CS410: Artificial Intelligence 2020 Fall
Homework 6: MDP
Due date: 23:59:59 (GMT +08:00), December 25, 2020

Chunyu Xue 518021910698

# 1 Micro-Blackjack

In micro-blackjack, you repeatedly draw a card (with replacement) that is equally likely to be a 2, 3, or 4. You can either Draw or Stop if the total score of the cards you have drawn is less than 6. If your total score is 6 or higher, the game ends, and you receive a utility of 0. When you Stop, your utility is equal to your total score (up to 5), and the game ends. When you Draw, you receive no utility. There is no discount ($\gamma = 1$). Let's formulate this problem as an MDP with the following states: 0, 2, 3, 4, 5 and a Done state, for when the game ends.

(a) What is the transition function and the reward function for this MDP?

The transition function is:

$T(s,\ Stop,\ Done) = 1$

$T(0,\ Draw,\ s') = \frac{1}{3},\quad s' \in \{2, 3, 4\}$

$T(2,\ Draw,\ s') = \frac{1}{3},\quad s' \in \{4, 5, Done]$

$$T(3,\ Draw,\ s') = \begin{cases} \frac{1}{3}, & s' = 5 \\ \frac{2}{3}, & s' = Done \end{cases}$$

$T(4,\ Draw,\ Done) = 1$

$T(5,\ Draw,\ Done) = 1$

$T(s,\ a,\ s') = 0$, otherwise

The reward function is:

$R(s,\ Stop,\ Done) = s,\quad s \leq 5$

$R(s,\ a,\ s') = 0$, otherwise

(b) Fill in the following table of value iteration values for the first 4 iterations.

| States | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $V_0$ | 0 | 0 | 0 | 0 | 0 |
| $V_1$ | 0 | 2 | 3 | 4 | 5 |
| $V_2$ | 3 | 3 | 3 | 4 | 5 |
| $V_3$ | 10/3 | 3 | 3 | 4 | 5 |
| $V_4$ | 10/3 | 3 | 3 | 4 | 5 |

(c) You should have noticed that value iteration converged above. What is the optimal policy for the MDP?

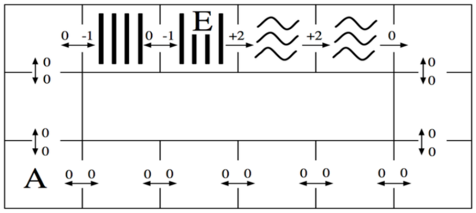
| States | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\pi^*$ | Draw | Draw | Stop | Stop | Stop |

(d) Perform one iteration of policy iteration for one step of this MDP, starting from the fixed policy below:

| States | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\pi_i$ | Draw | Stop | Draw | Stop | Draw |
| $V^{\pi_i}$ | 2 | 2 | 0 | 4 | 0 |
| $\pi_{i+1}$ | Draw | Stop | Stop | Stop | Stop |

## 2 Grid-World Water Park

Consider the MDP drawn below. The state space consists of all squares in a grid-world water park. There is a single waterslide that is composed of two ladder squares and two slide squares (marked with vertical bars and squiggly lines respectively). An agent in this water park can move from any square to any neighboring square, unless the current square is a slide in which case it must move forward one square along the slide. The actions are denoted by arrows between squares on the map and all deterministically move the agent in the given direction. The agent cannot stand still: it must move on each time step. Rewards are also shown below: the agent feels great pleasure as it slides down the water slide (+2), a certain amount of discomfort as it climbs the rungs of the ladder (-1), and receives rewards of 0 otherwise. The time horizon is infinite; this MDP goes on forever.



(a) How many (deterministic) policies $\pi$ are possible for this MDP?

**Answer:** $2^{11}$.

(b) Fill in the blank cells of this table with values that are correct for the corresponding function, discount, and state. *Hint: You should not need to do substantial calculation here.*

| | $\gamma$ | s=A | s=E |
|---|---|---|---|
| $V_3^*(s)$ | 1.0 | 0 | 4 |
| $V_{10}^*(s)$ | 1.0 | 2 | 4 |
| $V_{10}^*(s)$ | 0.1 | 0 | 2.2 |
| $Q_1^*(s, west)$ | 1.0 | ——— | 0 |
| $Q_{10}^*(s, west)$ | 1.0 | ——— | 3 |
| $V^*(s)$ | 1.0 | $\infty$ | $\infty$ |
| $V^*(s)$ | 0.1 | 0 | 2.2 |

