount=0.9









Value Iteration Converges

- $V^*(s)$ = expected sum of rewards from state s for ∞ steps
- $V_H^*(s)$ = expected sum of rewards from state s for H steps

- Additional reward over time step $H+1, H+2, \ldots$

$$\gamma^{H+1}R(s_{H+1}) + \gamma^{H+2}R(s_{H+2}) + \ldots \leq \gamma^{H+1}R_{max} + \gamma^{H+2}R_{max} + \ldots = \frac{\gamma^{H+1}}{1-\gamma}R_{max}$$

Converges to 0 when $H \to \infty$

and $V_H^* \to V^*$

Q-Values

- $Q^*(s, a) =$ expected utility startng in s, taking action a, and (thereafter) acting optimally
- Bell equation:

$$Q^*(s,a) = \sum_{s'} P(s' \mid s, a)(R(s,a,s') + \gamma \max_{a'} Q^*(s', a'))$$

- Q-value iteration

$$Q^*_{k+1}(s,a) \leftarrow \sum_{s'} P(s' \mid s, a)(R(s,a,s') + \gamma \max_{a'} Q^*_k(s', a'))$$

Policy Evaluation

- Q-value iteration

$$Q^*_{k+1}(s,a) \leftarrow \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma \max_{a'} Q^*_k(s',a'))$$

- Value iteration

$$V^*_k(s) \leftarrow \max_a \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma V^*_{k-1}(s'))$$

- Policy evaluation for a given policy $\pi(s)$

$$V^\pi_k(s) \leftarrow \sum_{s'} P(s'|s,\pi(s))(R(s,\pi(s),s') + \gamma V^\pi_{k-1}(s'))$$

When converged

$$\forall s,\ V^\pi(s) \leftarrow \sum_{s'} P(s'|s,\pi(s))(R(s,\pi(s),s') + \gamma V^\pi(s'))$$

Policy Iteration Guarantees

- Policy iteration guarantees convergence, where the current policy and its value function are the optimal policy and the optimal value function
- Guarantee Improvement
- Optimal at convergence
- Demo: https://www.datascienceblog.net/post/reinforcement-learning/mdps_dynamic_programming/