מבוא לבינה מלאכותית
236501

<u>דו"ח הגשה – תרגיל בית רטוב 3</u>

מתן צחורי 208936989

אלון פנפיל 318598166

חלק א' – MDP ו-RL

חלק א' – חלק יבש

שאלה 1:

א.

$$U^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \cdot R(s_t, a_t, s_{t+1}) | s_0 = s\right]$$

ב.

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

ג.

function VALUE-ITERATION(mdp, ϵ) **returns** a utility function
- **inputs:** $\boldsymbol{mdp}$ - an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$, rewards $R(s, a, s')$, discount γ
  - $\boldsymbol{\epsilon}$ - the maximum error allowed in the utility of any state
- **local variables:** $\boldsymbol{U}$, $\boldsymbol{U'}$ - vectors of utilities for states in S, initially zero
  - $\boldsymbol{\delta}$ — the maximum change in the utility of any state in an iteration
- **Repeat**
  - $U \leftarrow U'; \delta \leftarrow 0$
  - **for each** state s **in** S **do**
    - $U'^{(s)} \leftarrow \underset{a \in \mathbb{A}(s)}{max} \sum_{s'} P(s'|a,s)(R(s,a,s') + \gamma \cdot U(s'))$
    - **if** $|U'(s) - U(s)| > \delta$ **then** $\delta \leftarrow |U'(s) - U(s)|$
- **until** $\delta < \epsilon(1-\gamma)/\gamma$
- **return** U

ד.

function POLICY-ITERATION(mdp) **returns** a policy

inputs: $\boldsymbol{mdp}$ **-** an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$, rewards $R(s)$, discount γ

local variables: $\boldsymbol{U}$ − a vector of utilities for states in S, initially zero

$\boldsymbol{\pi}$ − a policy vector indexed by state, initially random

Repeat

$$U \leftarrow \sum_{s'} P(s'|\pi(s), s)[R(s, \pi(s), s') + \gamma U^{\pi}(s')]$$

$unchanged? \leftarrow true$

for each state s **in** S **do**

curr_eval $= \sum_{s'} P(s'|\pi(s), s)[R(s, \pi(s), s') + \gamma U(s')]$

if $\max_a \sum_{s'} P(s'|a, s)[R(s, a, s') + \gamma U(s')] > curr\_eval$ **then**

$\pi(s) \leftarrow \underset{a}{argmax} \sum_{s'} P(s'|a, s)[R(s, a, s') + \gamma U(s')]$

$unchanged? \leftarrow true$

until $unchanged?$

return π

בסעיפים ג' ו-ד', כאשר $\gamma = 1$, אנחנו כבר לא נמצאים במצב של $Discounted\ Rewards$, והתועלת עלולה להיות אינסופית. במצב זה, לא ניתן להבדיל בין איכות של מדיניות.
על מנת לפתור זאת, יש לדרוש שבסביבה יהיה קיים מצב סופי, ויש לדרוש על ה-mdp שהמדיניות תבטיח שהסוכן יגיע למצב סופי.

שאלה 2:

1. נגדיר את הבעיה כבעיית mdp באופן הבא:

נגדיר את קבוצת המצבים:

$$S = \{1,2, \dots, n\} \cup \{T\}$$

כאשר $s_{init} = 0$, $s_{final} = T$.

נגדיר את הפעולות לכל מצב:

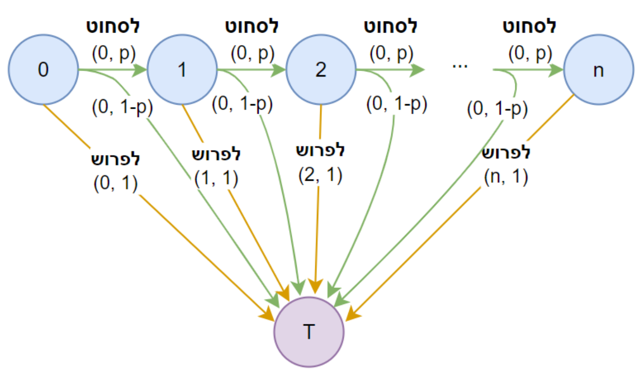
$$\forall s \in (1, \dots, n-1): A(s) = \{\text{"לסחוט"}, \text{"לפרוש"}\}, A(n) = \{\text{"לפרוש"}\}, A(s_{final}) = \{\emptyset\}$$

