דו"ח פרויקט

Reinforcement learning of mini-poker agents

by Itay Guy (305104184) and Dean Sharaby (311138747)

השוואה סטטיסטית בין ביצועי הסוכן החכם אל מול הסוכן האקראי (כאשר סכום התחלתי הוא 10)

a. הסוכן החכם מצליח לבלף את יריבו הסוכן האקראי ב-7.06% מסך הסיבובים (+10000)

הגדרת הבלוף: (reward-penalty)^2 < BLUFF_UNDERBOUND
המיושמת כך בקוד:

```
def updateIfBluff(self,state):
    if self._states.getFinalState() == States.WIN and self.action == Poker.ALLIN:
        idx = self._states.getLinearIndex(state)
        qtable = self._states.getQTable()
        stateRank = self._states.getExpectedAction(state)
        stateRank = stateRank[1]
        if stateRank < States.POOR_PERCENTILE*self._states.getTotalPercentile():
            self.bluffs += 1
```

b. כמות הסיבובים הממוצע עד שהסוכן החכם מנצח היא 11
c. ממוצע העונשים של הסוכן החכם לאורך תקופת הלימוד היא -2.915

```
AI has finished
Summary results:
-------------------------------
Player 0 (QAgent) games won: 54.50000000000001%
Player 0 (QAgent) rounds won: 55.947497949138636%
Player 0 (QAgent) frequency rounds per win: 11.18348623853211
QAgent plotting learning process graph..
Player 0 (QAgent) penalties: -2.9159146841673502
Player 0 (QAgent) saved Qtable
Player 1 (RandomAgent) games won: 45.5%
Player 1 (RandomAgent) rounds won: 45.365053322395404%
Player 1 (RandomAgent) frequency rounds per win: 13.395604395604396
Player 1 (RandomAgent) penalties: 0.0
```

d. מידול המרחב

1. <u>כל מצב הוא וקטור של חמישייה מהצורה:</u>

- $[rank\_card1(0-12), rank\_card2(0-12), is\_same\_suit(0-1), pot\_amount(0-3), is\_big\_blind(0-1)]$
לפי מודל זה קיימים 1456 = 2*4*2*7*13 מצבים שונים

- כל מצב ממופה ל2 פעולות שניתן לבצע במשחק בכל תור $[allin, fold]$
כאשר כל איבר בווקטור זו מכיל את ערך ה-reward/penalty שהצטבר עד לצעד ה-t במשחק

2. כל reward/penalty שניתן לסוכן הוא כפונקציה של זוג הקלפים שבהם אחז, הפעולה שביצע ותוצאת הסיבוב.
למשל, אם אחז ב-[A,A] ,ביצע ALLIN וזכה, אז יקבל reward מסויים אחרת penalty מסויים (באופן דינאמי לפי כמות הכסף שברשותו)

3. כמות הכסף שהפסיד/הרוויח מתעדכנת בתוך ה-reward/penalty בתוך נוסחת העדכון
בנוסף לכך, אם הסוכן זכה במשחקון כולו יקבל reward/penalty נוסף שמזכה אותו בהטבה לאורך כל הסבבים במשחק מתוך מחשבה שצעדיו היו ככל הנראה נכונים עד כדי השגת הניצחון:

