שאלה מס' 1

א.

$$U^{\pi}(s) = E_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(S_t, \pi(S_t), S_{t+1}) \middle| S_0 = s\right]$$

ב.

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

ג. עבור המקרה בו $\gamma = 1$ נקבל ש- $\in \frac{1-\gamma}{\gamma} = 0$ כלומר האלגוריתם לא יעצור לעולם כי תמיד $\delta \geq 0$

Repeat

U ← U'; δ←0

For each state s in S do:

U'[s]← $\max_{a \in A(s)} [\sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma(U(s'))]]$

if |U'[s] – U[s]| > δ then δ ←|U'[s] – U[s]|

Until $\delta < \in (1-\gamma)/\gamma$

Return U

ד. עבור המקרה בו $\gamma = 1$ ואופק אינסופי האלגו' לא יתכנס, אםנדרוש שהאופק יהיה סופי אזי שנקבל תועלת מקסימלית והאלגו יסתיים ויחזיר מדיניות אופטימלית.

Repeat

U ← POLICY-EVALUATION(π, U, mdp)

Unchanged? ← true

For each state s in S do:

if $\max_{a \in A(s)} [\sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma(U(s'))]] > \sum_{s'} P(s'|s,\pi[s])[R(s,\pi[s],s') + \gamma(U(s'))]$

then do π [s] ← $\operatorname{argmax}_{a \in A(s)} [\sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma(U(s'))]]$

Unchanged? ← true

Until Unchanged?

Return π

שאלה 2:

חלק א' – MDP ו־RL

שאלה 2

נתונים שני אנשים – "סוחט" ו-"קורבן". בכל שלב ה"סוחט" יכול:

1) "לפרוש" – לפרוש עם רווחי הסחיטה.
2) "לסחוט" – לדרוש תשלום של 1₪ בהסתברות p , ה"קורבן" יענה לדרישה. ובהסתברות $1-p$ ה"קורבן" יסרב לשלם וידווח למשטרה.

הנחות:

- לאחר שה"קורבן" דווח למשטרה, ה"סוחט" מאבד את כל הרווחים שנצברו ואינו יכול לסחוט שוב.
- לאחר שה"סוחט" מגיע לרווחים מצטברים של n₪ הוא פורש מיד. ⇐ $A(n)=\{$לפרוש$\}$
- מטרת הסוחט היא למקסם את סכום הכסף שהוא מרוויח. $i<n$ $A(i)=\{$לפרוש, לסחוט$\}$
- אופק סופי, ניתן להניח שגדול מאוד $\bar{\gamma}=1$

1. (4 נק') נסחו את הבעיה כבעיית MDP עם המצבים $i=0,1,\dots,n$ ומצב סיום T. (0 הוא מצב התחלתי) באופן ספציפי, כתבו את המצבים, הפעולות בבכל מצב, ההסתברויות המעבר והתגמולים.
הערה: התגמולים חייבים להיות אי-שליליים. ($A(s)$ - קבוצת הפעולות במצב s)

2. (2 נק') האם ניתן לנסח את הבעיה כבעיית MDP עם מצב יחיד ומצב סיום? נמקו.
3. (2 נק') האם ניתן לנסח את הבעיה כבעיית MDP כאשר חלק מהתגמולים שליליים? נמקו.

4. (9 נק') נתון כי n=3.
כעת נרצה למצוא מדיניות אופטימליות ומה התועלת של המצב ההתחלתי כפונקציה של p.
בתשובתכם מצאו עבור אילו ערכי p נקבל כל מדיניות – מצאו את a ו-b כך שהמדיניות בטווח הנתון לא תשתנה. מלאו את הערכים החסרים בטבלה שבעמוד הבא **ונמקו** היטב את תשובתכם.
הערה: כאשר המדיניות של מצב i יכולה לקבל יותר מפעולה אחת יש לציין את כל הפעולות.