נגדיר את מודל המעבר:

$$\forall s \in (1, \dots, n-1): P(s+1 \mid \text{"לסחוט"}, s) = p,\ \ P(T \mid \text{"לסחוט"}, s) = 1-p$$
$$\forall s \in (1, \dots, n): P(T \mid \text{"לפרוש"}, s) = 1$$

נגדיר את פונקציית התגמולים:

$$\forall s \in (1, \dots, n-1): R(s, \text{"טוחסל"}, s+1) = 0,\ \ R(s, \text{"טוחסל"}, T) = 0$$
$$\forall s \in (1, \dots, n): R(s, \text{"לפרוש"}, T) = s$$



נציג את ה-mdp בתרשים מצבים:
כאשר קשתות ירוקות הן פעולת סחיטה,
קשתות כתומות הן פעולת פרישה, והצמד
על הקשת הוא:
(<תגמול פעולה>,<הסתברות>)

2. לא ניתן לנסח את הבעיה עם מצב יחיד ומצב סיום, מכיוון שהתגמול משתנה בהתאם למספר הפעמים שהסוחט הצליח לסחוט והקורבן לא קרא למשטרה.

3. כן, היינו יכולים לנסח את הבעיה בצורה שונה בה התגמול על סחיטה מוצלחת הוא 1 במקום 0, והתגמול על סחיטה לא מוצלחת הוא $-s$ בהינתן שנמצאים במצב s במקום 0. התגמול על פרישה הוא 0.

4. בהינתן $n = 3$, לפי פונקציית העדכון של בלמן מתקיים:

$$U(T) = 0$$

$$U(0) = \max \begin{cases} 1 \cdot (0 + 1 \cdot U(T)) = 0 & \text{"לפרוש",} \\ p \cdot (0 + 1 \cdot U(1)) + (1-p) \cdot (0 + 1 \cdot U(T)) = p \cdot U(1) & \text{"לסחוט",} \end{cases}$$

$$U(1) = \max \begin{cases} 1 \cdot \big(1 + 1 \cdot U(T)\big) = 1 & \text{,"לפרוש"} \\ p \cdot \big(0 + 1 \cdot U(2)\big) + (1-p) \cdot \big(0 + 1 \cdot U(T)\big) = p \cdot U(2) & \text{,"לסחוט"} \end{cases}$$

$$U(2) = \max \begin{cases} 1 \cdot \big(2 + 1 \cdot U(T)\big) = 2 & \text{,"לפרוש"} \\ p \cdot \big(0 + 1 \cdot U(3)\big) + (1-p) \cdot \big(0 + 1 \cdot U(T)\big) = p \cdot U(3) & \text{,"לסחוט"} \end{cases}$$

$$U(3) = 1 \cdot \big(3 + 1 \cdot U(T)\big) = 3$$

נציב את $U(3)$ ל-$U(2)$, ונקבל שנעדיף לבחור "לסחוט" כאשר $p > \frac{2}{3}$.
עבור מקרה זה, נקבל שגם ב-$U(0), U(1)$ נעדיף "לסחוט".

במקרה ש- $p < \frac{2}{3}$:
נקבל שנעדיף לבחור "לסחוט" במצב 1 (ולפרוש ב-2) כאשר $p > \frac{1}{2}$.
במקרה ש- $\frac{1}{2} < p < \frac{2}{3}$:
נקבל שנעדיף "לסחוט" במצב 0.