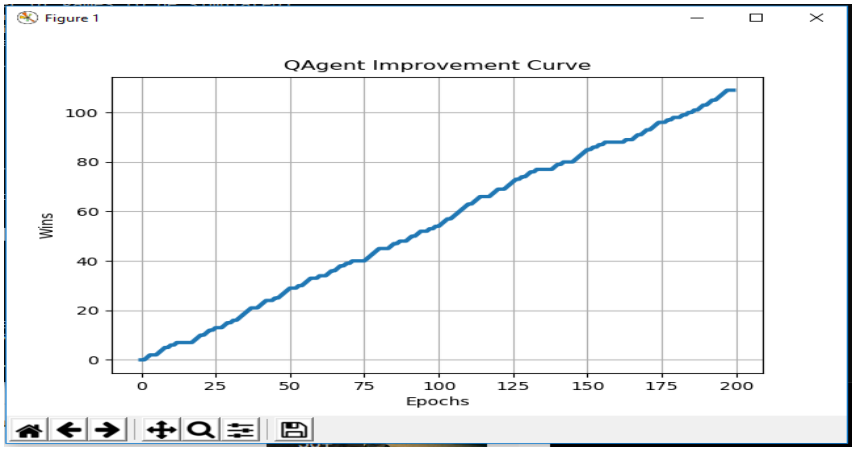
```
if Poker.INSTANCE.isGameover():
    logger.log(logger.DEBUG,"\n<<<GAME OVER>>>>")
    Poker.INSTANCE.printStatus()
    for idx,agent in enumerate(agents):
        # agent.normalize()
        agent.trackAgentGames()
        agent.backtrackUpdateStates()
        logger.log(logger.DEBUG,"Player {} ({}) total games won:
        logger.log(logger.DEBUG,"Player {} ({}) total rounds won:
```

4. הקלפים שמוגרלים לסוכן לאחר הלמידה – בזמן משחק ה-test – הינם מוגרלים מתוך מצבים שנתקלנו בהם בזמן הלמידה בלבד בכדי לייצר test אמין שמשקף את איכות למידת הסוכן

5. ערך המצב הבא ממודל ככמות הכסף המקסימאלית שיהיה ניתן להפסיד מול קלפים מתחרים אחרים (סימולציה פנימית)

c. גרף שיפור ביצועים של הסוכן החכם לאורך ניסיונות הלמידה

ניתן להבחין שישנה עליה לינארית יחסית חלקה בין תקופת הלימוד לבין כמות הנצחונות במשחקונים, כלומר ישנה הטיה של הסוכן החכם כלפי המצבים שבאמת כדאי לשחק בהם:



חלק עיקרי מהמדד של לימוד הסוכן הם ה-rewards/penalties שיש לו בטבלה על כל מצב ולהלן דוגמת הרצה של משחק עם ערכים אלו:

Folds Cases

```
all in / fold -5.004135047782363 -1.548
-----------------------------
|id|type|money|cards    |status   |
-----------------------------
|0 |SB  |9.5$ |[3♦, 7♣] | Fold    |
|1 |BB  |9.0$ |[4♥, 10♣]| Pending |
-----------------------------
Pot: 1.5%
Player 1(RandomAgent) wins
```

```
all in / fold -7.537650918808659 -1.66
-----------------------------
|id|type|money|cards   |status   |
-----------------------------
|0 |SB  |9.5$ |[4♥, 5♦]| Fold    |
|1 |BB  |9.0$ |[3♥, 7♦]| Pending |
-----------------------------
Pot: 1.5%
Player 1(RandomAgent) wins
```

```
all in / fold -9.037720733764692 -1.4505
-----------------------------
|id|type|money|cards   |status   |
-----------------------------
|0 |SB  |10.0$|[2♠, 8♣]| Fold    |
|1 |BB  |8.5$ |[7♠, J♣]| Pending |
-----------------------------
Pot: 1.5%
Player 1(RandomAgent) wins
```

All-In Cases

```
all in / fold 6.648374158238224 -1.457
----------------------------
|id|type|money|cards   |status  |
----------------------------
|0 |SB  |5.5$ |[7♦, 7♣]| All in |
|1 |BB  |0.0$ |[3♣, 6♥]| All in |
----------------------------
Pot: 14.5%
flop: [A♠, K♠, K♣, A♥, 7♥]
Player 0(QAgent) wins
```

```
all in / fold 9.786083450400076 -1.449
----------------------------
|id|type|money|cards    |status  |
----------------------------
|0 |SB  |6.5$ |[10♠, A♣]| All in |
|1 |BB  |6.0$ |[4♠, 8♣] | Fold   |
----------------------------
Pot: 7.5%
Player 0(QAgent) wins
```

```
all in / fold 13.047820039752274 -1.666
----------------------------
|id|type|money|cards   |status  |
----------------------------
|0 |SB  |0.5$ |[9♣, K♠]| All in |
|1 |BB  |0.0$ |[7♣, J♥]| All in |
----------------------------
Pot: 19.5%
flop: [7♥, K♣, A♦, 2♦, 2♣]
Player 0(QAgent) wins
```