נגדיר MDP בצורה הבא : $< \mathrm{S}, \mathrm{A}, \mathrm{P}, \mathrm{R}, \gamma >$

קבוצת המצבים S : $S = \{0,1, \dots, n\} \cup \{T\}$

Set of actions A:

$$\forall 0 \le i \le n-1: A(i) = \{\text{לסחות}, \text{לפרוש}\} \ , A(n) = \{\text{לפרוש}\}, A(T) = \emptyset$$

Rewards R: *R(s,a,s')* נבחר בתגמול על הקשתות

הסבר הבחירה שלני :

שאנחנו נמצאים במצב *i* נוכל לבחור לסחוט או לפרוש והבחרה מכן תגמול על המצווים לא יעבוד, עבור הבחרה ב לסחות קיים שני מצביים להצליח ולעבור למצב *i+1* ו לא להצליח ולעבר ל *T* ולכן בחרנו ב תגמול על הקשתות .

יהי מודר בצורה הבא :

- $\forall 0 \le i \le n:\ R(i,\text{לפרוש}, T) = i$
- $\forall 0 \le i \le n-1: R(i,\text{לסחות}, i+1) = 0$
- $\forall 0 \le i \le n-1: R(i,\text{לסחות}, T) = 0$

Transitions P: P(s'|s,a)

- $\forall 0 \le i \le n:\ P(T|i,\text{לפרוש}) = 1$
- $\forall 0 \le i \le n:\ P(T|i,\text{לסחות}) = 1-p$
- $\forall 0 \le i \le n:\ P(i+1|i,\text{לסחות}) = p$

Discount factor: $\gamma = 1$

start state: $s_{init} = 0$

terminal state: $s_{final} = T$

2. (2 נק') האם ניתן לנסח את הבעיה כבעיית MDP עם מצב יחיד ומצב סיום? נמקו.

לא ניתן

נפקע בעקרון *Markov*

Action outcomes depend only on the current state.

כי עבור עובר 2 מצבים לא נוכל לדעת עבור הצעד ה i את התגמול אז נצתרך לחשיב התקמול עד הצעד הנוכי אז עבורו בעולת לסחוט לא מוצלחית נצתרך לתקמול אי שליל לא קבוע והתקמולים הם קבועים עבור כל מצב ולא יכול להיות תלוי בהסטוריה ולכן לא ניתן.

3. (2 נק') האם ניתן לנסח את הבעיה כבעיית MDP כאשר חלק מהתגמולים שליליים? נמקו.

בדומה לסעיף 1 אבל שעוברים ממצב i ל $i+1$ נוסיף אחד עבור פעולת לסחות מוצלחת

עבור פעולת לסחות לא מוצלחת ממצב i נוריד i ונעבור ל T

ועבור פרישה נוסיף 0 .

4. (9 נק') נתון כי n=3.

כעת נרצה למצוא מדיניות אופטימליות ומה התועלת של המצב ההתחלתי כפונקציה של p.

בתשובתכם מצאו עבור אילו ערכי p נקבל כל מדיניות – מצאו את a ו-b כך שהמדיניות בטווח הנתון לא תשתנה. מלאו את הערכים החסרים בטבלה שבעמוד הבא **ונמקו** היטב את תשובתכם.

הערה: כאשר המדיניות של מצב i יכולה לקבל יותר מפעולה אחת יש לציין את כל הפעולות.

מאופן ההגדרה =*A(3)* לפרוש , עבור מדוניות אופטימלית תמיד קדי לנו בהתחלה לסחות לכן *A(0)* = לסחוט

ולכן קיים רק שלוש בסלסות אפשריות וכול ש ה *p* יותר גדול יודר לנו יותר לסחוט , יכולים לרות בטבלה שמיליתי את שלושת הבלת האפשריות וחישוב התועלת בכול בולסיה נשאר לחשיב את *a* ו *b* לפי מה הספרנו קודם עבור *b* יהי יותר קידי לשתמש בה כול שי עירך *p* יותר גדול אז עבור תועלת יותר גדולה יותר קדי לשמשם בה מלשתמש מ *a*זה מתקיים קאשר *2p^2>p* מכן נקביל כי *a=0.5* באופן דומה עבור חישוב *b*

