Developing and Comparing Reinforcement Learning Strategies in Blackjack

Ceci Chen, Hongxin Song, Lia Wang, Maggie Xu
DS-GA 1016 Computational Cognitive Modeling

Abstract

This study applies reinforcement learning techniques to develop a blackjack decision-making model. The model's performance is assessed against human strategies by analyzing 900,000 hands from a Kaggle dataset. Our objective is to determine if the model can outperform human decision-making in terms of winning rates, highlighting the potential of reinforcement learning in games of chance and strategy. Though our results show varied accuracy for different models, the machines generally do not outperform human in gaming strategies. We conclude that reinforcement learning is not sufficient for winning a game with high degree of randomness like blackjack. Human decision-making consists of more complex features and conditions than pure machine learning algorithms.

Introduction

Reinforcement Learning (RL) and Cognitive Science are deeply intertwined, with RL providing a robust framework for modeling the complex decision-making processes observed in biological systems, and cognitive science offering insights to enhance and ground RL algorithms in biological realism. In RL, agents learn from their environment through a system of rewards and punishments, mirroring the way humans and animals learn through similar feedback mechanisms. This process is akin to the operation of the brain's reward systems, such as those mediated by dopamine, which reinforce behaviors based on received signals of reward or punishment. Cognitive science utilizes these parallels to better understand neural learning processes, while RL gains a framework for simulating cognitive behaviors in artificial agents.

Cognitive science, on the other hand, benefits from RL's ability to model and predict human behavior in controlled experiments, facilitating deeper understanding of underlying cognitive processes such as exploration vs. exploitation. Moreover, integrating cognitive models like memory and attention into RL can lead to more efficient learning algorithms, while insights from neuroscience can improve the effectiveness of RL techniques. This symbiotic relationship not only advances our understanding of human cognition but also aids in developing more human-like machines, thereby driving progress in fields ranging from behavioral psychology to cognitive robotics and adaptive educational systems.

In this project, we introduce a reinforcement learning model designed to optimize decision-making in the game of blackjack. We aim to explore whether machines can develop strategies to maximize an individual's benefits in a game characterized by significant randomness. The model utilizes advanced RL algorithms—**Q-learning, SARSA (State-Action-Reward-State-Action), and Monte Carlo (MC) methods**—to simulate and learn from millions of blackjack hands, learning optimal strategies by adjusting its policy based on the game's current state.

Blackjack, also known as twenty-one, is a popular card game where players compete against a dealer. The game's primary goal is to draw cards to achieve a total closer to 21 than the dealer, without exceeding 21. Its combination of chance, strategy, and skill provides a fertile ground for exploring the capabilities of reinforcement learning (RL) in decision-making under uncertainty. The rules we following in this project is as below:

"The game starts by dealing two cards each to the player and the dealer. The dealer shows only one card , while the other remains face down. If the player's initial two cards are an ace and a 10-card, totaling 21, this is known as a natural. The player wins with a natural unless the dealer also has one, leading to a tie. If there's no natural, the player can take additional cards one at a time (hits) until he chooses to stop (stands) or exceeds 21 (busts). If the player busts, he loses immediately. If he sticks, the dealer then plays. The dealer must follow a set rule: hitting on totals below 17 and standing on 17 or above. If the dealer busts, the player wins. The game's result—win, lose, or draw—depends on whose total is nearest to 21. An ace is considered usable if the player can count it as 11 without busting." (Governor of Poker, 2022)

The core of our model is a policy network that evaluates the expected rewards for taking actions (hit, stand, double down, split) based on the current sum of the cards, the dealer's visible card, and other game rules. Through the application of Q-learning, SARSA, and MC algorithms, the model iterative adjusts its strategies, aiming to converge on the most effective approach for maximizing winnings.

A critical aspect of our study is to **compare these machine-derived strategies against human decision-making**. This comparison will be grounded in the analysis of 900,000 hands of blackjack data sourced from Kaggle, providing a robust dataset for evaluating the final winning rates of both the model and human players. By evaluating the final winning rates of both the model and

human players, we aim to determine whether a self-learning model can outperform human intuition and rationality in this domain of high randomness and strategic depth.

By leveraging this extensive dataset and sophisticated RL techniques, we aim to demonstrate the potential of reinforcement learning to master complex decision-making tasks and to provide insights into the effectiveness of AI techniques in gaming and entertainment scenarios.

Data

The dataset was created by analyzing a million Blackjack hands using code that simulates the card distribution typical of a casino setting. The objective is to identify patterns and develop a strategy that maximizes winning opportunities. It includes various columns that represent the cards dealt to both players and the dealer, the total sum of the cards, and the outcome of each round, indicating whether it was won by the player or the dealer.