פרמטרים של האלגוריתם

GAMA=0.1, EPSILON=0.1, ALPHA=0.2

- שינוי של epsilon גדל מ-0.1 ל-0.7 והשפיע לטובה על ממוצע הנצחונות של הסוכן החכם בזמן משחק הבדיקה (57.9% נצחונות אל מול 54.4%)

- שינוי של Gama מ-0.1 ל-0.7 גרר עליה של אחוז הנצחונות מ-54.4% ל-55.3% מכיוון שנתן משקל גבוה יותר לפונקציה בהמשך נוסחת העדכון שמכניסה מידע ל-reward/penalty כפונקציה של ערכי הפעולות הנוכחיים שיש בטבלה

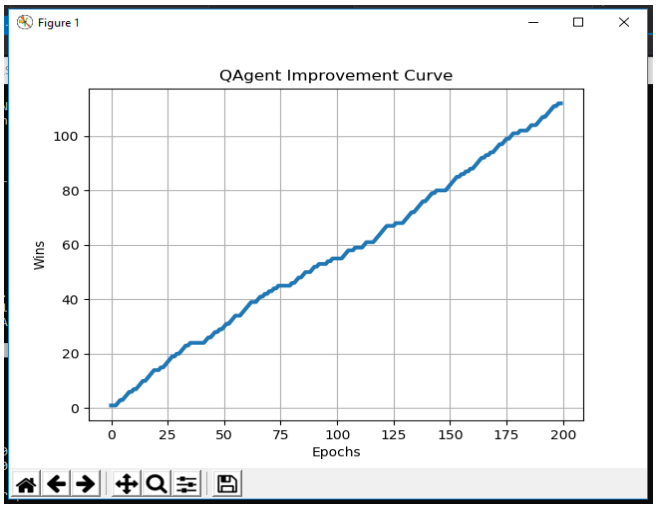
- שינוי של alpha מ-0.2 ל-0.7 גרר ירידה של אחוז ההצלחות עם עליית כמות האיטרציות

רגישות של התוצאות

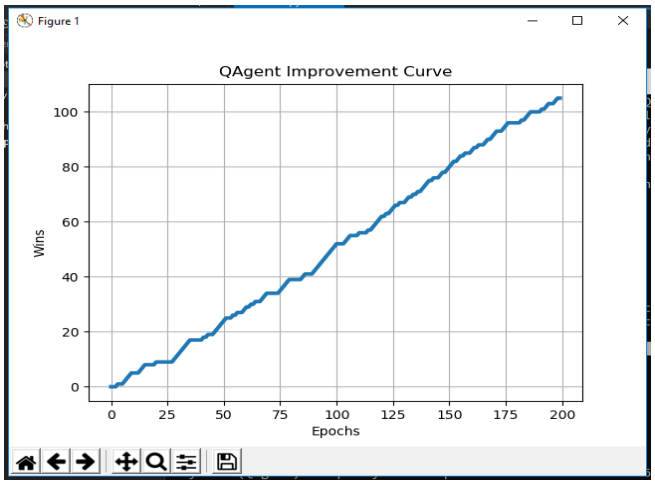
ניתן להבחין שתוצאות הלמידה והיישום תלויות באופן משמעותי בפרמטרים שהוצגו הנ"ל

ניתן לראות את הגרף של (על אותם כמות איטרציות):

