## Value Iteration Algorithm

- For each state s, figure out the expected reward of starting in s and acting optimally.

→ Use the Bellman Equation (value function) ⊛

↳ Value function near high-reward states will be large

get only "best" action

$$⊛ \quad V^*(s) = \max_a \underbrace{\sum_{s'} T(s,a,s')}_{\text{transition function}} [\overset{\text{reward}}{R}(s,a,s') + \gamma V^*(s')]$$

→ reward

→ value function for starting in s' (future reward)

state I'm in

action I took

state I ended up in

The reason this is needed is because we can't remove stochasticity completely: the agent has a high chance of performing the desired action, but there's also a chance that they don't.

Algorithm:

1. Initialize all $V^*(s)$ to 0 (except the reward states)
2. While not converged
   a. For each state compute $V^*(s)$

$$V^*_{k+1}(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*_k(s')]$$

Convergence criterion

Policy extraction: max → argmax

$$\pi^*(s) = \operatorname{argmax} \sum T(s,a,s')[R(s,a,s') + \gamma V^*_k(s')]$$

## Το πρόβλημα με το αυτόνομο robot

| | 1 | 2 | 3 |
|---|---|---|---|
| 2 | | | 10 |
| 1 | | | -10 |

intended dir: 0.8

$\perp$ dir: 0.1

backwards: 0.0

$\gamma = 0.9$

α) k=1:

$$\begin{pmatrix} 0 & 0 & 10 \\ 0 & 0 & -10 \end{pmatrix} \text{ (initialization)}$$

k=2: Τα (1,1), (2,1) δεν γίνονται update γιατί στις γειτονιές τους έχουν μηδενικά.

(1,2): Βέλτιστη δράση = αριστερά, γιατί αν πάει πάνω τότε έχει 10% πιθανότητα να καταλήξει στο -10

(2,2): Βέλτιστη δράση = δεξιά, με $V = 0.9(0.8 \times 10 + 0.1 \times 0 + 0.1 \times 0) = 7.2$

$$\begin{pmatrix} 0 & 7.2 & 10 \\ 0 & 0 & -10 \end{pmatrix}$$

---

k=3: (Τυπικά η εκφώνηση θέλει μέχρι k=2)

(1,1): No updates

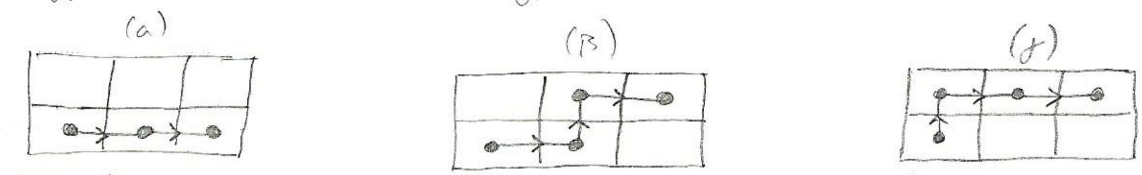
(1,2): Βέλτιστη δράση = πάνω:

$$V = 0.9(7.2 \times 0.8 + (-10) \cdot 0.1 + 0 \times 0.1) = 4.284$$

(2,1): Βέλτιστη δράση = δεξιά: $V = 0.9(7.2 \times 0.8 + 0 \times 0.1 + 0 \times 0.1) = 5.184$

(2,2): Βέλτιστη δράση = δεξιά: $V = 0.9(10 \times 0.8 + 0 \times 0.1 + \overset{\text{τοίχος}}{7.2} \times 0.1) = 7.848$

β) Σε ό,τι αφορά το (β) σκέλος της άσκησης, υπάρχουν 2 δυνατές προσεγγίσεις. Σε κάθε προσέγγιση, ισχύουν:




Reward = −10 Reward = +10 Reward = +10

Προσέγγιση #1: Αρχικά $V = 0$ παντού. Μετά το πείραμα (α):

$$V_2 = V_1 + \alpha(U_1 - V_1) \Rightarrow V_2 = -10\alpha, \text{ για τα } (1,1), (1,2).$$

0 για τα άλλα

Μετά το (β):

| | | |
|---|---|---|
| 0 | 0 | 10 |
| $-10\alpha$ | $-10\alpha$ | $-10$ |

$$V_3 = -10\alpha + \alpha(10 + 10\alpha) = 10\alpha^2 \longrightarrow$$

| | | |
|---|---|---|
| 0 | $10\alpha$ | 10 |
| $10\alpha^2$ | $10\alpha^2$ | $-10$ |

Μετά το (γ):

| | | |
|---|---|---|
| 0 | $10\alpha$ | 10 |
| $10\alpha^2$ | $10\alpha^2$ | $-10$ |

$\longrightarrow$

| | | |
|---|---|---|
| $10\alpha$ | $20\alpha - 10\alpha^2$ | 10 |
| $10\alpha + 10\alpha^2 - 10\alpha^3$ | $10\alpha^2$ | $-10$ |

Άρα οι τιμές των κελιών (1,1) και (2,2) είναι $10a + 10a^2 - 10a^3$ και $20a - 10\alpha^2$, αντίστοιχα, όπου $\alpha$: learning rate

Προσέγγιση #2: Το (1,1) εμφανίζεται και στα 3 πειράματα, άρα

$$V(1,1) = \frac{-10+10+10}{3} = 10/3$$

Το (2,2) εμφανίζεται στα 2 πειράματα με reward +10, άρα

$$V(2,2) = \frac{+10+10}{2} = 10$$

First visit MC vs Every visit MC