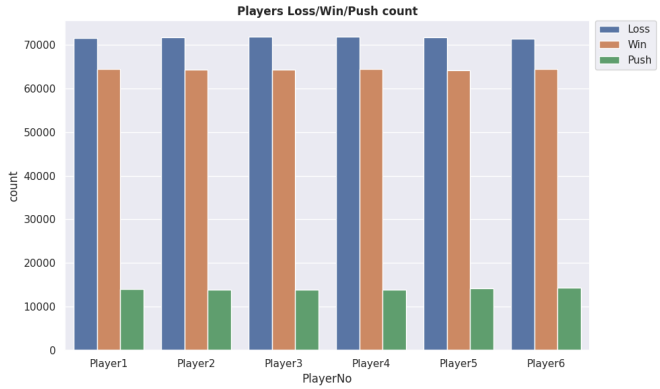

Figure 1: The figure shows Loss/Win/Push count of the human dataset we collected from Kaggle

Methods/Models

OpenAI Gym

In our project, we employed the ‘gym‘ package launched by OpenAI to evaluate reinforcement learning techniques. The ‘gym‘ environment provides a well-defined a structured simulation of Blackjack, in which the primary objective is to achieve a card total closer to 21 than the dealer without busting. This environment not only embodies the basic rules and operations of Blackjack but also introduces stochasticity that mirrors real-world various scenarios, making it an ideal platform for evaluating and improving learning algorithms.

Utilizing the ‘gym‘ package, our method involves applying different reinforcement learning strategies to build a decision-making model. With the package’s simulations, we are able to optimize actions and policies in various game states. This package setup allows for the consistent and repeatable evaluation of our model’s effectiveness, which makes our finding applicable in the broader context of AI-driven decision making and gaming strategies.

Monte Carlo Method

In reinforcement learning, the Monte Carlo method is usually used when the environment is unknown. This method estimates the value function by calculating the average of return from sampled episodes. The agent will continuously interact with the environment to explore in a self-play mode and collect rewards after each episodes ends for every state-action pair. The values of these pairs are updated by averaging the returns observed at the end of each episode. To balance exploration and exploitation, we also combined the Monte Carlo method with ε-greedy policy, which selects actions randomly with a probability of exploration (ε) and chooses the best-known action with a probability of exploitation(1-ε). As learning progress continues, ε will gradually decay, shifting from exploration towards more exploitation.

In our blackjack case, we initialized by setting all state-action values to zero and the ε-greedy policy is set with an initial value. During the simulation, actions of each game of Blackjack are chosen with the ε-greedy policy and the game will continue until the player stands or busts. The returns of every state-action pair will be collected at the end of each game and the estimation of vales will be updated accordingly. The policy will also be updated accordingly, which will prefer action with highest value estimation for following games in each state.

After millions of times simulating, the Blackjack agent is able to improve the playing strategy and update the policy to achieve the optimal ones, which help itself make better decisions because it will gain more information from data as the game proceeds.

Temporal Difference Learning

Temporal Difference learning is a method that learns through bootstrapping from the current estimate of Q-values. Different from the Monte-Carlos Method, TD learning updates estimates based on immediate estimates instead of waiting until the end of each episode to update. TD error is the difference between the estimation of the current state and the prediction of the current policy using the next state’s estimation, which is given by a formula:

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$$

δ_t is the TD error at time t, R_{t+1} is the reward received after proceeding to the next state, γ is the discount factor, $V(S_{t+1})$ is the estimation of the next state, and $V(S_t)$ is the current estimation of the current state.

Q-Learning Q-learning is a off-policy method of TD learning. The Q-learning method learns through the action-value function, which is known as Q-value. By

updating the Q-table, we aim to maximize the overall reward. For the Black jack problem, we initialize the Q-values table for each state-action pair with zero. Win for player would yield positive rewards where as lose would yield negative rewards, and a draw would give zero reward. The exploration rate decreases after each episode, monitored by the exponential decay function.

The Q-values are updated using the following rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

s is the current state, a is the action taken, s' is the new state after taking action a, r is the reward received after taking action a, α is the learning rate ($0 < \alpha \leq 1$), γ is the discount factor ($0 \leq \gamma < 1$), and $\max_{a'} Q(s',a')$ is the highest Q-value for the next state s' considering all possible actions a'.

