תרגיל בית 3 – MDP ומבוא ללמידה

עברו על כלל ההנחיות לפני תחילת התרגיל.

הנחיות כלליות:

- תאריך ההגשה: לחלק א' של התרגיל (MDP) – עד ליום האחרון של הסמסטר - **08/04/2024** ב23:59
  לחלק ב' של התרגיל (מבוא ללמידה) – עד לסוף מועדי א' - **17/05/2024** ב23:59
- את המטלה יש להגיש **בזוגות בלבד**.
- יש להגיש מטלות מוקלדות בלבד. פתרונות בכתב יד לא ייבדקו.
- ניתן לשלוח שאלות בנוגע לתרגיל בפיאצה בלבד.
- המתרגל האחראי על תרגיל זה: **דניאל אלגריסי**.
- בקשות דחיה מוצדקות (מילואים, אשפוז וכו') יש לשלוח למתרגל האחראי (**ספיר טובול**) בלבד.
- במהלך התרגיל ייתכן שנעלה עדכונים, למסמך הנ"ל – תפורסם הודעה בהתאם.
- העדכונים הינם מחייבים, ועליכם להתעדכן עד מועד הגשת התרגיל.
- שימו לב, העתקות תטופלנה בחומרה.
- התשובות לסעיפים בהם מופיע הסימון ✍ צריכים להופיע בדוח.
- לחלק הרטוב מסופק שלד של הקוד.
- אנחנו קשובים לפניות שלכם במהלך התרגיל ומעדכנים את המסמך הזה בהתאם. גרסאות עדכניות של המסמך יועלו לאתר. הבהרות ועדכונים שנוספים אחרי הפרסום הראשוני יסומנו כאן בצהוב. ייתכן שתפורסמנה גרסאות רבות – אל תיבהלו מכך. השינויים בכל גרסה יכולים להיות קטנים.

לצורך הנוחות:

הבהרות ועדכונים גרסה ראשונה סומנו ככה.

הבהרות ועדכונים גרסה שניה סומנו ככה.

שימו לב שאתם משתמשים רק בספריות הפייתון המאושרות בתרגיל (מצוינות בתחילת כל חלק רטוב)
לא יתקבל קוד עם ספריות נוספות

מומלץ לחזור על שקפי ההרצאות והתרגולים הרלוונטיים לפני תחילת העבודה על התרגיל.

חלק א' – MDP (44 נק')

רקע

בחלק זה נעסוק בתהליכי החלטה מרקובים, נתעניין בתהליך עם **אופק אינסופי** (מדיניות סטציונרית).

חלק א' - חלק היבש ✍

1. בתרגול ראינו את משוואת בלמן כאשר התגמול ניתן עבור המצב הנוכחי בלבד, כלומר $R: S \to \mathbb{R}$, למתן תגמול זה נקרא "תגמול על הצמתים" מכיוון שהוא תלוי בצומת שהסוכן נמצא בו.
בהתאם להגדרה זו הצגנו בתרגול את האלגוריתמים Value iteration ו-Policy Iteration למציאת המדיניות האופטימלית.

כעת, נרחיב את ההגדרה הזו, לתגמול המקבל את המצב הנוכחי והמצב אליו הגיע הסוכן, כלומר: $R: S \times S \to \mathbb{R}$, למתן תגמול זה נקרא "תגמול תוצאתי". לצורך שלמות ההגדרה, נגדיר שאם לכל $a \in A$ מתקיים - $P(s'|s,a) = 0$ אז $R(s,s') = -\infty$.

א. (1 נק') התאימו את הנוסחה של התוחלת של התועלת מהתרגול, עבור התוחלת של התועלת המתקבלת במקרה של " תגמול תוצאתי ", אין צורך לנמק.

$$U^{\pi}(s) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^t \, R(s_t, s_{t+1}) | s_0 = s]$$

ב. (1 נק') כתבו מחדש את נוסחת משוואת בלמן עבור המקרה של " תגמול תוצאתי ", אין צורך לנמק.

$$U(s) = \max_{a \in A(s)} \sum_{s' \in S} P(s'|s,a) * (R(s,s') + \gamma U(s'))$$

בסעיפים הבאים התייחסו גם למקרה בו $\gamma = 1$, והסבירו מה לדעתכם התנאים שצריכים להתקיים על הסביבה\mdp על מנת שתמיד נצליח למצוא את המדיניות האופטימלית.

ג. (2 נק') נסחו את אלגוריתם Value Iteration עבור המקרה של " תגמול תוצאתי ".

