Monte Carlo Blackjack Simulation

Dragomir Daria Nicoleta, Jilavu Izabela Maria, Soare Alex Antonio

January 2025

Abstract

This document describes a Monte Carlo simulation applied to the game of Blackjack. The project aims to estimate optimal strategies and probabilities of winning under various conditions. The theoretical background, simulation design, results, and analysis are discussed in detail.

Contents

1 Introduction **2**
- 1.1 Game Rules 2

2 Monte Carlo Methods **3**

3 Monte Carlo Tree Search (MCTS) **3**
- 3.1 Overview of the Algorithm 3
- 3.2 Mathematical Formula 4
- 3.3 Explanation of the Formula 4
- 3.4 Visualization 4
- 3.5 Advantages of MCTS 4
- 3.6 Application to Blackjack 4

4 Implementation Details **5**
- 4.1 Simulation Initialization 5
- 4.2 Simulating a Game of Blackjack 5
- 4.3 Large-Scale Simulation 5

5 Results and Analysis **6**
- 5.1 Win Probabilities for Hit vs. Stand 6
- 5.2 Distribution of Player Scores 6

6 Theoretical Foundations **6**
- 6.1 Central Limit Theorem and Normal Distribution 6
  - 6.1.1 Relation to Normal Distribution 7
  - 6.1.2 Implications in Blackjack 8
- 6.2 Chebyshev's Inequality 8

6.3 Chernoff-Hoeffding Inequality 8
6.3.1 Application to Blackjack 8
6.3.2 Practical Insights 9

7 Discussion **9**

8 Conclusion **9**

9 References **9**

1 Introduction

Blackjack is a popular card game where players aim to beat the dealer by achieving a hand value as close as possible to 21 without exceeding it. This project employs Monte Carlo methods to simulate gameplay and derive insights into optimal strategies.

1.1 Game Rules

The game of Blackjack is played between a player (or players) and a dealer, following these rules:

- **Objective:** The goal is to have a hand value closer to 21 than the dealer's hand without exceeding 21.
- **Card Values:**
  - Number cards (2-10) are worth their face value.
  - Face cards (Jack, Queen, King) are worth 10 points each.
  - An Ace can be worth either 1 or 11 points, depending on which value benefits the hand most.
- **Gameplay:**
  1. Each player is dealt two cards face-up, while the dealer receives one card face-up and one card face-down (the "hole" card).
  2. Players may choose to:
     - **Hit:** Take another card to improve their hand value.
     - **Stand:** Keep their current hand value and end their turn.
     - **Double Down:** Double their initial bet and receive only one additional card (optional, based on house rules).
     - **Split:** If the first two cards have the same value, they may be split into two separate hands, each with its own bet (optional, based on house rules).
  3. The dealer must follow a fixed strategy:

  - The dealer must hit if their hand value is less than 17.
  - The dealer must stand if their hand value is 17 or higher.

- **Winning Conditions:**
  - A hand value exceeding 21 is a "bust," resulting in an automatic loss.
  - If the player's hand value is closer to 21 than the dealer's, the player wins.
  - A hand value of 21 with the first two cards (an Ace and a 10-value card) is called "Blackjack" and typically pays out 3:2.
  - If the dealer also has Blackjack, the result is a "push" (tie).

2 Monte Carlo Methods

Monte Carlo algorithms are stochastic simulation techniques used to solve complex problems through repeated random sampling. For this project, the simulation estimates probabilities based on:

- The player's score before a decision.
- The dealer's visible card (upcard).
- The player's action (hit or stand).

3 Monte Carlo Tree Search (MCTS)

Monte Carlo Tree Search (MCTS) is a heuristic search algorithm for decision-making problems. It is particularly effective in scenarios with large state spaces, such as board games or simulations like Blackjack.

3.1 Overview of the Algorithm

MCTS consists of four main steps:

1. **Selection:** Starting from the root node, recursively select child nodes using a selection policy until a leaf node is reached.
2. **Expansion:** If the leaf node is not terminal, add one or more child nodes to explore new states.
3. **Simulation:** Perform a random simulation (rollout) from the new node to a terminal state, recording the result.
4. **Backpropagation:** Propagate the simulation result back up the tree, updating the statistics for visited nodes.

3.2 Mathematical Formula

The key to MCTS is the Upper Confidence Bound for Trees (UCT), which balances exploration and exploitation:

$$UCT = \frac{w_i}{n_i} + c\sqrt{\frac{\ln N}{n_i}}$$

where:

- w_i: Total reward obtained from simulations passing through node i.
- n_i: Number of times node i has been visited.
- N: Total number of simulations from the root node.
- c: Exploration parameter, controlling the balance between exploration and exploitation.

3.3 Explanation of the Formula

The UCT formula combines two terms:

- The first term, $\frac{w_i}{n_i}$, represents the exploitation term, encouraging nodes with higher average rewards.
- The second term, $c\sqrt{\frac{\ln N}{n_i}}$, represents the exploration term, favoring nodes that have been visited less often relative to the total number of simulations.