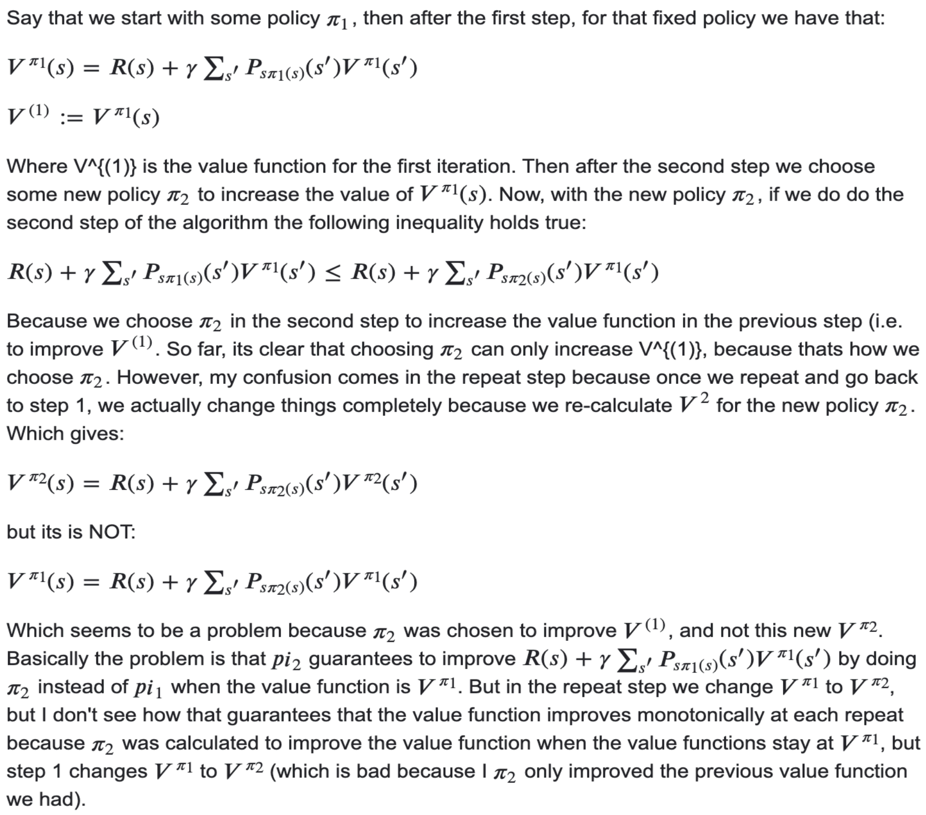
# 3 Analysis of value iteration and policy iteration

(a) Please give an example where the value iteration does not converge when the discount $\gamma = 1$.

**Answer**: The value iteration will fail to converge when the action space is too large, the state space is too large or random interference.

(b) Try to prove the policy improvement method can indeed improve the previous policy and then prove its convergence.

**Prove:**



Say that we start with some policy $\pi_1$, then after the first step, for that fixed policy we have that:

$V^{\pi 1}(s) = R(s) + \gamma \sum_{s'} P_{s\pi 1(s)}(s')V^{\pi 1}(s')$

$V^{(1)} := V^{\pi 1}(s)$

Where V^{(1)} is the value function for the first iteration. Then after the second step we choose some new policy $\pi_2$ to increase the value of $V^{\pi 1}(s)$. Now, with the new policy $\pi_2$, if we do do the second step of the algorithm the following inequality holds true:

$R(s) + \gamma \sum_{s'} P_{s\pi 1(s)}(s')V^{\pi 1}(s') \leq R(s) + \gamma \sum_{s'} P_{s\pi 2(s)}(s')V^{\pi 1}(s')$

Because we choose $\pi_2$ in the second step to increase the value function in the previous step (i.e. to improve $V^{(1)}$. So far, its clear that choosing $\pi_2$ can only increase V^{(1)}, because thats how we choose $\pi_2$. However, my confusion comes in the repeat step because once we repeat and go back to step 1, we actually change things completely because we re-calculate $V^2$ for the new policy $\pi_2$. Which gives:

$V^{\pi 2}(s) = R(s) + \gamma \sum_{s'} P_{s\pi 2(s)}(s')V^{\pi 2}(s')$

but its is NOT:

$V^{\pi 1}(s) = R(s) + \gamma \sum_{s'} P_{s\pi 2(s)}(s')V^{\pi 1}(s')$

Which seems to be a problem because $\pi_2$ was chosen to improve $V^{(1)}$, and not this new $V^{\pi 2}$. Basically the problem is that $pi_2$ guarantees to improve $R(s) + \gamma \sum_{s'} P_{s\pi 1(s)}(s')V^{\pi 1}(s')$ by doing $\pi_2$ instead of $pi_1$ when the value function is $V^{\pi 1}$. But in the repeat step we change $V^{\pi 1}$ to $V^{\pi 2}$, but I don't see how that guarantees that the value function improves monotonically at each repeat because $\pi_2$ was calculated to improve the value function when the value functions stay at $V^{\pi 1}$, but step 1 changes $V^{\pi 1}$ to $V^{\pi 2}$ (which is bad because I $\pi_2$ only improved the previous value function we had).