במקרה ש- $p < \frac{1}{2}$:
נקבל שנעדיף "לפרוש" במצב 1. במצב 0 נעדיף "לסחוט", ובמצב 2, 3 לא משנה מה נעשה כי לא נגיע לשם.

$$a = \frac{1}{2}\,, \qquad b = \frac{2}{3}$$

תועלת	מדיניות	ערכי p
$V^{\pi_1}(0) = p$	$\pi_1(0) =$ "לסחוט" $\pi_1(1) =$ "לפרוש" $\pi_1(2) =$ "לסחוט"/"לפרוש" $\pi_1(3) =$ "לפרוש"	$0 < p < a$
$V^{\pi_2}(0) = 2p^2$	$\pi_2(0) =$ "לסחוט" $\pi_2(1) =$ "לסחוט" $\pi_2(2) =$ "לפרוש" $\pi_2(3) =$ "לפרוש"	$a < p < b$
$V^{\pi_3}(0) = 3p^3$	$\pi_3(0) =$ "לסחוט" $\pi_3(1) =$ "לסחוט" $\pi_3(2) =$ "לסחוט" $\pi_3(3) =$ "לפרוש"	$b < p < 1$

חלק ג' – רטוב

שאלה 1:

לאחר הרצת האלגוריתם עם 10, 100 ו-1000 אפיזודות, קיבלנו את התוצאות הבאות:

עבור 10 אפיזודות –

```
Final utility:                            Final policy:
| 0.539 | 0.670 | 0.808 | 1.0   |         | RIGHT | RIGHT | RIGHT | None |
| 0.422 | None  | 0.417 | -1.0  |         | UP    | None  | UP    | None |
| 0.305 | 0.218 | 0.277 | 0.074 |         | UP    | LEFT  | UP    | LEFT |
```

עבור 100 אפיזודות –

```
Final utility:                            Final policy:
| 0.523 | 0.660 | 0.802 | 1.0   |         | RIGHT | RIGHT | RIGHT | None |
| 0.411 | None  | 0.492 | -1.0  |         | UP    | None  | UP    | None |
| 0.308 | 0.264 | 0.353 | 0.149 |         | UP    | RIGHT | UP    | LEFT |
```

עבור 1000 אפיזודות –

```
Final utility:                            Final policy:
| 0.507 | 0.648 | 0.794 | 1.0   |         | RIGHT | RIGHT | RIGHT | None |
| 0.397 | None  | 0.489 | -1.0  |         | UP    | None  | UP    | None |
| 0.296 | 0.258 | 0.349 | 0.139 |         | UP    | RIGHT | UP    | LEFT |
```

ניתן לראות שלא כל המדיניות שהתקבלו הן זהות. המדיניות עבור 10 אפיזודות שונה מהמדיניות האופטימלית, ולעומת זאת, המדיניות שהתקבלה עבור 100 ו-1000 אפיזודות היא המדיניות האופטימלית.
ניתן לראות שככל שנבצע יותר אפיזודות, כך נקרב טוב יותר את הסתברויות המעבר בין המצבים, וכך נגיע לחישוב מדיניות זהה לחישוב עם ההסתברויות המקוריות. עם זאת, הרצת יותר אפיזודות יגזול יותר זמן חישוב ומשאבים.

שאלה 2:

נציע את האלגוריתם הבא כגרסת $Anytime$ של אלגוריתם ה-Adp:

```
function ADP_Anytime(sim: Simulator, n_rows: int, n_cols: int, actions: List[Action], time_limit: float):
    num_episodes = 10

    while not exceed time_limit:
        rewards_matrix, transition_probs = adp_algorithm(sim, num_episodes, n_rows, n_cols, actions)
        num_episodes += 1

    return rewards_matrix, transition_probs
```

האלגוריתם שהצענו מפעיל את אלגוריתם Adp שמימשנו בכל פעם עם כמות אפיזודות הולכת וגדלה עד שנגמרת הגבלת הזמן. ככל שיש לאלגוריתם יותר זמן לרוץ, כך הוא מגיע למספר אפיזודות גדול יותר ובכך הוא משתפר ככל שניתן בזמן המוגבל שלו.

