Fall 2018 - Artificial Intelligence (CSCI-3613 - 10134)

Final Report

Emil Kalbaliyev, Mardan Safarov, Vasif Vahidov

Q-learning for Blackjack

Abstract

The purpose of our project is to create Blackjack game which shows an optimal strategy to player for earning more rewards than usual player. This project endeavors to make a agent to learn how to play our version of blackjack game and defeat the opponent. We implemented Q-learning to obtain optimal strategy.

PROJECT DESCRIPTION

Blackjack, also known as twenty-one, is a comparing card game between players and a dealer, where each player in turn competes against the dealer. It is played with one or more decks of 52 cards, and is the most widely played casino banking game in the world. Objectives of the blackjack are :

- Get 21 points on the player's first two cards without allowing dealer get 21 points;
- Reach a final score higher than the dealer without exceeding 21
- Or let the dealer draw additional cards until their hand exceeds 21.

In this project, we have created simplified version of Blackjack game where player can take 3 decision:

- **Hit (H)**: Take another card from the dealer.
- **Stand (S)**: Take no more cards
- **Double down (D):** The player is allowed to increase the initial bet by up to 100% in exchange for committing to stand after receiving exactly one more card

Based on selected decisions and actions, final result is rewarded as following:

-1 : if player busts (get more than 21); dealer get 21; dealer get more than player

0 :both get 21

1 : if dealer busts; player get more than dealer

1.5 : if player is 21

2x : increase reward by 2 times if double down chosen

METHODOLOGY

Reinforcement Learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. It refers to goal-oriented algorithms, which learn how to attain a complex objective (goal) or maximize along a particular dimension over many steps; for example, maximize the points won in a game over many moves. They can start from a blank slate, and under the right conditions they achieve superhuman performance.

For finding best strategy for our version of Blackjack game, we implemented Q-learning algorithm, one of the Reinforcement Learning techniques, to our project. Q-learning algorithm is

$$Q^{new}(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$

There are several parameters we should pass for learning best policy.

- Learning rate (α) - determines to what extent newly acquired information overrides old information. In this project, we take α =0.01 as learning rate.
- Discount factor (γ) - determines the importance of future rewards. We take discount factor as 1.

Furthermore, we use Epsilon - Greedy selection method to choose a random action with a some probability. In other words, with given probability ε, agent act randomly while with 1-ε probability, agent act on current policy. We take ε = 0.99, which means that the selected action is exploring most of the time (choosing random actions), while selected action is still exploitative in a small probability(choose best action believed).

RESULTS

Experiment 1. After creating our version of Blackjack game, we implement Q-learning algorithm to the game. However, we faced the challenge what to take as values of unknown parameters, such as, learning rate (α) and epsilon-greedy (ε). So we decided to find best values for α and ε. We made our agent to play 10000 games and tested learned policy on 10000 games for each iteration while changing the values of α and ε. During the experiment learning rate took values between $0.01 < \alpha < 0.5$, and epsilon greedy took values between $0.01 < \varepsilon < 1$ while Discount factor value $\gamma = 1$. For each iteration, value of α and ε is increased by 0.01.

While analyzing win rates for each iteration (Figure 1), we observe that while in low values of learning rate (α) we get more winning rate. In case of epsilon, in different value of epsilon, we get different results as seen from Figure 2. However, best result is achieved when $\varepsilon = 0.99$ and $\alpha = 0.01$ (Figure 3).

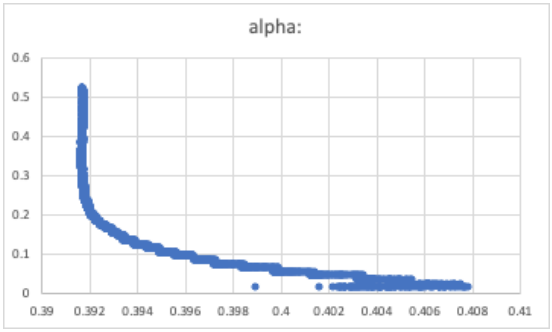

Figure 1. Learning rate vs Winning rate

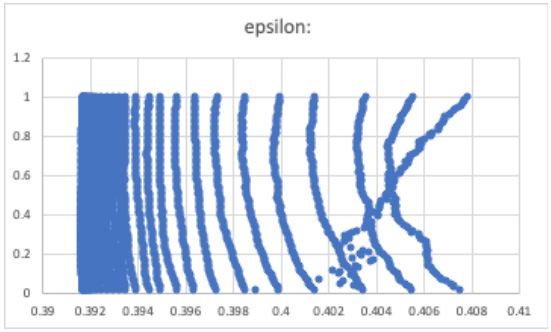

Figure 2. Epsilon vs Winning rate

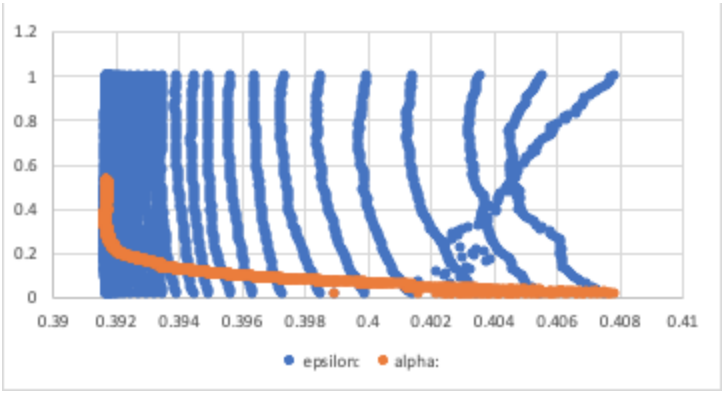

Figure 3. Relations of Learning rate, Epsilon and Winning rate

As result, we decided to take $\varepsilon = 0.99$, $\alpha = 0.01$ and $\gamma = 1$. With this parameter we make our agent to play 1000000 games and learn best strategy for Blackjack game. Learned policy is shown in Table 1 and 2 where H-HIT, S-Stand, D-Double.