For $\boldsymbol{\gamma} < \mathbf{1}$:

```
function VALUE-ITERATION(mdp, ε) returns a utility function
  inputs: mdp, an MDP with states S, actions A(s), transition model P(s' | s, a),
            rewards R(s,s') discount γ
          ε, the maximum error allowed in the utility of any state
  local variables: U, U', vectors of utilities for states in S, initially zero
                   δ, the maximum change in the utility of any state in an iteration

  repeat
     U ← U'; δ ← 0
     for each state s in S do
        U(s) = max_{a∈A(s)} Σ_{s'∈S} P(s'|s,a) * (R(s,s') + γU(s')) a) U[s']
                                                   s'
        if |U'[s] − U[s]| > δ then δ ← |U'[s] − U[s]|
  until δ < ε(1 − γ)/γ
  return U
```

For $\boldsymbol{\gamma} = \mathbf{1}$, its the same pseudo – code but the algorithm stops for $\delta < \epsilon$ or $\delta = 0$.

ד. (2 נק') נסחו את אלגוריתם Policy Iteration עבור המקרה של " תגמול תוצאתי ".

```
function POLICY-ITERATION(mdp) returns a policy
  inputs: mdp, an MDP with states S, actions A(s), transition model P(s' | s, a)
  local variables: U, a vector of utilities for states in S, initially zero
                   π, a policy vector indexed by state, initially random

  repeat
     U ← POLICY-EVALUATION(π, U, mdp)
     unchanged? ← true
     for each state s in S do
        if max_{a ∈ A(s)} Σ_{s'} 1 P(s' | s, a) U[s'] > Σ_{s'} 2 P(s' | s, π[s]) U[s'] then do
           π[s] ← argmax_{a ∈ A(s)} 3 Σ_{s'} P(s' | s, a) U[s']
           unchanged? ← false
  until unchanged?
  return π
```

Where 1,2,3 are the following:

1: $P(s'|s,a) * (R(s,s') + \gamma U(s'))$

2: $P(s'|s,\pi[s]) * (R(s,s') + \gamma U(s'))$

3: $P(s'|s,a) * (R(s,s') + \gamma U(s'))$

And POLICY-EVALUTATION changes to be the following:

$function\ POLICY - EVALUATION(\pi, U, mdp)\ returns\ a\ utility\ function$

$repeat$

$\quad \delta \leftarrow 0$

$\quad U' \leftarrow copy\ of\ U$

$\quad for\ each\ state\ s\ in\ S\ do$

$\qquad U[s] \leftarrow \Sigma s'\ P(s' \mid s, \pi[s]) * (R(s, s') + \gamma * U'[s'])$

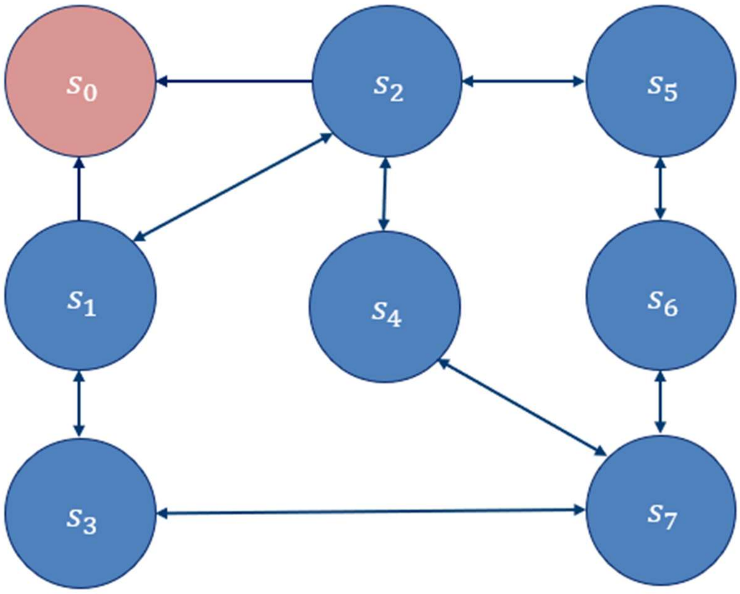
$\qquad \delta \leftarrow max(\delta, |U[s] - U'[s]|)$

$until\ \delta < \varepsilon * (1 - \gamma) / \gamma$

$return\ U$

For $\boldsymbol{\gamma = 1}$, its the same pseudo − code but the algorithm stops for $\delta < \epsilon$ or $\delta = 0$.

נתון הגרף הבא:



נתונים:

- $\gamma = 0.5$ (Discount factor).
- אופק אינסופי.
- $S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$ – קבוצת המצבים – מתארים את מיקום הסוכן בגרף.
- $S_G = \{s_0\}$ – קבוצת המצבים הסופיים.
- קבוצת הפעולות לכל מצב (על פי הגרף), לדוגמא: $A(s_3) = \{\uparrow, \rightarrow\}$.
- תגמולים ("תגמול תוצאתי"):

$$\forall s \in S, s' \in S \backslash S_G: \ \ R(s, s') = -1, \qquad R(s_1, s_0) = 5, \qquad R(s_2, s_0) = 7$$

- מודל המעבר הוא דטרמיניסטי, כלומר כל פעולה מצליחה בהסתברות אחת.