This balance ensures that the algorithm explores promising nodes while still considering less-explored options.

3.4 Visualization

The figure below illustrates the Monte Carlo Tree Search process:

3.5 Advantages of MCTS

- Efficient handling of large state spaces without exhaustive search.
- Can adapt to various domains with minimal parameter tuning.
- As simulations increase, the algorithm converges to optimal strategies.

3.6 Application to Blackjack

In this project, MCTS is applied to simulate possible game states in Blackjack, enabling strategic decision-making by estimating the probabilities of winning for various actions (hit, stand, etc.). The algorithm ensures optimal decisions based on extensive simulations and statistical analysis.

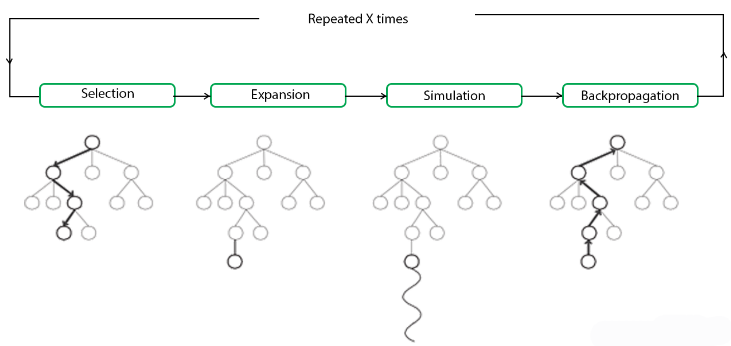

Figure 1: Monte Carlo Tree Search (MCTS) Process

4 Implementation Details

4.1 Simulation Initialization

The simulation uses data structures to store the number of wins and total simulations for all combinations of:

- Player's score: 2 to 21.
- Dealer's upcard: 1 to 10.
- Player's action: hit or stand.

4.2 Simulating a Game of Blackjack

A single game is simulated by:

1. Generating initial hands for the player and dealer.
2. Calculating scores and determining the player's action randomly.
3. Resolving the game based on Blackjack rules.
4. Recording results for probability calculation.

4.3 Large-Scale Simulation

The simulation runs 10^8 games to calculate probabilities for each parameter combination. Probabilities are derived using:

$$P = \frac{\text{Number of Wins}}{\text{Total Simulations}}$$

5 Results and Analysis

5.1 Win Probabilities for Hit vs. Stand

The graph in Figure 2 illustrates the win probabilities for hitting versus standing, categorized by the dealer's upcard. Each line represents a specific upcard and compares the probabilities for both actions.

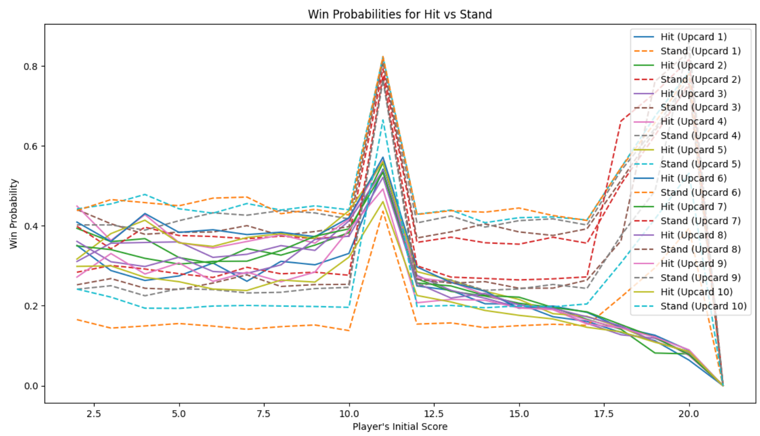

Figure 2: Win Probabilities for Hit vs. Stand

5.2 Distribution of Player Scores

The histogram in Figure **??** shows the distribution of player scores. A Gaussian fit is overlaid, demonstrating that the scores approximately follow a normal distribution.

6 Theoretical Foundations

6.1 Central Limit Theorem and Normal Distribution

The Central Limit Theorem (CLT) is a fundamental concept in probability and statistics. It states that, given a sufficiently large number of independent random variables, their normalized sum tends toward a normal distribution, regardless of the original distribution of the variables. Formally:

$$\frac{\sum_{i=1}^{n} X_i - n\mu}{\sqrt{n}\sigma} \xrightarrow{d} \mathcal{N}(0,1) \quad \text{as } n \to \infty$$

where:

- X_i: Independent and identically distributed random variables.
- μ: Mean of the random variables.
- σ: Standard deviation of the random variables.
- $\mathcal{N}(0, 1)$: Standard normal distribution with mean 0 and variance 1.