SARSA SARSA, the State-Action-Reward-State-Action algorithm, is a on-policy learning methods. Different from Q-Learning, it learns the value of the current policy, including the exploration steps. Similar to Q-Learning, we first initialize a q-value table with zero values. For each episode, the initial state s is observed and then an action a is chosen for that state using a policy derived from Q. After executing a, the OpenAI envioronment would generate next state s' and a corresponding reward r. Then, the next action a is chosen by the same policy. For next step, we need to update the Q-value of (s, a) by the formula:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma Q(s',a') - Q(s,a) \right]$$

where α is the learning rate and γ is the discount factor.

Q-Learning with Double Q-Tables Method Unlike standard Q-learning, which often overestimates action values due to the maximization step in its update rule, Double Q-learning addresses this by maintaining and updating two separate Q-tables: one table to determine the best action and the other to evaluate its value, which can reduce bias and often leads to more stable and accurate learning outcomes.

In our blackjack model, we start by initializing two Q-tables with zero values to separately estimate the action values. This separation aims to reduce the overestimation bias common in traditional Q-learning. In the observed state, the model uses an ε-greedy strategy, randomly selecting actions with probability ε to encourage exploration, and with probability $1-\varepsilon$to maximize the combined values from both Q-tables. In the self-play process, the model executes the chosen action and observes the resulting reward and new state. The update to the Q-table that directed the action selection involves calculating the Temporal Difference (TD) error, which is expressed as:

$$Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \alpha \left[R_{t+1} + \gamma \max_{a} Q(S_{t+1},a) - Q(S_t,A_t) \right]$$

where Q denotes the Q-table value, S_t the current state, A_t the action taken, R_{t+1} the reward received, α the learning rate, and γ the discount factor.

By simulating millions of hands and iteratively updating the Q-tables based on the outcomes, the agent refined its strategy over time.

Results

From the provided table, it's clear that the winning rates of different models and methods in blackjack vary, with the Q-Learning with double Q-Tables model achieving the highest rate at 44.32%. This is followed closely by human players, who have a winning rate of 42.8%. The other RL models, such as Monte Carlo, standard Q-Learning, and SARSA, show lower performance, with winning rates of 39.19%, 40.73%, and 39.62% respectively.

Table 1: Winning Rate for Each Model

Model & Method	Winning Rate
Monte Carlo	39.19%
Q-Learning	40.73%
SARSA	39.62%
Q-Learning w/ double Q-Tables	44.32%
Human	42.8%

The three Q-Learning methods display varied effectiveness in the Blackjack simulations. While Standard Q-Learning offers robustness and simplicity, it can sometimes lead to less precise Q-value estimations due to overestimation biases. SARSA is more conservative compared to the standard method, but it might result in lower winning rates where aggressive strategies prevail. Meanwhile, Double Q-Tables mitigates the shortcomings of the other two, leading to a more balanced and effective approach in scenarios requiring precise decision-making under uncertainty.

For the Q-learning model, we plot out the value of each player's hands and the dealer's card condition. These two plots display how the estimated values of different states change with respect to the player's total hand sum and the dealer's displayed card. We can see that both graphs have a peak around 20 and 21, which means they know what action to take around these stages. This is logical because any move other than 'stand' would be foolish since it would give the player with busts. The plot for usable ace shows overall higher performance compared to that of without usable ace. This suggests the importance of having an ace. Since players can use ace flexibly as 1 or 10, it helps with generating more optimal strategies.

