מבוא לבינה מלאכותית

תרגיל 3

דנה שבתאי 314822214

רואי גרייף 315111401

חלק א

חלק א -

שאלה 1

א) $U^{\pi}(s) \ = \ E_{\pi}[\sum_{t=0}^{\infty} \gamma^{t} R(S_{t}, \ \pi(S_{t}), \ S_{t+1})]$

ב) $U^{\pi}(s) \ = R(s, \ \pi(s), \ s') \ + \gamma \, max_{a \in A(s)} \sum_{s'} P(s' \mid s, \ \pi(s)) U(s')]$

ג)

function VALUE-ITERATION(mdp, ε)

repeat

$U \ = \ U' ; \ \delta \ = 0$

for each state s in S do

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

If $|U'(s) - U(s)| > \delta$ then $\delta \ = |U'(s) - U(s)|$

until $\delta > \frac{\varepsilon(1-\gamma)}{\gamma}$ or (γ==1 and δ==0)

return U

ד)

function POLICY-ITERATION(mdp)

repeat

U = POLICY-EVALUATION(π, U, mdp)

unchanged = true

```
      for each state s in S do
```

if $\quad max_{a\in A(s)} \sum_{s'} P(s' \mid s, a)U(s') > max_{a\in A(s)} \sum_{s'} P(s' \mid s, \pi(s))U(s')$ then do

$\pi(s) = argmax_{a\in A(s)} \sum_{s'} P(s' \mid s, a)U(s')$

```
            unchanged = false
until unchanged == true
return π
```

נשים לב שנצילח למצוא את המדיניות האופטימלית רק כאשר התכנסות מובטחת לנו. זה מתקיים כאשר מרחב המצבים סופי, מצב הפעולות סופי, קיים מצב סופי, פונקציית התגמולים חסומה, ואין מעגל תגמולים חיובי.

נתייחס למקרה בו $\gamma = 1$:

עבור VALUE-ITERATION תנאי העצירה יהיה כאשר $\delta = 0$.

מתקיים ש- $\delta = 0$ כאשר $U(s) == U'(s)$ כלומר אין הבדל הין הערכים באיטרציה הנוכחית לבין האיטרציה הקודמת - כלומר התכנסנו למדיניות אופטימלית.

שאלה 2

1. <u>המצבים:</u>

$S = \{s_0, s_1,..., s_n\} \cup \{T\}$ כאשר s_i המצב בו לסוחט יש i שקלים.

<u>הפעולות:</u>

בכל מצב חוץ מ-s_n ו-T, ישנן 2 פעולות: "לסחוט" או "לפרוש", כלומר

$A(s_i) = \{"QUIT", "EXTORT"\}\ for\ i = 0, 1,... n - 1$

הסתברויות המעבר:

$P(s_{i+1} \mid s_i, \text{"}EXTORT\text{"}) = p$ - אם הסוחט מחליט לסחוט והקורבן מסכים בהסתברות p , עוברים למצב s_{i+1}.

$P(T \mid s, \text{"}EXTORT\text{"}) = 1 - p$ - אם הסוחט מחליט לסחוט והקורבן מדווח למשטרה בהסתברות $1 - p$, עוברים למצב T.

$P(T \mid s_i, \text{"}QUIT\text{"}) = 1$ - אם הסוחט פורש, עוברים למצב T בהסתברות של 1.

תגמולים:

$R(s_i, \text{"}EXTORT\text{"}, s_{i+1}) = 0$ כאשר הסוחט מחליט לסחוט והקורבן מסכים, עוברים למצב s_{i+1} והסוחט מקבל שקל, נתחשב בזה בתגמול במעבר למצב T.

$R(s_i, \text{"}EXTORT\text{"}, T) = 0$ כאשר הסוחט מחליט לסחוט והקורבן משווח למשטרה, עוברים למצב T וכל הכסף נאבד - לכן התגמול הוא 0.

$R(s_i, \text{"}QUIT\text{"}, T) = i$ כאשר הסוחט פורש עוברים למצב T והוא מקבל את כל הכסף שסחט עד כה, ולכן התגמול הוא i.

2. לא. כיוון שבלי i מצבים המתארים כמה כסף סחט הסוחט בכל שלב במשחק, אין דרך לשמור כמה כסף הוא צבר.
3. כן. הכל יהיה זהה לסעיף 1 חוץ מהתגמולים:

$R(s_i, \text{"}EXTORT\text{"}, s_{i+1}) = 1$ כאשר הסוחט מחליט לסחוט והקורבן מסכים, עוברים למצב s_{i+1} והסוחט מקבל שקל.

$R(s_i, \text{"}EXTORT\text{"}, T) = -i$ כאשר הסוחט מחליט לסחוט והקורבן מדווח למשטרה, עוברים למצב T וכל הכסף נאבד - ובגלל שהיה במצב s_i היו לו i שקלים ולכן התגמול יהיה $i -$.