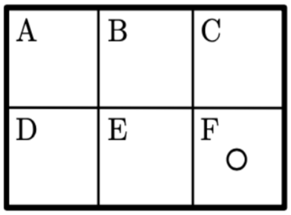
Therefore, the policy improvement method can indeed improve the previous policy, and it is indeed converged.

# 4 MDP

Pacman is using MDPs to maximize his expected utility. In each environment:

- Pacman has the standard actions {North, East, South, West} unless blocked by an outer wall
- There is a reward of 1 point when eating the dot (for example, in the grid below, $R(C, South, F) = 1$)
- The game ends when the dot is eaten

(a) Consider the following grid where there is a single food pellet in the bottom right corner (F). The discount factor is 0.5. There is no living reward. The states are simply the grid locations.



(a.i) What is the optimal policy for each state?

| State | $\pi$(state) |
|---|---|
| A | East or South |
| B | East or South |
| C | South |
| D | East |
| E | East |

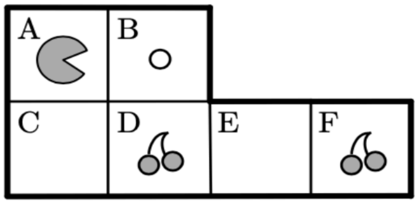
(a.ii) What is the optimal value for the state of being in the upper left corner (A)? Reminder: the discount factor is 0.5.

$V^*(A)$=0.25

(a.iii) Using value iteration with the value of all states equal to zero at $k = 0$, for which iteration $k$ will $V_k(A) = V^*(A)$ ?

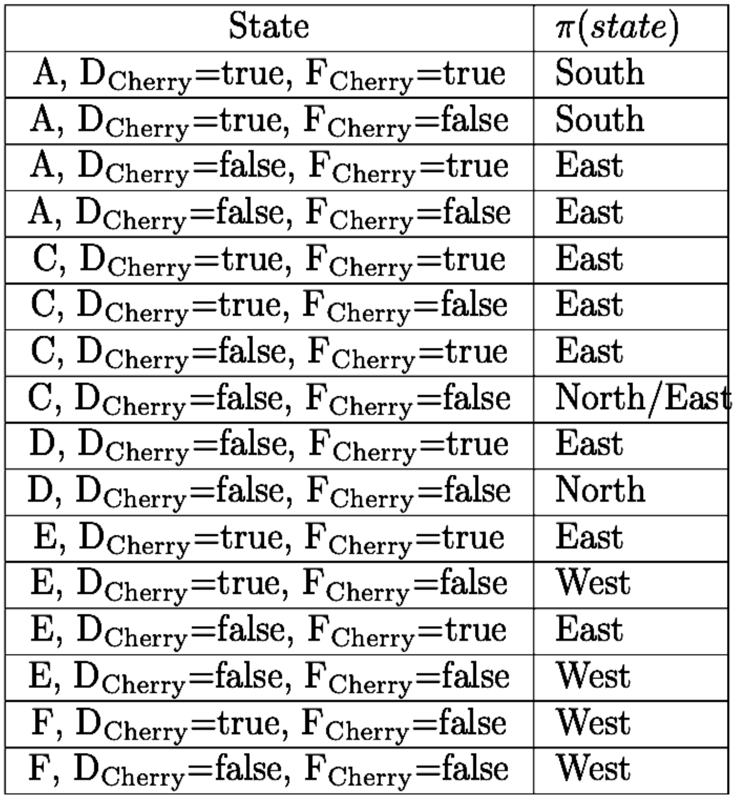
$k = 3$

(b) Consider a new Pacman level that begins with cherries in locations D and F. Landing on a grid position with cherries is worth 5 points and then the cherries at that position disappear. There is still one dot, worth 1 point. The game still only ends when the dot is eaten.



(b.i) With no discount ($\gamma = 1$) and a living reward of -1, what is the optimal policy for the states in this level's state space?

