Reinforcement Learning

Intelligent Systems Lab Assignment

Yahor Dziomin, Alexandru Gliga

March 2024

1 Introduction

In this study, we implement Q-learning, one of the most popular reinforcement learning algorithms, to tackle the problem of playing Blackjack. Our aim is to explore various strategies and parameters in order to understand how they affect the learning process and the agent's performance.

In this sense, we propose the following research questions:

1. How does learning progress vary for different strategies:
   (a) Bot takes into account only his hand;
   (b) Bot takes into account both his and dealer's hand.
2. How does the additional information about existing active Ace impact the learning progress of the above-mentioned algorithms?
3. How does the exploration parameter ϵ influence the learning process of the best-performing found by us agent?

2 Methods

In this section, we describe the methods employed in our study to implement Q-learning algorithms for playing Blackjack. Additionally, we provide details on the rewards function used in our experiments to reinforce the learning process.

2.1 Bot Algorithms

We implement two distinct bot algorithms for our Q-learning agent to address the problem of playing Blackjack. These algorithms vary in terms of the information considered during decision-making:

1. **Bot Considering Only Its Hand:** This algorithm involves the agent making decisions based solely on the information about its own hand. The agent does not take into account the dealer's hand when determining its actions in the game.
2. **Bot Considering Both Its and Dealer's Hand:** In contrast, this algorithm equips the agent with knowledge about both its own hand and the dealer's hand. This allows the agent to make decisions by considering the potential outcomes and strategies of both parties involved in the game.

Exploratory Algorithm: Both versions of the bots utilize an epsilon-greedy exploratory algorithm. This method balances between exploration and exploitation by choosing a random action with probability ϵ, and the action with the highest Q-value otherwise. We set ϵ to a balanced value of 0.1.

2.2 Rewards Function

In our experiments, we define the rewards function as follows:

- If the bot wins the game, it receives a reward of 1 point.

- If the bot loses the game, it receives a reward of -1 point.
- In the case of a draw, the bot receives a reward of 0 points.

3 Experiments and Results

3.1 Experiment Setup

We conducted a series of experiments to address the research questions outlined in the introduction. Each experiment was designed to evaluate the performance of the implemented bot algorithms under different conditions.

3.1.1 Blackjack Environment

The experiments were conducted within a standardized Blackjack environment, which consisted of the code provided from the start.

3.1.2 Parameter Settings

We maintained consistent parameter settings across all experiments, including the learning rate ($\alpha = 0.1$), discount factor ($\gamma = 0.9$), and exploration parameter ($\epsilon = 0.1$).

3.2 Experimental Design

For each research question, we designed specific experiments to investigate its impact on the learning process and the agent's performance.

3.2.1 Learning Progress for Different Strategies

To assess how different strategies affect learning progress, we conducted experiments comparing the performance of the bot algorithms considering only the agent's hand versus considering both the agent's and the dealer's hands. We also considered the random agent that we got at the start. For that, we plotted their average accumulated reward over 100.000 games. Also, we ran this experiment 100 times and averaged the results (Figure 1).

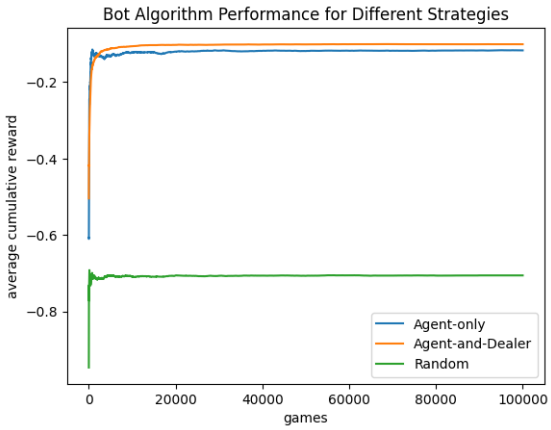

Figure 1: Bot Algorithm Performance for Different Strategies

3.2.2 Impact of Active Ace Information

We investigated the influence of active ace information on learning progress by comparing the performance of the bot algorithms with and without consideration of active aces in the hand. For each of these bot algorithms, we create a variant that also takes into account whether the bot holds an active ace in its hand. This extension enables the bot to adapt its decision-making process based on the presence of an ace and its potential value of 1 or 11.

Again, we plotted their average accumulated reward over 100.000 games and we ran this experiment 100 times and averaged the results (Figure 2).

3.2.3 Effect of Exploration Parameter ϵ on Learning Process of Best Algorithm

To study the effect of different ϵ values on the learning process, we varied the exploration parameter ϵ within a predefined range ([0, 0.5, 0.1, 0.15, 0.2, 0.25, 0.3]) and monitored the learning progress (averaged cumulative reward) of our best agent over 100 averaged iterations.

