תרגיל בית 3 – MDP

**יגל מימון 316611078**

**יורם פרטוש 209300708**

**חלק א׳ – MDP (44 נק׳)**

**רקע**

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

**חלק א׳ - חלק היבש** [Writing Hand on Apple ](https://emojipedia.org/apple/ios-14.6/writing-hand/)

1. בתרגול ראינו את משוואת בלמן כאשר התגמול ניתן עבור המצב הנוכחי בלבד, כלומר , 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: , למתן תגמול זה נקרא "תגמול תוצאתי". לצורך שלמות ההגדרה, נגדיר שאם לכל   
 מתקיים - אז .

1. (1 נק') התאימו את הנוסחה של התוחלת של התועלת מהתרגול, עבור התוחלת של התועלת המתקבלת במקרה של ״ תגמול תוצאתי ״, אין צורך לנמק.
2. (1 נק') כתבו מחדש את נוסחת משוואת בלמן עבור המקרה של ״ תגמול תוצאתי ״, אין צורך לנמק.

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

1. (2 נק') נסחו את אלגוריתם Value Iteration עבור המקרה של ״ תגמול תוצאתי ״.

While

return U

אם גמא הוא 1 אין דעיכה ונשנה את תנאי העצירה. כיוון שגמא חיובי ו- האלגוריתם לא יעצור לעולם לכן התנאי יהיה .

התנאי למציאת מדיניות אופטימלית יהיה שמרחב המצבים ופעולות סופי בלי מעגל עם תועלת חיובית אחרת ווקטור התועלות ישאף לאינסוף.

1. (2 נק') נסחו את אלגוריתם Policy Iteration עבור המקרה של ״ תגמול תוצאתי ״.

תנאי התכנסות: מרחב המצבים והפעולות צריך להיות סופי, אם יהיה מעגל בעל סכום rewards חיובי – PolicyEvaluation תחזיר וקטור תועלות שמתבדר לאינסוף.

נתון הגרף הבא:



נתונים:

* (Discount factor) .
* אופק אינסופי.
* – קבוצת המצבים – מתארים את מיקום הסוכן בגרף.
* – קבוצת המצבים הסופיים.
* קבוצת הפעולות לכל מצב (על פי הגרף), לדוגמא: .
* תגמולים ("תגמול תוצאתי"):
* מודל המעבר הוא דטרמיניסטי, כלומר כל פעולה מצליחה בהסתברות אחת.

1. (יבש 2 נק') הרץ את האלגוריתם Value iteration שכתבת על הגרף הנתון. ומלא את הערכים בטבלה הבאה, כאשר . (ייתכן שלא צריך למלא את כולה).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 5 | 5 | 5 | 5 | 0 |  |
|  |  |  |  | 7 | 7 | 7 | 7 | 0 |  |
|  |  |  |  | 1.5 | 1.5 | 1.5 | -1 | 0 |  |
|  |  |  |  | 2.5 | 2.5 | 2.5 | -1 | 0 |  |
|  |  |  |  | 2.5 | 2.5 | 2.5 | -1 | 0 |  |
|  |  |  |  | 0.25 | 0.25 | -1.5 | -1 | 0 |  |
|  |  |  |  | 0.25 | 0.25 | -1.5 | -1 | 0 |  |

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

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

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | ↖ | ↖ |  |  |  |

1. (יבש 2 נק') חיזרי על הסעיף הקודם. הפעם עם **אופק סופי כאשר** (שימי לב, המדיניות לא חייבת להסתיים במצב מסיים, ישנם מצבים שלא יכולים להגיע למצב מסיים עם אופק זה. ישנם צמתים עם מספר תשובות נכונות, נקבל את כולם).

יש אופק סופי אז המדיניות לא קבועה ובכל צעד נמקסם את התועלת כתלות בזמן:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **↖** |  |

(כאשר יש מספר חיצים, החץ ה-I הוא הצעד ה-i)

1. (1 נק') ללא תלות בשינוי של הסעיף הקודם. אם , מה מספר המדיניות האופטימליות הקיימות? נמקו.

מספר המדיניות הקיימות הוא 48, נחשב:

בכל המצבים חוץ מ-{s1, s2} לכל מצב שיתקדמו אליו נקבל ש-R=-1 וערך התועלת של מצב היעד מוכפל ב-0 אז הוא חסר השפעה. במצבים s1, s2 נעדיף ללכת למצב הסופי כי שם התגמול גדול יותר. נספור כמה אופציות שקולות יש לכל מצב:

עבור: s1, s2: אפשרות אחת.

עבור s3, s4, s5, s6: 2 אפשרויות

עבור s7: 3 אפשרויות.

נסכום ונקבל:

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

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

כעת נתוגמל ב-2 עבור מעבר מ-1 ל-2 או ההפך, ומכיוון שמקדם הדעיכה הוא 1 אז אחרי איטרציה אחת הצעד מ1 יהיה ל2 ולא למצב הסופי, ואחרי האיטרציה השנייה הצעד יהיה מ2 ל1 וחוזר חלילה. אז ערכי התועלת שלהם יגדלו עוד ועוד (גם של שאר הצמתים) אבל לא נגיע למצב הסופי וניתקע בין 1 ל2 לפי המדיניות הזו.

נספח MDP:

דוגמת הרצה (שימו לב שהרצה זו השתמשה במודל הסתברותי שונה משלכם)

יצירת הסביבה:

mdp = MDP(board=board\_env,  
 terminal\_states=terminal\_states\_env,  
 transition\_function=transition\_function\_env,  
 gamma=1.0)

הדפסת הלוח עם התגמולים לכל מצב:

print('@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@')  
print("@@@@@@ The board and rewards @@@@@@")  
print('@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@')  
mdp.print\_rewards()

פלט:

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

Value iteration:

print('@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@')  
print("@@@@@@@@@ Value iteration @@@@@@@@@")  
print('@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@')  
  
U = [[0, 0, 0, 0],  
 [0, 0, 0, 0],  
 [0, 0, 0, 0]]  
print("\nInitial utility:")  
mdp.print\_utility(U)  
print("\nFinal utility:")  
U\_new = value\_iteration(mdp, U)  
mdp.print\_utility(U\_new)  
print("\nFinal policy:")  
policy = get\_policy(mdp, U\_new)  
mdp.print\_policy(policy)

פלט:



Policy iteration:

print('@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@')  
print("@@@@@@@@@ Policy iteration @@@@@@@@")  
print('@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@')  
  
print("\nPolicy evaluation:")  
U\_eval = policy\_evaluation(mdp, policy)  
mdp.print\_utility(U\_eval)  
  
policy = [['UP', 'UP', 'UP', 0],  
 ['UP', 'WALL', 'UP', 0],  
 ['UP', 'UP', 'UP', 'UP']]  
print("\nInitial policy:")  
mdp.print\_policy(policy)  
print("\nFinal policy:")  
policy\_new = policy\_iteration(mdp, policy)  
mdp.print\_policy(policy\_new)  
  
print("Done!")

פלט:

