Homework assignment 1

Ori Roth 315859553

Shahar Cohen 313566077

הקוד מורכב מספר שיטות מרכזיות הבאות למצוא את המדיניות הטובה ביותר למשחק בלאקג'ק. הקוד מתבסס על שיפור מטריצת המדיניות בהתאם לתצפיות המתקבלות מהסביבה Blackjack, ומשפר באופן איטרטיבי את מטריצת המדיניות (policy) הקובעת האם לקחת עוד קלף בהתאם למצב, במטרה להגיע ל21.

כל הקוד מתועד ומוסבר מורחב על כל פונקציה נמצא בקוד.

הקוד מתבצע באמצעות הפעולות הנ"ל:

transition_matrix

פונקציה שמחשבת את מטריצת המעבר ומטריצת התגמול עבור הסביבה .(הבלאק ג'ק) באמצעות סימולציה. הפונקציה מאתחלת את המטריצות הללו בהתבסס על מצב הסביבה ומרחבי הפעולה. לאחר מכן מבצעת מספר מוגדר של שלבים, בוחר באקראי פעולות בכל שלב ומעדכן את המטריצות בהתאם.

הפונקציה תחזיר:

מטריצת מעברים P , מטריצת תגמול R

כל מטריצה כזאת מוגדרת כגודל (n,d,a,n+1,d+1)

כאשר:

n - סכום השחקן

d - קלף הדילר המוצג

a - מספר הפעולות האפשריות

המטריצות ממפות מעבר מכל מצב למצב אחר אפשרי, הוספנו מצב נוסף(n+1,d+1) שהוא מצב מסיים, מגיעים למצב זה רק כאשר המשחק מסתיים וכן מטריצת התגמול מתעדכנת בנקודה זאת.

** את הפונקציות הבאות הגדרנו לפי האלגוריתם שהוצג בהרצאות

1. Initialization
   $V(s) \in \Re$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Repeat
   $\Delta \leftarrow 0$
   For each $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $V(s) \leftarrow \sum_{s'} \mathcal{P}_{ss'}^{\pi(s)} \left[\mathcal{R}_{ss'}^{\pi(s)} + \gamma V(s') \right]$
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   $b \leftarrow \pi(s)$
   $\pi(s) \leftarrow \arg\max_a \sum_{s'} \mathcal{P}_{ss'}^{a} \left[\mathcal{R}_{ss'}^{a} + \gamma V(s') \right]$
   If $b \neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop; else go to 2

modified_policy_iteration

פונקציה זו מתחילה בביצוע מדיניות ראשונית קבועה, מעריכה ומשפרת את המדיניות באופן איטרטיבי.
המדיניות משתנה על סמך ערכי התגמול המחושבים בשלב הערכת המדיניות

המדיניות הראשונית שנקבע היא לשלוף קלף כל עוד לא הגענו ל-21.

הפונקציה מבצעת את הצעדים הבאים:

1. קריאה לפונקציה התגמול בכדי לקבל את V
2. ביצוע policy improvement לפי פונקציית התגמול שהתקבלה

הפונקציה מחזירה את מטריצת המדיניות P, ומטריצת הערך V

policy_evaluation

פונקציה שמעריכה מדיניות נתונה באמצעות משוואת בלמן ומעדכנת באופן איטרטיבי את פונקציית הערך V.

initial_policy

פונקציה היוצרת את המדיניות ההתחלתית, מדיניות שתמיד מבקשת עוד קלף אם הערך נמוך מ-21. מחזירה מטריצת מדיניות i_policy

play_game

פונקציה המקבלת את הסביבה ומדיניות ומשחקת את המשחק.

הפונקציה מדפיסה את הרווח הממוצע לאחר 100 משחקים.

value_function_q3

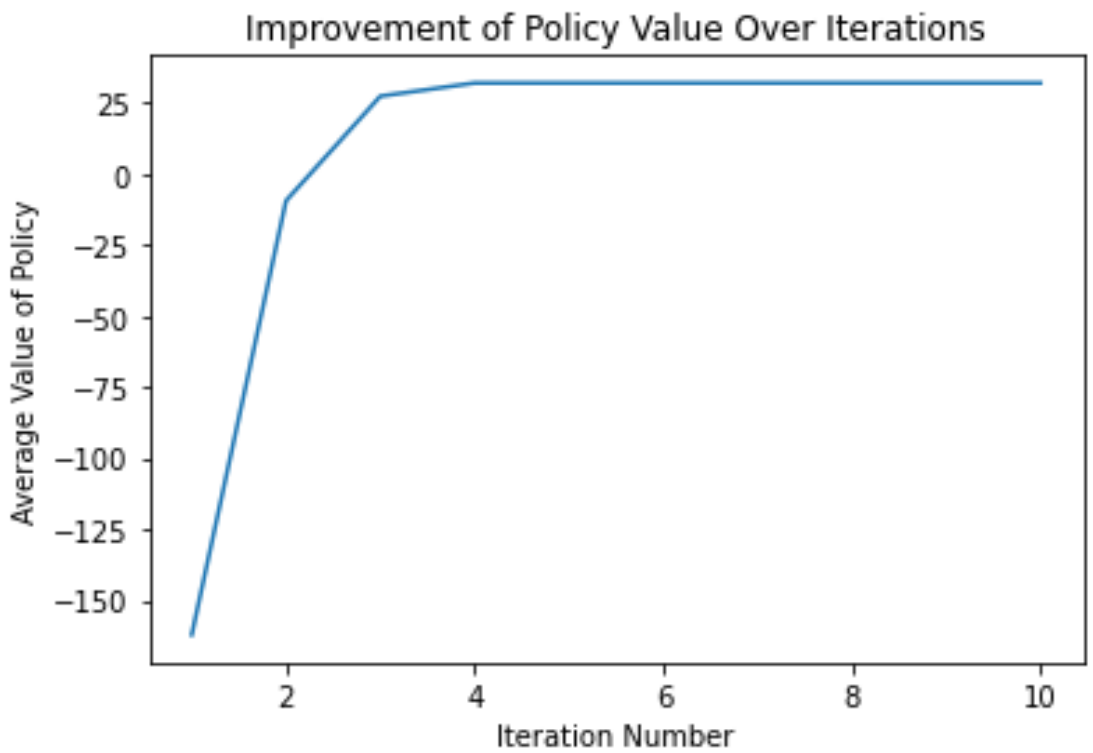
הפונקציה מקבלת מטריצת ערך V ומצב ומחזירה את הערך של המצב

Ploting

הפונקציה מקבלת פונקצית ערך ומדיניות ומפיקה פלוט מדיניות ופלוט פונקציית הערך.

בנוסף הוספנו עוד מספר שיטות להפקת גרפים, הדפסת ערך של מצב מסוים, והפעלת המשחק בהתאם למדיניות.

גרף שיפור ערך המדיניות:



טבלה המתארת את המדיניות האופטימלית בהתאם למצבים התחלתיים:

הטבלה מייצגת את הסכום מ-4-20 והמדיניות כאשר 0 זה לא לקחת קלף ו-1 זה לקחת