ה. (יבש 2 נק') הרץ את האלגוריתם Value iteration שכתבת על הגרף הנתון. ומלא את הערכים בטבלה הבאה, כאשר $\forall s \in S\cancel{\setminus S_G}: U_0(s) = 0$. (ייתכן שלא צריך למלא את כולה).

	$U_0(s_i)$	$U_1(s_i)$	$U_2(s_i)$	$U_3(s_i)$	$U_4(s_i)$	$U_5(s_i)$	$U_6(s_i)$	$U_7(s_i)$	$U_8(s_i)$
s_1	0	5	5	5	5	5	5	5	5
s_2	0	7	7	7	7	7	7	7	7
s_3	0	-1	1.5	1.5	1.5	1.5	1.5	1.5	1.5
s_4	0	-1	2.5	2.5	2.5	2.5	2.5	2.5	2.5
s_5	0	-1	2.5	2.5	2.5	2.5	2.5	2.5	2.5
s_6	0	-1	-1.5	0.25	0.25	0.25	0.25	0.25	0.25
s_7	0	-1	-1.5	0.25	0.25	0.25	0.25	0.25	0.25

ו. (יבש 2 נק') הרץ את האלגוריתם Policy iteration שכתבת על הגרף הנתון. ומלא את הערכים בטבלה הבאה, כאשר המדיניות ההתחלתית π_0 מופיעה בעמודה הראשונה בטבלה. (ייתכן שלא צריך למלא את כולה).

הניחו שבמידה ולא קיים שיפור, האלגוריתם יבחר תמיד להשאיר את הפעולה הקודמת.

	$\pi_0(s_i)$	$\pi_1(s_i)$	$\pi_2(s_i)$	$\pi_3(s_i)$	$\pi_4(s_i)$	$\pi_5(s_i)$	$\pi_6(s_i)$	$\pi_7(s_i)$	$\pi_8(s_i)$
s_1	↓	↑	↑	↑	↑				
s_2	↓	←	←	←	←				
s_3	→	→	↑	↑	↑				
s_4	↑	↑	↑	↑	↑				
s_5	←	←	←	←	←				
s_6	↑	↑	↑	↑	↑				
s_7	↑	↑	↑	↖	↖				

ז. (יבש 2 נק') חיזרי על הסעיף הקודם. הפעם עם **אופק סופי כאשר** $\boldsymbol{N = 2}$ (שימי לב, המדיניות לא חייבת להסתיים במצב מסיים, ישנם מצבים שלא יבולים להגיע למצב מסיים עם אופק זה. ישנם צמתים עם מספר תשובות נכונות, נקבל את כולם).

	$\pi_0(s_i)$	$\pi_1(s_i)$	$\pi_2(s_i)$	$\pi_3(s_i)$	$\pi_4(s_i)$	$\pi_5(s_i)$	$\pi_6(s_i)$	$\pi_7(s_i)$	$\pi_8(s_i)$
s_1	↓	↑	↑	↑	↑				
s_2	↓	←	←	←	←				
s_3	→	→	↑	↑	↑				
s_4	↑	↑	↑	↑	↑				
s_5	←	←	←	←	←				
s_6	↑	↑	↑	↑	↑				
s_7	↑	↑	↑	↖	↖				

ח. (1 נק') ללא תלות בשינוי של הסעיף הקודם. אם $\gamma = 0$, מה מספר המדיניות האופטימליות הקיימות? נמקו.

48. A discount factor γ which is equal to 0 means that the algorithm doesn't take into consideration future rewards, therefor it only takes into consideration the reward function for each transition. For s1 and s2 the optimal policy is to move to the terminal state s0, but for the rest of the states s3,s4,s5,s6,s7 any possible transition is considered optimal since R(s,s')=-1 for all those transitions. Therefor the answer would be a multiplication of the number of possible transitions for each state that is not s1 and s2.
-> 2*2*2*2*3=48.

ט. (1 נק') ללא תלות בשנוי של הסעיף הקודם, הסבירי מה היה קורה אם

$$R(s_1, s_2) = R(s_2, s_1) = 2, \qquad \gamma = 1$$

בתשובתך, התייחסי גם לערבי התועלות של כל צומת וגם לשינוי במדיניות, אין צורך לחשב.

The algorithm would not converge. The reason behind that is that for $\gamma = 1$ and a positive reward on the transition between the 2 states that lead to the terminal state(s_0), the U value would keep increasing for all states after each iteration, specifically for the s1 and s2 states where the policy for s1 would be to transition to s2 and for s2 it would be a transition to s1 since this policy maximizes the U value after each iteration.

The policy for the other states that are not s1 and s2 would not change, but for s1 and s2 it does and instead of moving to s0 from both states the new policy would be to get stuck in a cycle between s1 and s2.