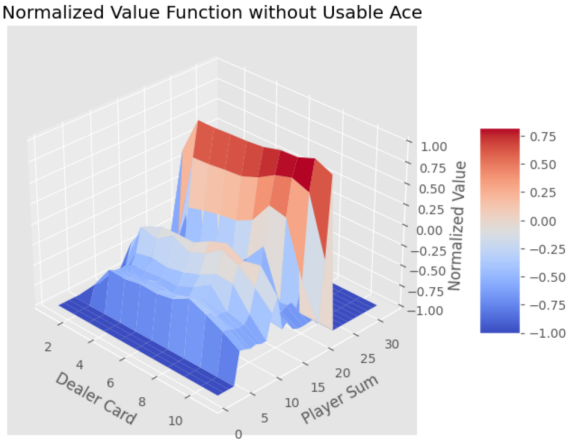

Figure 2: Q-Learning Q-table Value Function without Usable Ace

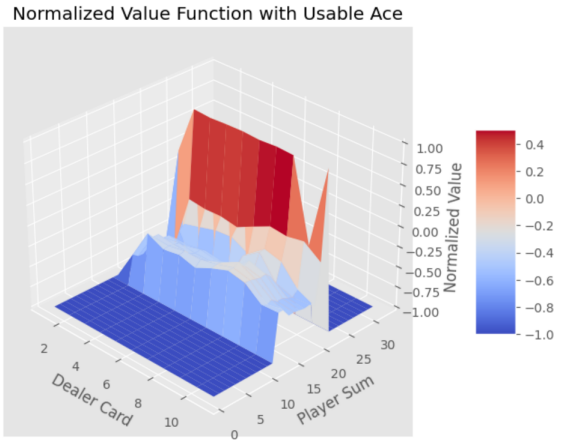

Figure 3: Q-Learning Q-table Value Function with Usable Ace

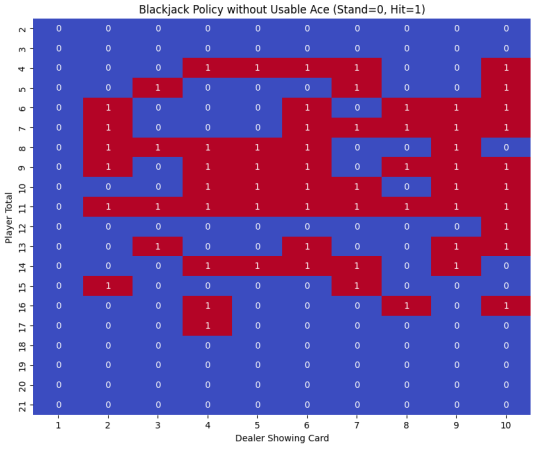

Figure 4: The figure shows the policy of the model after testing of whether the model choose to stand or hit without usable Ace

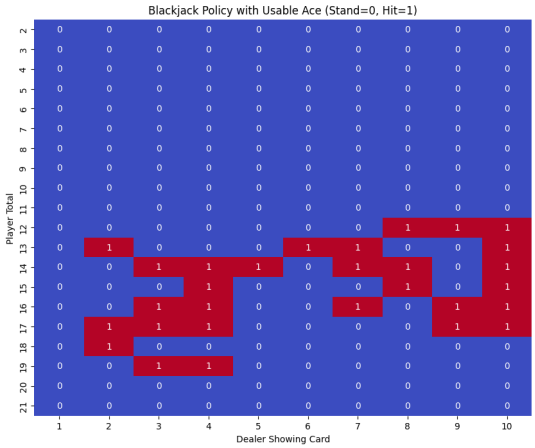

Figure 5: The figure shows the policy of the model after testing of whether the model choose to stand or hit with usable Ace

Monte Carlo method gives out the policy shown in the figures above. Under the condition when Ace is usable (count as 11 to the final total value), the model gives a strategy that choose to hit when the sum of first two cards between around 13 to 18. Under the condition when Ace is not usable (count as 1 to the final total value), the model gives a strategy that choose to hit when the sum of first two cards between around 4 to 14.

Conclusion & Discussion

These results suggest that while the advanced Q-Learning model with double Q-Tables slightly outperforms human players, the margin is not overwhelmingly large. Furthermore, the other models do not exceed human performance. These results indicate that in the domain of blackjack, which involves a high degree of randomness and requires strategic depth, the Reinforcement Learning (RL) models do not universally outperform human intuition and rationality. This conclusion supports the idea that while RL can be an effective tool for navigating complex and uncertain environments, there remains a significant component of human strategic thinking and decision-making that these models have yet to fully replicate or exceed.

When looking at specific method like Monte Carlo method we used here, we can make a possible conclusion here that usable Ace do increase the chance of winning in the game. This guess also can be illustrated by the Figure 4 and Figure 5 in Result section that Monte Carlo method do suggest a policy to avoid ”hits” when the sum of player’s first two cards are below 12 with usable Ace. This is also evident from the image in Sutton’s book (Figure 6) that he says ”the front-most values with usable ace are slightly higher than that without usable ace”. We think this might due to the reason that a player with usable ace will have greater chance to hit. Even if the value crosses above 21, the player will still be able to keep in safe state by making the usable Ace into non-usable.