3p^3>2p^2 מכן נקביל כי *b=2/3*

$\boldsymbol{a} =$ 0.5

$\boldsymbol{b} =$ $\frac{2}{3}$

ערכי p	מדיניות	תועלות
$0 < p < \boldsymbol{a}$	$\pi_1(0) =$ לסחות $\pi_1(1) =$ לפרוש $\pi_1(2) =$ לפרוש $\pi_1(3) =$ לפרוש	$V^{\pi_1}(0) =$ p $(1-p)\cdot 0 + p\cdot 1 = p$
$\boldsymbol{a} < p < \boldsymbol{b}$	$\pi_2(0) =$ לסחות $\pi_2(1) =$ לסחות $\pi_2(2) =$ לפרוש $\pi_2(3) =$ לפרוש	$V^{\pi_2}(0) =$ $2p^2$ $(1-p)\cdot 0 + p((1-p)\cdot 0 + p\cdot 2) = 2p^2$
$\boldsymbol{b} < p < 1$	$\pi_3(0) =$ לסחות $\pi_3(1) =$ לסחות $\pi_3(2) =$ לסחות $\pi_3(3) =$ לפרוש	$V^{\pi_3}(0) =$ $3p^3$

פרק חישוב ישיר:

$$\cancel{(1-p)\cdot 0} + p(\cancel{0\cdot(1-p)} + p[\cancel{(1-p)\cdot 0} + p\cdot 3]$$

$$= p[p(p\cdot 3)] = 3p^3$$

חלק ג:

- *Done*
- *Done*
- *Done*
- *Done*
- *Done*
- גרסת *anytime*:

```
function anytime:
k=1
episodes=1
lastResult=None
        while time > 0
        result=adp_algorith(episodes)
        if (time < 0)
            return lastResult

        result = lastResult
       episodes *= 10
return lastResult
```

<u>חלק ב' – מבוא ללמידה</u>

חלק א' – יבש

א1.

עבור d=1 נקבל שאין תלות בבחירת פונק' המרחק עבור על ערך K כי שתיהן ייתנו אותו ערך:

Euclid(x, y) = $\sqrt{(x_1 - y_1)^2}$ = $|x_1 - y_1|$

Manhattan(x, y) = $|x_1 - y_1|$

א2.

עבור בעיית קלסיפיקציה נבחר k=1, d=2,נקבע את הסיווג של הדוג' לפי סיווג השכן הקרוב ביותר מאוסף דוגמאות האימון.

הסט של האימון יהיה: D= $\{d_1 =<(0,0), ->, d_2 =<(1,1.25), +>\}$

הדוג' תהיה: $t_1 = (1,1)$

לפי האוקלידי : נקבל שהסיווג של t_1 הוא – כי d_1 הוא השכן הקרוב ביותר לכן סיווגו יהיה כמוהו:

Euclid(d_1, t_1) = $\sqrt{(0-1)^2 + (0-1)^2}$ = $\sqrt{2}$ $\cong$ 1.4

Euclid(d_2, t_1) = $\sqrt{(1-1)^2 + (2.5-1)^2}$ = $\sqrt{(1.5)^2}$ $\cong$ 1.5

לפי מנהטן: נקבל שהסיווג של t_1 הוא + כי d_2 הוא השכן הקרוב ביותר לכן סיווגו יהיה כמוהו:

Manhattan(d_1, t_1) = $|0-1|$ + $|0-1|$= 2

Manhattan(d_2, t_1) = $|1-1|$ + $|2.5-1|$= 1.5

א3.

עבור k=5 ו- k=7 הדיוק מרבי על קבו' האימון ➔ 10 מתוך 14 סיווגים נכונים.

א4.