6.1.1 Relation to Normal Distribution

The CLT provides a bridge between the discrete world of random variables and the continuous world of the normal distribution. This relationship is crucial for understanding why many phenomena in nature and data science exhibit a bell-shaped curve. Some key insights are:

- The sum (or average) of a large number of independent random variables approximates a normal distribution, even if the original variables are not normally distributed.
- The approximation improves as the sample size n increases, making the normal distribution a universal tool for statistical inference.
- The normal distribution arises naturally as the limiting distribution for many stochastic processes due to the CLT.

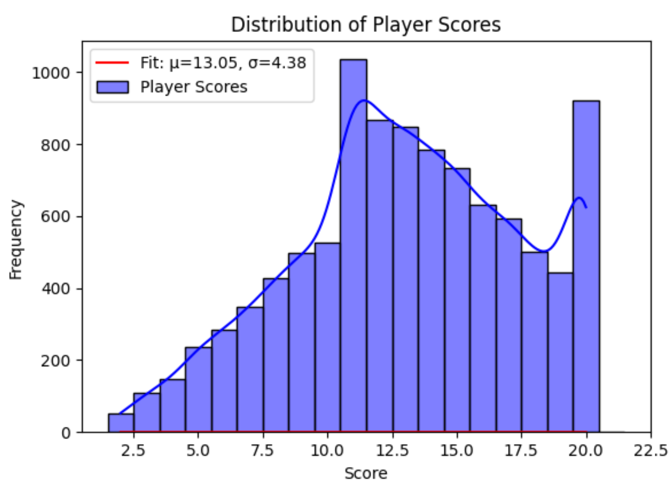

Figure 3: Distribution of Player Scores Approaching a Normal Distribution

6.1.2 Implications in Blackjack

In the context of this project, the CLT helps explain the distribution of player scores after numerous simulated games:

- As the number of simulated games increases, the distribution of scores approaches a normal distribution.
- This phenomenon is observed in the histogram of player scores (Figure 3), where the data closely matches a Gaussian curve.
- The normal distribution allows for analytical approximations and statistical predictions about gameplay outcomes.

6.2 Chebyshev's Inequality

Quantifies the probability of deviation from the expected value:

$$P(|\bar{X} - \mu| \geq \epsilon) \leq \frac{\sigma^2}{N\epsilon^2}$$

6.3 Chernoff-Hoeffding Inequality

The Chernoff-Hoeffding inequality provides a bound on the probability that the sum of independent random variables deviates significantly from its expected value. The inequality is expressed as:

$$P(|\hat{P}(A) - P(A)| \geq \epsilon) \leq 2\exp(-2N\epsilon^2)$$

where:

- $\hat{P}(A)$: Empirical probability (observed frequency).
- $P(A)$: True probability.
- N: Number of independent samples.
- ϵ: Margin of error.

This inequality shows that the probability of a large deviation decreases exponentially with the number of samples N. It is more precise than Chebyshev's inequality for bounded variables, such as Bernoulli variables ($X_i \in \{0, 1\}$).

6.3.1 Application to Blackjack

In the context of our Blackjack simulation, the Chernoff-Hoeffding inequality is used to determine the minimum number of simulations required to estimate the win probability within a desired error margin ϵ and confidence level L. We model the result of a Blackjack action (e.g., "hit" or "stand") using a Bernoulli random variable with probability p, since the outcome is binary: win (1) or loss (0).

$$X \sim \text{Bernoulli}(p), \quad P(X=1)=p, \quad P(X=0)=1-p$$

To determine the required number of simulations N, we set:

$$P(|\bar{X}-\mu| \geq \epsilon) \leq 2\exp(-2N\epsilon^2) \leq L$$

Taking the natural logarithm and solving for N, we get:

$$-2N\epsilon^2 \leq \ln\left(\frac{1-L}{2}\right) \quad \Rightarrow \quad N \geq \frac{\ln\left(\frac{2}{1-L}\right)}{2\epsilon^2}$$

For a margin of error $\epsilon = 0.05$ and confidence level $L = 0.98$ (98% confidence), we calculate:

$$N \geq \frac{\ln\left(\frac{2}{0.02}\right)}{2(0.05)^2} = \frac{\ln(100)}{0.005} = \frac{4.6}{0.005} = 921$$

Thus, at least 921 simulations are required per hand to achieve a margin of error of 0.05 with a 98% confidence level.

6.3.2 Practical Insights

This inequality ensures that the number of simulations grows logarithmically with the confidence level, making it an efficient tool for ensuring the accuracy of Monte Carlo simulations in Blackjack.

7 Discussion

The Monte Carlo method successfully estimates winning probabilities and highlights optimal strategies. The incorporation of MCTS offers a robust decision-making framework. Limitations include computational costs and reliance on assumptions about randomness.

8 Conclusion

This project demonstrates the power of Monte Carlo simulations in analyzing complex systems like Blackjack. Future work could involve incorporating more advanced strategies or exploring more scenarios.

9 References

- Monte Carlo Method - Wikipedia
- Monte Carlo Tree Search - Wikipedia
- Law of large numbers - Wikipedia
- Probability and Statistics Course