**Answer:**



| State | $\pi(state)$ |
|---|---|
| A, $\text{D}_{\text{Cherry}}$=true, $\text{F}_{\text{Cherry}}$=true | South |
| A, $\text{D}_{\text{Cherry}}$=true, $\text{F}_{\text{Cherry}}$=false | South |
| A, $\text{D}_{\text{Cherry}}$=false, $\text{F}_{\text{Cherry}}$=true | East |
| A, $\text{D}_{\text{Cherry}}$=false, $\text{F}_{\text{Cherry}}$=false | East |
| C, $\text{D}_{\text{Cherry}}$=true, $\text{F}_{\text{Cherry}}$=true | East |
| C, $\text{D}_{\text{Cherry}}$=true, $\text{F}_{\text{Cherry}}$=false | East |
| C, $\text{D}_{\text{Cherry}}$=false, $\text{F}_{\text{Cherry}}$=true | East |
| C, $\text{D}_{\text{Cherry}}$=false, $\text{F}_{\text{Cherry}}$=false | North/East |
| D, $\text{D}_{\text{Cherry}}$=false, $\text{F}_{\text{Cherry}}$=true | East |
| D, $\text{D}_{\text{Cherry}}$=false, $\text{F}_{\text{Cherry}}$=false | North |
| E, $\text{D}_{\text{Cherry}}$=true, $\text{F}_{\text{Cherry}}$=true | East |
| E, $\text{D}_{\text{Cherry}}$=true, $\text{F}_{\text{Cherry}}$=false | West |
| E, $\text{D}_{\text{Cherry}}$=false, $\text{F}_{\text{Cherry}}$=true | East |
| E, $\text{D}_{\text{Cherry}}$=false, $\text{F}_{\text{Cherry}}$=false | West |
| F, $\text{D}_{\text{Cherry}}$=true, $\text{F}_{\text{Cherry}}$=false | West |
| F, $\text{D}_{\text{Cherry}}$=false, $\text{F}_{\text{Cherry}}$=false | West |

(b.ii) With no discount ($\gamma = 1$), what is the range of living reward values such that Pacman eats exactly one cherry when starting at position A?

**Answer:** Valid range for the living reward is (-2.5,-1.25).

# 5 How do you Value It(eration)

(a) Fill out the following True/False questions.

1. Let $A$ be the set of all actions and $S$ the set of states for some MDP. Assuming that $|A| << |S|$, one iteration of value iteration is generally faster than one iteration of policy iteration that solves a linear system during policy evaluation.
2. For any MDP, changing the discount factor does not affect the optimal policy for the MDP.

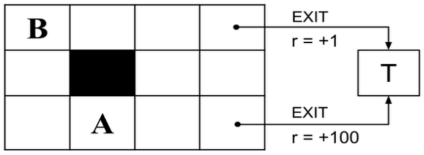
**Answer:**

1. True.
2. False.

The following problem will take place in various instances of a grid world MDP. Shaded cells represent walls. In all states, the agent has available actions $\uparrow, \downarrow, \leftarrow, \rightarrow$. Performing an action that would transition to an invalid state (outside the grid or into a wall) results in the agent remaining in its original state. In states with an arrow coming out, the agent has an additional action EXIT. In the event that the EXIT action is taken, the agent receives the labeled reward and ends the game in the terminal state T . Unless otherwise stated, all other transitions receive no reward, and all transitions are deterministic.

For all parts of the problem, assume that value iteration begins with all states initialized to zero, i.e., $V_0(s) = 0, \forall s$. Let the discount factor be $\gamma = 0.5$ for all following parts.

(b) Suppose that we are performing value iteration on the grid world MDP below.



(b.i) What are the optimal values for A and B?

**Answer:**
$V^*(A) =$ _25___; $V^*(B) =$ _$\frac{25}{8}$___;

(b.ii) After how many iterations $k$ will we have $V_k(s) = V^*(s)$ for all states $s$? If it never occurs, write "never". Write your answer below.

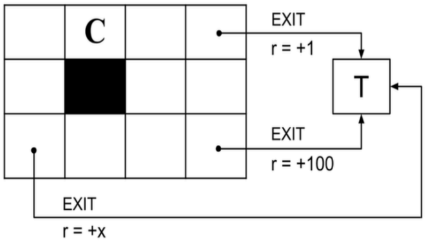
**Answer:** 6.

(b.iii) Suppose that we wanted to re-design the reward function. For which of the following new reward functions would the optimal policy remain unchanged? Let $R(s, a, s')$ be the original reward function.

- $R_1(s, a, s') = 10R(s, a, s')$
- $R_2(s, a, s') = 1 + R(s, a, s')$
- $R_3(s, a, s') = R^2(s, a, s')$
- $R_4(s, a, s') = -1$
- None

**Answer:** 1, 2, 3.

(c) For the following problem, we add a new state in which we can take the EXIT action with a reward of $+x$.