Alpha = 0.2

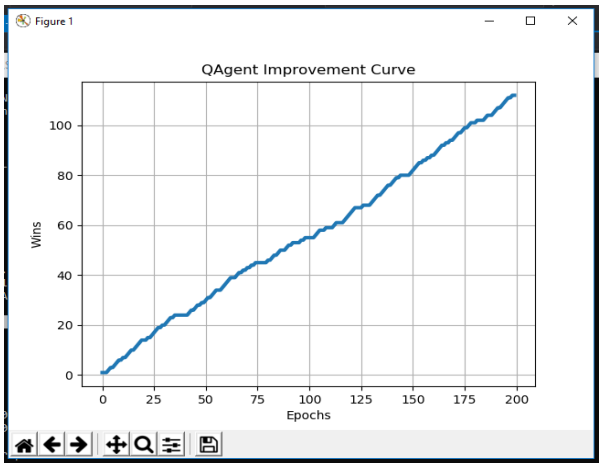


Alpha = 0.1

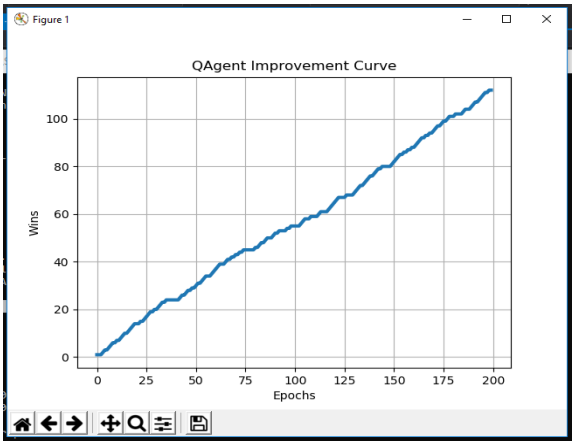


- השינוי הוא קטן אבל התוצאה משתנה מהר יותר

Epsilon = 0.2

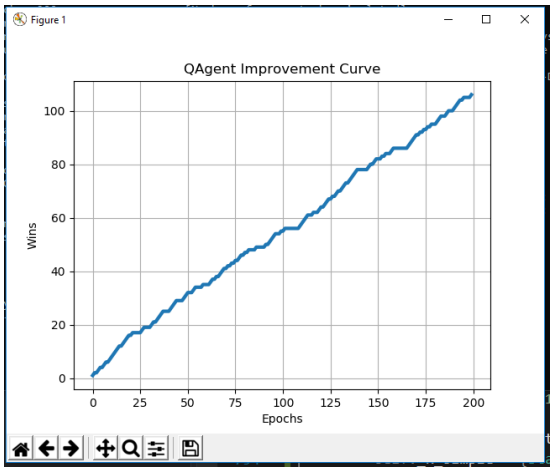


Epsilon = 0.1

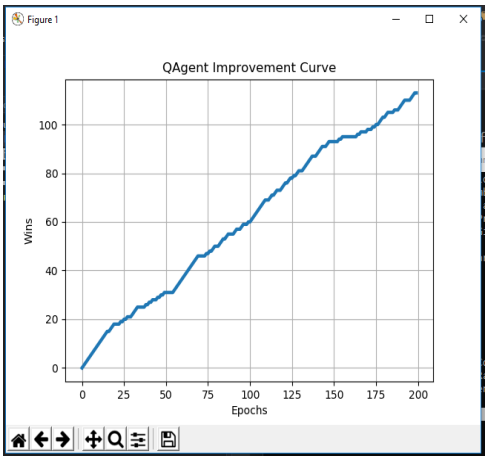


- השינוי הוא קטן אבל התוצאה מתחילה בכמות ניצחונות פחותה ואז עולה מהר

gamma = 0.2



gamma = 0.1



- שינויים מהירים יותר בעקבות עדכון משמעותי יותר של השלב הבא

סיכום

האלגוריתמיקה והתהליך כולה של למידת הסוכן היא מאוד משמעותית, אבל כיש לנו כבר מודל שניתן לאמן אותו באופן אידיאלי עובר חוט מאוד דק שצריך לחקור אותו בין סוכן טוב לסוכן לא טוב שתלוי בעיקר בפרמטרים ובכמות המשחקים שניתן לאמן אותו עליהם.

אם סכום הכסף משתנה הסטטיסטיקה תיראה טיפה אחרת (לטובה וגם לרעה) אבל ההתנהגות שתארנו הנ"ל תשמר – נדרש עדיין לבחור פרמטרים בצורה נכונה.

למשל, עבור Starting amount=5 נקבל ב-200 משחקונים 59% הצלחה בעוד שגם כמות ה-penalty תרד משמעותית (~-1.8)