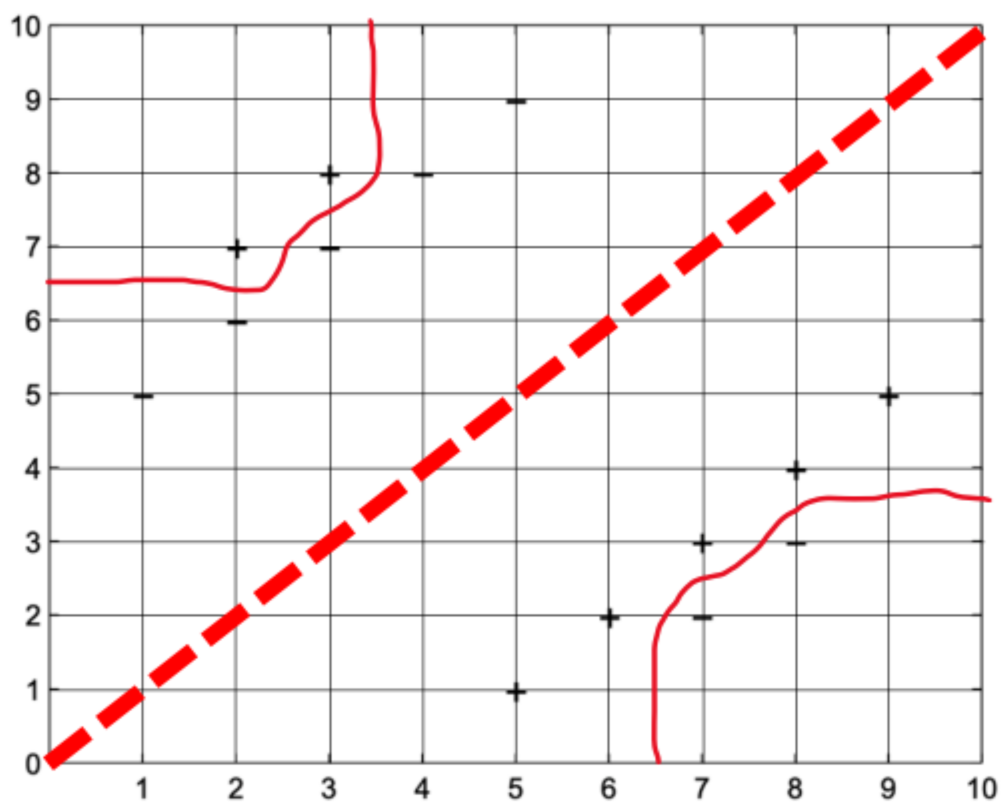
עבור k = 14 הסיווג של קבו' האימון יהיה majority כי כאשר מסווגים דוג' K השכנים הקרובים ביתור שלה הם כל דוגמאות האימון הקיימות ו- 14 הוא בעצם מס' דוגמאות האיון הקיימות. כלומר, הסיווג הוא הסיווג שיש לרוב דוגאות האימון; majoity

5) (2 נק') נמקו מדוע שימוש בערכי k גדולים או קטנים מדי יכול להיות גרוע עבור קבוצת הדגימות הנ"ל.

-תשובה: עבור ערכי k שקטנים מדי דוגמה יכולה להיספג בצורה שגויה בגלל overfitting - המודל יהיה ממש רגיש וקשה להתמודד עם רעש, שזה עלול לפגוע ברמת הדיוק המירבי .לדוגמה עבור k=3 לפי נקודות המבחן הנתונות אם נתבוננן בדוגמת מבחן שנמצאת קרוב לקצוות של אחד הצדדים)למשל בנקודה (6,1) (תסווג בהתאם לערך השגוי שלהן .עבור ערכי k שהם גדולים מדי המודל יתחשב בדוגמאות מבחן שהן לא ממש רלוונטיות לדוגמת המבחן וזה יפגע ברמת הדיוק. למשל עבור K גדול נקודת מבחן אם תיוג שלילי שנופלת באיזור הימני התחתון של הגרף תקבל סיווג חיובי בגלל כמות הפלוסים הגדולה יחסית שנמצאת סביבה, באותו אופן עבור נקודת מבחן עם ציוג חיובי שהסיווג לה נמצא באישור העליון השמאלי, היא תקבל סיווג שלילי מכיוון שיש הרבה יותר נקודות מבחן עם סיווג שלילי מאשר חיובי.) לעומת המקרה בו היינו בוחרים k קטן יותר כמו 4 למשל שהיה נותן סיווג תואם עבור מקרים אלה) .

6) (2 נק') שרטט את גבול ההחלטה של 1-nearest neighbor עבור הגרף

תשובה:



גבול ההחלטה מפריד בין איזורים בהם מסווגים דוגמאות בסימונים שונים, הסיווג לפי נקודה קרובה ביותר.