(c.i) For what values of $x$ is it guaranteed that our optimal policy $\pi^*$ has $\pi^*(C) = \leftarrow$? Write $\infty$ and $-\infty$ if there is no upper or lower bound, respectively. Write the upper and lower bounds in each respective box.

_50_ $< x <$ ___$\infty$_ _

(c.ii) For what values of $x$ does value iteration take the minimum number of iterations $k$ to converge to $V^*$ for all states? Write $\infty$ and $-\infty$ if there is no upper or lower bound, respectively. Write the upper and lower bounds in each respective box.

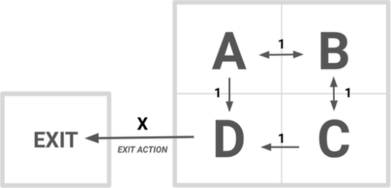
___50_ __ $\leq x \leq$ ___200_ _

(c.iii) Fill the box with value $k$, the minimum number of iterations until $V_k$ has converged to $V^*$ for all states.

$k =$ ___4___

# 6 Strange MDPs

In this MDP, the available actions at state A, B, C are LEFT, RIGHT, UP, and DOWN unless there is a wall in that direction. The only action at state D is the EXIT ACTION and gives the agent a reward of x. The reward for non-exit actions is always 1.



(a) Let all actions be deterministic. Assume $\gamma = 0.5$ . Express the following in terms of x.

**Answer:**

$$V^*(D) = \_x\_; \quad V^*(C) = \_max\{1+0.5x, 2\}\_$$

$$V^*(A) = \_max\{1+0.5x, 2\}\_\_\_; \quad V^*(B) = \_\max\{1+0.5(1+0.5x), 2\}\_\_$$

(b) Let any non-exit action be successful with probability $= 0.5$ . Otherwise, the agent stays in the same state with reward $= 0$. The EXIT ACTION from the state D is still deterministic and will always succeed. Assume that $\gamma = 0.5$.

For which value of x does $Q^*(A, DOWN) = Q^*(A, RIGHT)$? Box your answer and justify/show your work.

**Answer:**

$$Q^*(A, \text{Down}) = Q^*(A, \text{Right})$$

$$\Rightarrow V^*(A) = Q^*(A, \text{Down}) = Q^*(A, \text{Right})$$

① $V^*(A) = Q^*(A, \text{Down}) = \frac{1}{2}(0 + \frac{1}{2}V^*(A)) + \frac{1}{2}(1 + \frac{1}{2}x)$

$= \frac{1}{2} + \frac{1}{4}V^*(A) + \frac{1}{4}x$

② $V^*(A) = \frac{2}{3} + \frac{1}{3}x$

③ $V^*(A) = Q^*(A, \text{Right}) = \frac{1}{2}(0, \frac{1}{2}V^*(A)) + \frac{1}{2}(1 + \frac{1}{2}V^*(B))$

$= \frac{1}{2} + \frac{1}{4}V^*(A) + \frac{1}{4}V^*(B)$

④ $V^*(A) = \frac{2}{3} + \frac{1}{3}V^*(B)$

⑤ $V^*(B) = \frac{1}{2}(0 + \frac{1}{2}V^*(B)) + \frac{1}{2}(1 + \frac{1}{2}V^*(A))$

$= \frac{1}{2} + \frac{1}{4}V^*(B) + \frac{1}{4}V^*(A)$

⑥ $V^*(B) = \frac{2}{3} + \frac{1}{3}V^*(A)$

②④⑥ $\Rightarrow$ $x = 1$

Therefore, the value of x is 1.

(c) We now add one more layer of complexity. Turns out that the reward function is not guaranteed to give a particular reward when the agent takes an action. Every time an agent transitions from one state to another, once the agent reaches the new state s', a fair 6-sided dice is rolled. If the dices lands with value x, the agent receives the reward $R(s, a, s') + x$. The sides of dice have value 1, 2, 3, 4, 5 and 6.

Write down the new bellman update equation for $V_{k+1}(s)$ in terms of $T(s, a, s')$, $R(s, a, s')$, $V_k(s')$, and $\gamma$.

$$V_{k+1}(s) = \max_a \sum_{s'} T(s, a, s')\left[\frac{1}{6}\left(\sum_{i=1}^{6} R(s, a, s') + i\right) + \gamma V_k(s)\right]$$

$$= \max_a \sum_{s'} T(s, a, s')\left(R(s, a, s') + 3.5 + \gamma V_k(s)\right)$$