$R(s_i, \text{"}QUIT\text{"}, T) = 0$ כאשר הסוחט פורש עוברים למצב T והוא מקבל את כל הכסף שסחט עד כה, ולכן התגמול הוא כל הכסף שצבר המעברים הקודמים.

.4

a=0.5

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

$0 < p < 0.5$	$\pi_1(0) =$ לסחוט $\pi_1(1) =$ לפרוש $\pi_1(2) =$ לסחוט/לפרוש $\pi_1(3) =$ לפרוש	$U^{\pi_1(0)} = p$
$0.5 < p < \frac{2}{3}$	$\pi_2(0) =$ לסחוט $\pi_2(1) =$ לסחוט $\pi_2(2) =$ לפרוש $\pi_2(3) =$ לפרוש	$U^{\pi_2(0)} = 2p^2$
$\frac{2}{3} < p < 1$	$\pi_3(0) =$ לסחוט $\pi_3(1) =$ לסחוט $\pi_3(2) =$ לסחוט $\pi_3(3) =$ לפרוש	$U^{\pi_3(0)} = 3p^3$

נחשב:

<u>עבור מצב 3:</u>

תמיד נפרוש לפי הגדרת השאלה.

<u>עבור מצב 2:</u>

סחיטה: $p * (0 + 1 * 3) = 3p$

פרישה: 2

לכן עבור המקרים בהם $2 < 3p$ נפרוש אחרת נסחוט - כלומר לכל $1 > p > \frac{2}{3}$ נסחוט אחרת נפרוש.

<u>עבור מצב 1:</u>

סחיטה: $p * (0 + 1 * max(3p, 2)) = p * max(3p, 2)$

פרישה: 1

לכן עבור המקרים בהם $3p > 2$ כלומר $\frac{2}{3} < p < 1$ מתקיים: סחיטה תהיה $3p^2$

לכן עבור המקרים בהם $3p^2 > 1$ נפרוש אחרת נסחוט - כלומר לכל $\frac{2}{3} < p < 1$ נסחוט אחרת נפרוש.

אם $3p < 2$ כלומר $0 < p < \frac{2}{3}$ מתקיים: סחיטה תהיה $2p$

לכן עבור המקרים בהם $2p > 1$ נפרוש אחרת נסחוט - כלומר לכל $\frac{1}{2} < p$ נסחוט אחרת נפרוש.

וסה"כ נקבל עבור מצב 1 שלכל $0.5 < p < 1$ נסחוט אחרת נפרוש.

<u>עבור מצב 0:</u>

סחיטה:

$$p * (0 + 1 * max(p * (0 + 1 * max(3p, 2), 1)) = p * max(p * (0 + 1 * max(3p, 2), 1)$$

פרישה: 0

לכל p מתקיים $p * max(p * (0 + 1 * max(3p, 2), 1) > 0$ ולכן נסחוט לכל p.

נחשב את התועלת לפי כל מדיניות:

$$U^{\pi_1(0)} = (1 - p) * 0 + p * (1 * 1) = p$$

$$U^{\pi_2(0)} = (1 - p) * 0 + p * ((1 - p) * 0 + p * (1 * 2)) = 2p^2$$

$$U^{\pi_3(0)} = (1 - p) * 0 + p * ((1 - p) * 0 + p * ((1 - p) * 0 + p * (1 * 3))) = 3p^3$$

על מנת ש- π_1 תהיה אופטימלית נדרוש:

$$p > 2p^2 \text{ וגם } p > 3p^3 \Rightarrow 0 < p < 0.5$$

על מנת ש- π_2 תהיה אופטימלית נדרוש:

$2p^2 > p$ וגם $2p^2 > 3p^3 \Rightarrow 0.5 < p < \frac{2}{3}$

על מנת ש- π_3 תהיה אופטימלית נדרוש:

$3p^3 > p$ וגם $3p^3 > 2p^2 \Rightarrow \frac{2}{3} < p < 1$

חלק ב - בוצע

חלק ג -

<u>רטוב -</u> בוצע

<u>יבש -</u>

- המדיניות זהות וכולן זהות למדיניות האידיאלית.

```
10 episodes policy:
| RIGHT | RIGHT | RIGHT | None  |
| UP    | None  | UP    | None  |
| UP    | RIGHT | UP    | LEFT  |


100 episodes policy:
| RIGHT | RIGHT | RIGHT | None  |
| UP    | None  | UP    | None  |
| UP    | RIGHT | UP    | LEFT  |


1000 episodes policy:
| RIGHT | RIGHT | RIGHT | None  |
| UP    | None  | UP    | None  |
| UP    | RIGHT | UP    | LEFT  |
```

יתרונות של הגדלת מספר ה-episodes:

1. התכנסות טובה יותר -עם יותר episodes לאלגוריתם יש יותר הזדמנויות להתבונן וללמוד מהסביבה, מה שמוביל להערכות מדויקות יותר של ההסתברויות לפונקציית המעברים. זה גורר גם שאיכות המדיניות תהיה טובה יותר כי לסוכן יש נתונים מהימנים יותר לקבל החלטות על סמכם.
2. עם מספר גדול של episodes גם יש הזמנגנות לעבור בכל המצבים ובכך לגלות את התמלוגים של כל שלב.