Hard										
	2	3	4	5	6	7	8	9	T	A
2	H	H	H	H	H	H	H	H	H	H
3	H	H	H	H	H	H	H	H	H	H
4	S	H	H	S	D	H	H	H	H	H
5	D	H	D	D	H	H	H	H	H	H
6	D	D	S	S	S	D	S	H	H	H
7	H	D	H	H	H	S	D	H	H	H
8	D	S	S	H	H	D	H	H	H	H
9	H	D	D	D	H	D	D	H	H	H
10	D	D	D	D	D	H	D	H	D	H
11	D	D	D	D	D	D	D	D	D	H
12	S	D	D	D	H	H	D	H	H	H
13	H	D	S	S	S	H	H	H	H	H
14	S	S	H	S	S	D	H	D	H	H
15	H	D	S	S	H	H	S	S	H	H
16	S	S	S	S	S	S	D	H	S	H
17	D	S	S	S	S	S	S	H	S	S
18	S	S	S	S	S	S	S	S	S	D
19	S	S	S	S	S	S	S	S	S	S
20	S	S	S	S	S	S	S	S	S	S
21	S	S	S	S	S	S	S	S	S	S

Table 1. Hard table - If player has no Ace

Soft										
	2	3	4	5	6	7	8	9	T	A
13	H	S	S	H	S	D	H	H	H	H
14	D	S	H	D	S	H	D	D	H	H
15	S	H	S	D	H	S	S	H	H	H
16	S	D	D	H	H	D	D	H	H	H
17	H	D	D	D	H	D	D	H	H	H
18	S	S	H	S	H	S	D	S	H	H
19	D	H	S	H	S	D	D	D	H	H
20	S	D	S	D	H	S	D	D	S	S
21	S	S	D	S	S	S	D	S	S	S

Table 2. Soft table - If player has Ace

After finding best policy for our game, we tested learned policy on 500000 game, and get **41.6 %** winning rate.

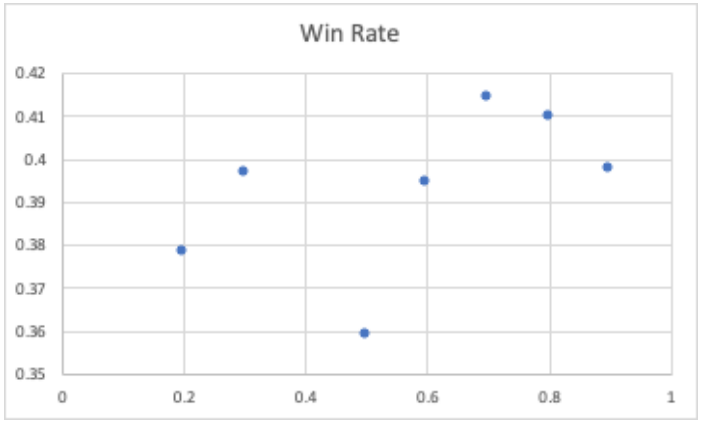

Figure 4. Relation between winning rate and decrease rate of epsilon

Experiment 2. For increasing winning rate, we do another experiment, which is about decreasing values of ε over time. We make our agent to play 100000 games and learn best policy, and we made tested learned policy on 50000 games. During learning process. We use same values of parameter($\varepsilon = 0.99$, $\alpha = 0.01$ and $\gamma = 1$), however decreased epsilon by 0.2 (0.3, 0.5, 0.6, 0.7, 0.8, 0.9) for each 1000 games. As shown in Figure 4, in any decrease rate of epsilon we could not get better result as previous. So, we decided not to use this method for our project.

Experiment 3.* For getting better results, we decided to increase the number of games that our agent try to learn optimal policy. Furthermore, we try to find appropriate values for earning rate (α) and also discount factor (γ) while taking epsilon-greedy as ε=0.9. We made our agent to play 50000 games for learning policy and tested learned policy on 100000 games for each iteration while changing the values of α and γ . Considering time limitation, during the experiment learning rate took values between $0.1 < \alpha < 1$, and discount factor took values between $0.1 < \gamma < 1$ while value of epsilon $\varepsilon = 0.9$. For each iteration, value of α and γ is increased by 0.1. As result of our experiment, we find out that winning rate increased to **43.54%** when policy learned with parameter of $\alpha = 0.1$, $\gamma = 0.1$, $\varepsilon = 0.9$. Now, our game use this policy to assist user to play optimally.

**After the instructor`s suggestion on our project presentation*

CONCLUSION

As a result, as a team, we researched and found the technique of reinforcement learning in order to achieve the best results for randomly played Blackjack game. The most efficient method to use for the type of our game, Q-Learning algorithm is applied in our code-based function. Moreover, we used Epsilon-Greedy selection method to increase the best possible random actions to be achieved for our gameplay. Despite the fact that in our previous attempts, the maximum winning rate was 41.6%, having the number of games increased by proportional amount we successfully obtained 43.54 percentage of winning rate for our Blackjack game.