To be sure that we picked our best algorithm, we calculated the average accumulated reward over 10.000.000 games for our 2 best-performing models (agent that knows of both hands and agent that ad-

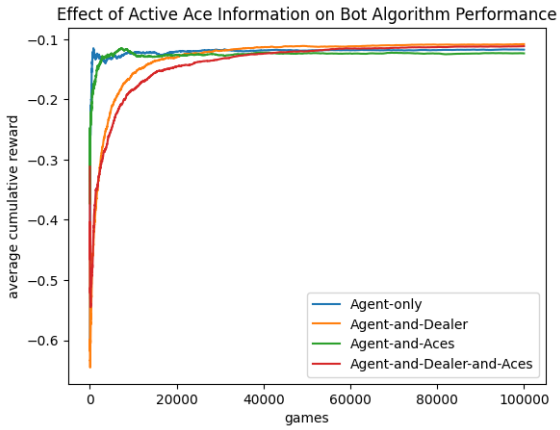

Figure 2: Effect of Active Ace Information on Bot Algorithm Performance

ditionally knows if it holds an Ace). We averaged the results over 100 iterations and plotted the last 10.000 games (Figure 3).

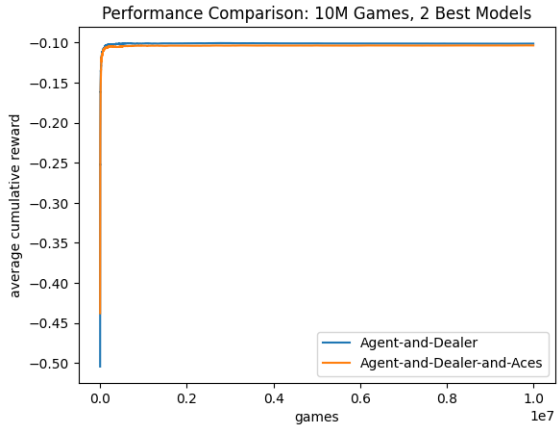

Figure 3: Performance Comparison: 10M Games, 2 Best Models

Given the previous results, we used the agent that knows of both hands to see what is the effect of the above-mentioned ϵ values on the learning process. We plotted the average accumulated reward over 100.000 games and we ran this experiment 100 times and averaged the results for each ϵ (Figure 4).

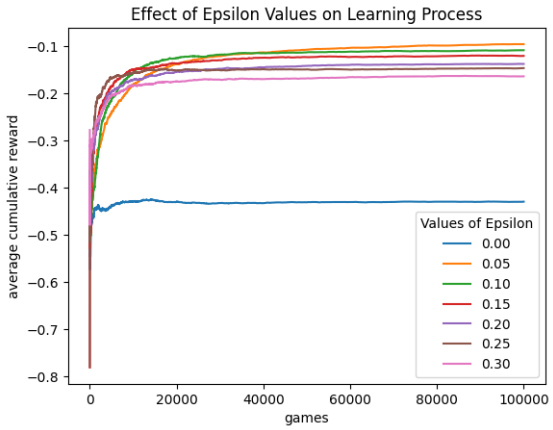

Figure 4: Effect of Epsilon Values on Learning Process with Agent Knowledge of Both Hands

4 Discussion and Conclusion

In this study, we explored the application of Q-learning algorithms to the problem of playing Blackjack and investigated various strategies and parameters that influence learning progress and agent performance. Through a series of experiments, we addressed several research questions regarding the impact of different factors on the learning process.

Firstly, we examined how learning progress varies for different strategies, specifically comparing the performance of bot algorithms considering only the agent's hand versus considering both the agent's and the dealer's hands. Our results, as depicted in Figure 1, indicate that considering both hands leads to improved performance. This suggests that incorporating information about the dealer's hand is beneficial for the decision-making process in Blackjack. Also, we see that the learning curve for the agent that considers only its hand is very sharp, while the other one is much smoother. The difference in convergence between agents can be explained by the fact that information about dealer's hand expands the number of Q-value pairs by a factor of 10 while an ace increases the number of states by a factor of 2. Thus, it takes considerably more time to converge for an agent with its and dealer's hand information com-

pared to its own hand with an indicator of an ace. Interestingly, additional information does not always improve the performance: an agent with notion of both hands plays better than an agent which takes into consideration both hands and presence of an ace in its hand. An additional dimension in the Q-Table can make a model more complex and stable but not always better. Furthermore, the performance of the random agent is far below the other 2.

Secondly, we investigated the influence of active ace information on learning progress by comparing the performance of bot algorithms with and without consideration of active aces in the hand. As illustrated in Figure 2, our findings show that accounting for active ace information did not further enhance the performance of the bot algorithms. So, this cannot suggest that adapting the decision-making process based on the presence of an ace and its potential value can lead to better outcomes in Blackjack.

Finally, we studied the effect of the exploration parameter (ϵ value) on the learning process of the best-performing algorithm. By varying ϵ within a predefined range and monitoring the learning progress, we gained insights into how exploration-exploitation trade-offs impact the agent's performance. The best performance was shown by the ϵ with a value of 0.05. Bigger ϵ's, however, showed a worse score proportional to their increase. Moreover, the worst performance was shown by the $\epsilon = 0$, which means that exploration is a vital component for the learning performance in our scenario.

In conclusion, our study highlights the importance of considering various strategies and parameters in the design and implementation of reinforcement learning algorithms for playing Blackjack. By understanding how these factors influence learning progress and agent performance, we can make informed decisions to optimize the behavior and effectiveness of the agents in real-world scenarios.

5 Acknowledgements

We would like to express our gratitude to ChaptGPT. It helped us structure the report and reformulate our sentences in a more readable way. Also, the last time we touched java code was the first-year project. So, we used it as a fast documentation and debug mechanism for Java syntax.