חסרונות של הגדלת מספר ה-episodes:

1. צריך יותר זמן ומשאבים לחיושב כדי להריץ את האלגוריתם על מספר גדול יותר של episodes.
2. בזבזני - לפעמים בהרצת מספר גדול מאוד של episodes השוני של פונקציית המעברים והתמלוגים כבר לא תשתנה ואם נמשיך להריץ ל-episodes נוספים אחרי שלב זה - הרי שאנחנו סתם מריצים והתוצאה לא תשתנה באופן משמעותי.

```
function AnytimeADP(simulator, num_rows, num_cols, actions,
num_episodes):

    Initialize reward_matrix, value_function

    start_time = current time

while current time - start_time < max_time:

    for count in range(1, ∞):

        reward_matrix, value_function = RunADPLearner(simulator,
num_rows, num_cols, actions, count)


    return reward_matrix, value_function
```

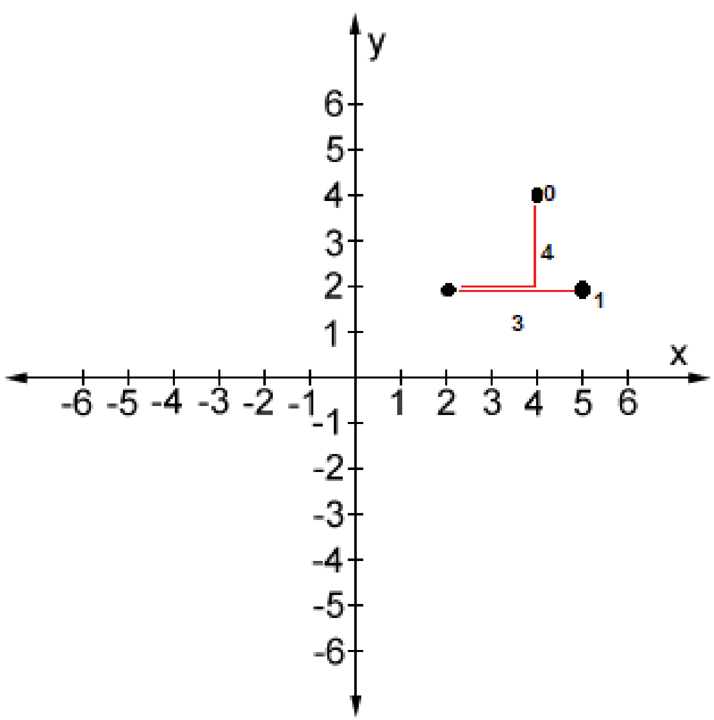
האלגוריתם אכן עומד בעיקרון אלגוריתמי anytime שמצאנו שכן ככל שמריצים את האלגוריתם עם num_episodes=count גדול יותר כך הפתרון המתקבל מדויק יותר, ותמיד נחזיר את הפתרון האחרון (והמדויק ביותר) שקיבלנו לפני שהזמן יסתיים.

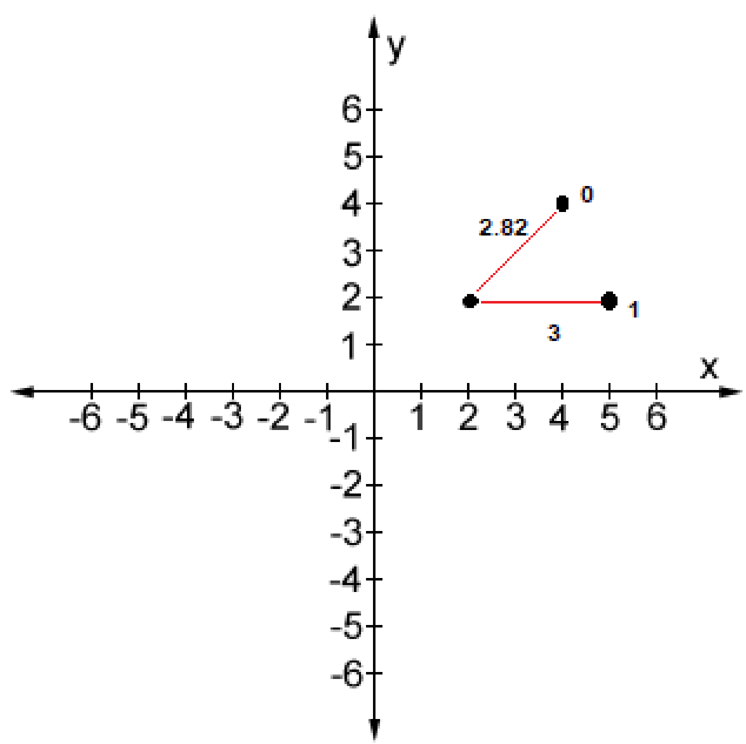
חלק ב

שאלה 1

א)