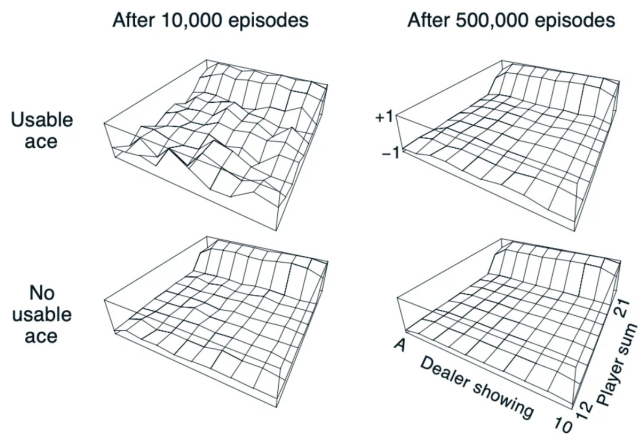

Figure 6: Simulation Result from Sutton's Book

Possible Improvements for RL Models

Human players often excel in blackjack due to a combination of cognitive skills, emotional intelligence, and experiential learning, which allow them to adapt swiftly and effectively to the dynamic conditions of the game. This adaptability is critical in blackjack, where the ability to adjust strategies in response to the unfolding game state and the perceived strategies of opponents can significantly influence outcomes. Humans intuitively understand and react to subtle cues, such as an opponent's demeanor or betting patterns, which are indicators of potential strategy shifts. This capacity for nuanced interpretation and response is something that basic RL models, which typically operate on explicit numerical inputs and follow predefined algorithms, may not fully capture.

Furthermore, humans are adept at qualitative risk assessment, a skill deeply embedded in our decision-making processes. This involves more than simply calculating odds; it encompasses evaluating the situational context—like the stage of the game, the stakes, and the behavior of other players—and making decisions that balance potential gains against risks. This ability to manage trade-offs between aggressive plays (aiming for higher returns at higher risk) and conservative strategies (aiming for safer but possibly lower returns) is crucial in games like blackjack. While RL models can be trained to approximate these trade-offs, they often do so within the confines of statistical optimization without the broader context that human players consider.

To integrate these human-like capabilities into RL models, advancements could include the development of more sophisticated models that incorporate theories from psychology and behavioral economics. For example, RL could be enhanced by integrating models of human cognitive biases or emotional responses to create richer, more adaptive learning algorithms. These models would benefit from datasets enriched with annotations on player behavior and psychology, allowing them to train on and predict not just the statistically optimal action but also the psychologically strategic one. This approach would mark a significant step toward bridging the gap between human and machine capabilities in strategic games like blackjack.

Future Work

While Q-learning effectively develops strategy based on accumulated experience, it inherently lacks the real-time adaptability provided by counting, which directly assesses the current state of the deck (Thorpe, 1966). In future work, we can further explore the role of card counting systems in enhancing the decision-making strategies within Blackjack. Card counting is a strategy used to determine whether the next hand is likely to give a probable advantage to the player or the dealer. It can significantly shift the odds in favor of the player. By integrating various card counting techniques—such as the Hi-Lo system, the KO system, or more complex strategies like the Wong Halves—into our existing reinforcement learning models, we can analyze their impact on the overall effectiveness of the strategies developed.

It would also be interesting to simulate a series of games where these counting systems are implemented to assess their viability in real-time gaming scenarios. Nevertheless, we should also delve into the legal and ethical considerations of counting system to ensure that the approaches remain viable within the typical constraints of both online and physical casino environments. Furthermore, the stochastic nature of gambling games means that outcomes are substantially influenced by chance, limiting the efficacy of predictive machine learning models. The inherent randomness of card dealing and shuffling presents a challenge for machine learning tools to consistently gain an advantage (Brown Sandholm, 2017).

References

Brown, N., & Sandholm, T. (2017). Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science (New York, N.Y.)*, *359*(6374), 418–424. Retrieved from https://doi.org/10.1126/science.aao1733 doi: 10.1126/science.aao1733

Collins, A. G. E. (2019). Reinforcement learning: bringing together computation and cognition. *Current Opinion in Behavioral Sciences*, *29*, 63-68. Retrieved from https://www.sciencedirect.com/science/article/pii/S235215461830175X (Artificial Intelligence) doi: https://doi.org/10.1016/j.cobeha.2019.04.011

Governor of Poker. (2022). *What are the blackjack rules.* Governor of Poker Support. Retrieved from https://www.governorofpoker.com/support/?s=gameplay\&f=what-are-the-blackjack-rules\&l=en

Mojocolors. (2019). *900,000 hands of blackjack results.* Kaggle dataset. Retrieved from https://www.kaggle.com/datasets/mojocolors/900000-hands-of-blackjack-results/data

Richard S. Sutton, A. G. B. (2018). *Reinforcement learning, second edition: An introduction*. MIT Press. Retrieved from https://books.google.com/books?hl=en&lr=&id=uWV0DwAAQBAJ&oi=fnd&pg=PR7&ots=mjoJo1Y3l9&sig=-fhzqtS9eexHLm07_TLq_9RMoGQ#v=onepage&q&f=false

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.

Thorpe, E. O. (1966). *Beat the dealer: A winning strategy for the game of twenty-one*. New York: Vintage Books.