חלק ב' – מבוא ללמידה

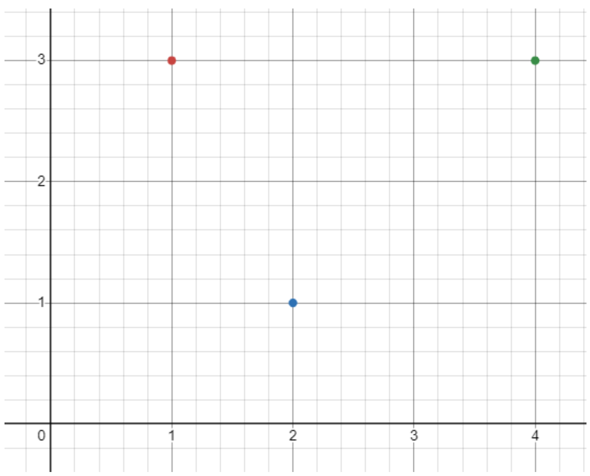
חלק א' – חלק יבש

KNN -

א. עבור פונקציות המרחק הנתונות – מרחק אוקלידי ומרחק מנהטן:

1) כאשר $d = 1$, נקבל שחישוב המרחק לפי כל אחת מהפונקציות הנ"ל הוא זהה, ולכן לכל k במקרה זה נקבל את אותו החזאי.
כאשר $d > 1$, פונקציות המרחק שונות זו מזו, ולא מובטח לנו ל-k כלשהו שנקבל חזאי זהה.

2) למשל עבור הדוגמה הבאה, נקבל שבחירת פונקציית המרחק תשנה את סיווג דגימת המבחן: נניח $k = 1$, $d = 2$, ונתון המדגם הבא כאשר נקודה כחולה מתויגת כחיובית ונקודה אדומה מתויגת כשלילית. הנק' הירוקה היא נק' המבחן:



עבור מרחק מנהטן, נקבל שהמרחק מדגימת המבחן לנק' הכחולה הוא 4, ולנק' האדומה הוא 3, ולכן נסווג את נקודת המבחן כשלילית.

עבור מרחק אוקלידי, נקבל שהמרחק מדגימת המבחן לנק' הכחולה הוא $\sqrt{8} = 2.82$, ולנק' האדומה הוא $\sqrt{9} = 3$, ולכן נסווג את נקודת המבחן כחיובית.

3) עבור קבוצת האימון הנתונה ופונקציית מרחק אוקלידי נקבל:
עבור $k = 1$, נסווג לא נכון את כל הדגימות פרט ל-$(5,9), (1,5), (9,5), (5,1)$ – $\frac{4}{14}$
עבור $k = 2$, נסווג לא נכון את כל הדגימות פרט ל-$(8,4), (9,5), (6,2), (5,1)$ - $\frac{4}{14}$
עבור $k = 3$, נכון את כל הדגימות פרט ל-$(3,8), (3,7), (2,7), (8,3), (7,3), (7,2)$ – $\frac{8}{14}$
עבור $k = 4$, נטעה על כל הנקודות בחלק השמאלי העליון, ולכן כבר נקבל חיזוי פחות טוב מ-3.
עבור $k = 5$, נסווג נכון על את שתי החמישיות למעלה ולמטה, ונטעה על הזוגות. לכן נקבל $\frac{10}{14}$.
עבור $k = 6$, לא ישתנה.
ל-k גדולים יותר לא נצליח לשפר, ולכן $k = 5$ ייתן דיוק מירבי.

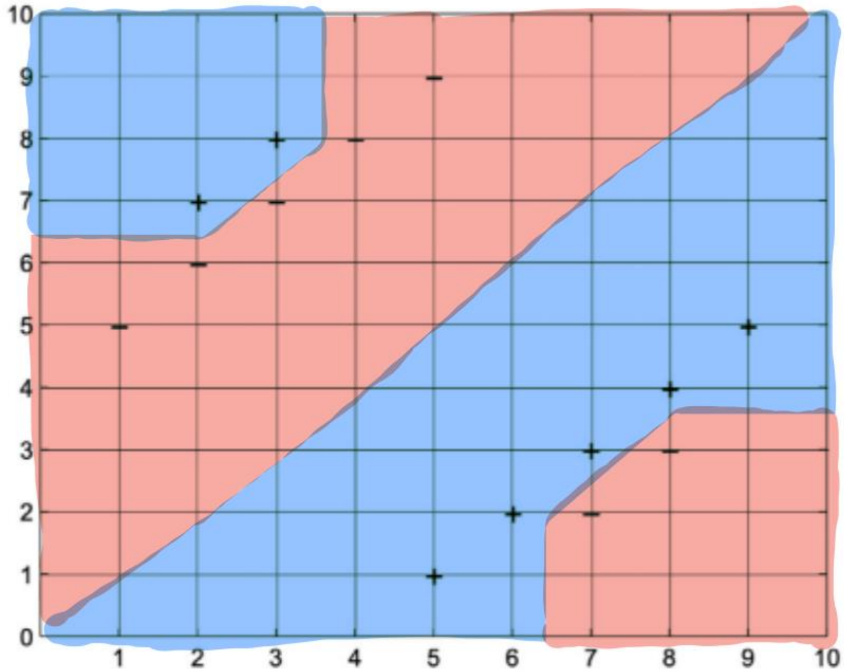
4) ל-k שעבורו נסתכל על לפחות חצי מקבוצת האימון ייתן לנו מסווג $majority$. במדגם הנתון יש 14 דגימות, ולכן עבור $k = 7$ ומעלה נקבל מסווג $majority$.