1) עבור d=1 נקבל כי מרחק מנהאטן ומרחק אוקלידי מחזירים אותו ערך (כי מדובר על מרחב חד מימדי) ולכן נקבל כי לכל K אין תלות בבחירת הפונקציה.
2) ניתן לראות כי עבור d=2,k=1 אנחנו מקבלים 0 עבור מרחק אוקלידי (גרף ימני) ו1 כאשר נשתמש במרחק מנהטן (גרף שמאלי).



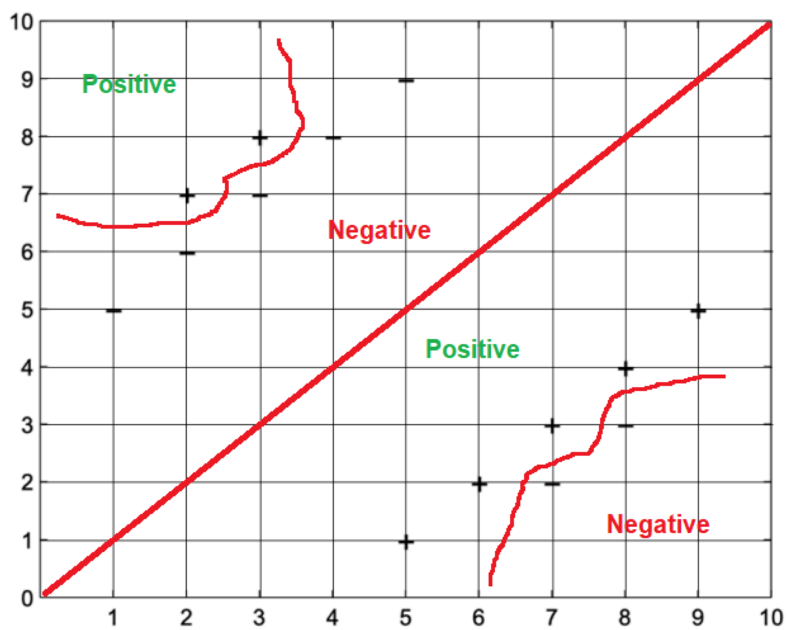


3) נשים לב כי בכל צד יש לנו 2 דוגמאות רועשות ולכן נרצה לבחור k=5 על מנת שה2 דוגמאות האלו לא יקבלו השפעה מירבית, במקום נסווג לפי ה3 האחרים שהם מסווגים נכון. והדיוק יהיה $\frac{10}{14}$.

4) עבור לקבל מסווג רוב נבחר k=14 ובכך ניתן סיווג לפי הרוב מכיוון שיש 14 איברים, נשים לב כי המרחק לא ישפיע כאן, נקבל החלטה לפי הרוב.

5) עבור ערכי K קטנים מדי אנחנו עלולים ללמוד מדוגמאות שהם רעש ולכן לקבל סיווג לא נכון. עבור ערכי K גדולים מדי אנחנו נסתכל על יותר מדי דוגמאות ולכן אם יש לנו יותר סיווגים על 0 מ1 אנחנו נסווג את הדוגמא החדשה ב0 רק בגלל שיש יותר כאלה ולאו דווקא בגלל שזה הסיווג הנכון.

6) עבור k=1 נקבל את הגרף הבא:



מתפצלים ונהנים:

לא נכון, יהא ϵ, עם כלל ϵ החלטה ועץ T. ד"ז גזום עם כלל החלטה. נבחר את הקבוצת אימון הבאה $\{98, 99, 101\}$ המסווגת מספרים מעל 100 כTRUE ומתחת ל100 כFALSE עבור איבר חדש $x = 100 + 2\epsilon$ מתקיים $|x - \epsilon| < \epsilon$ ולכן T יסווג את x כTRUE, כי x>100 עם זאת T" הגזום עם כלל החלטה רגיל יסווג אותו כFALSE כיוון שיסווג על פי הרוב שהם $\{98, 99\}$ ומסווגים כFALSE.

חלק רטוב

הדיוק שקיבלנו:

```
PS C:\Users\Roey\OneDrive\שולחן העבודה\AI\HW3\ID3> python ID3_experiments.py
>>
Test Accuracy: 94.69%
PS C:\Users\Roey\OneDrive\שולחן העבודה\AI\HW3\ID3>
```