5) שימוש ב-k קטן מדי עלול לגרום לרגישות יתר לרעש ולדגימות יוצאות מן הכלל. עם k קטן, המודל נוטה לתת סיווג לפי מספר מצומצם של שכנים שעלולים להיות תוצאה של רעש או דגימות יוצאות דופן שלא מייצגות את ההתפלגות ממנה נדגם המדגם. בכך עלול המודל לעשות

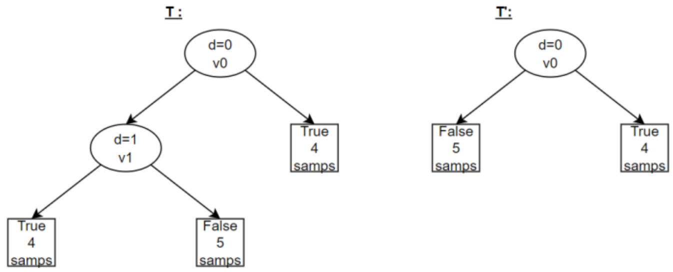
"$overfitting$" למדגם, ולתת ביצועים לא טובים על דגימות חדשות.

שימוש ב-k גדול מדי עלול ליצור תופעה הפוכה של "$underfitting$", כאשר המודל מסתכל על יותר מדי שכנים הוא עלול לבצע החלטות שמושפעות יותר מהטרנד הכללי של המדגם ונתקרב יותר למסווג "$majority$" שמוטה לטובת התיוג הנפוץ יותר, במקום להכליל.

(6



מתפצלים ונהנים –



הטענה לא נכונה, נראה זאת בעזרת דוגמה נגדית.

למשל עבור העצים T ו-T' המתקבל מ-T הבאים:

לכל אפסילון שנבחר ϵ' נוכל לקחת דגימה:

$$x = \left(v_0 + \frac{\epsilon_0'}{2}, v_1 + \epsilon_1' + 1\right)$$

בעץ T בשורש מתקיים $\left|v_0 + \frac{\epsilon_0'}{2} - v_0\right| < \epsilon_0'$, ולכן לפי כלל ההחלטה אפסילון נבדוק את שני הבנים. בצומת הבא מתקיים $|v_1 + \epsilon_1' + 1 - v_1| > \epsilon_1'$, ולכן נבחר בענף אחד בלבד. מכיוון וערך הפיצ'ר גדול מ-v_1, נבחר בענף ימין. בסה"כ נסווג $False$, כיוון וזה הסיווג הנפוץ ביותר בכל העלים שהגענו אליהם.

בעץ T', בשורש מתקיים שהפיצ'ר 0 גדול מ-v_0 ולכן לפי כלל ההחלטה הרגיל נבחר בענף ימין, ונסווג $True$ לפי העלה אליו הגענו. קיבלנו סיווגים שונים בשני העצים, ולכן הטענה לא נכונה לכל דגימה $x \in \mathbb{R}^d$.

חלק ג' – חלק רטוב ID3

1. ממומש בקובץ $utils.py$
2. א. ממומש בקבצים $ID3.py$ ו-$ID3\_experiments.py$.
   ב. הדיוק שהתקבל בהרצת הניסוי הוא: 94.69%