Final Project Report: AI Texas Hold'em Poker Strategy Evolution and Optimization

Abstract

This project explores the application of genetic algorithms to evolve and optimize strategies for Texas Hold'em Poker. We implemented a simulation where AI agents play against each other, with their strategies evolving over time based on their performance. The evolution process is driven by genetic algorithms, utilizing a fitness function based on agents' balance and games won. We tracked the policy changes of agents over time and analyzed how different strategies emerged and dominated the gameplay. Our results show significant shifts in strategy, particularly in the handling of strong hands like "AA", with agents evolving more aggressive betting patterns.

Introduction

Texas Hold'em is one of the most popular variants of poker, and its complexity makes it a suitable challenge for AI strategy optimization. The goal of this project is to develop AI agents that can adapt and optimize their poker strategies through simulated play, utilizing genetic algorithms. The evolutionary approach allows for the development of complex strategies without explicit programming, making it an ideal method for exploring optimal play in a stochastic environment.

Related Work

Previous research has demonstrated the effectiveness of reinforcement learning and genetic algorithms in optimizing poker strategies. This project builds on such work by focusing on the evolution of preflop and postflop strategies, using policy matrices that agents adapt over time. The project is distinct in its focus on tracking policy changes and analyzing how strategies evolve in response to competitive pressures.

Problem Statement and Methods

The main problem addressed in this project is the optimization of poker strategies using genetic algorithms. The agents' strategies are represented by policy matrices that dictate the probabilities of various actions (fold, check/call, raise) based on the game state. The genetic algorithm's fitness function evaluates agents based on their accumulated balance and the number of games won, encouraging strategies that maximize long-term success.

Implementation

Game Simulation: The Texas Hold'em game was implemented in Python, allowing multiple AI agents to play against each other. Each agent was initialized with a basic strategy, represented by a policy matrix.

Genetic Algorithm: The evolution of strategies was driven by genetic algorithms. Agents with higher balances and more games won were selected for reproduction, and their strategies were mutated and crossed over to create new agents.

Policy Tracking: The project tracked changes in policy matrices over time, particularly focusing on how the strategy for handling strong hands like "AA" evolved.

Experiments and Results

The simulation was run for 1000 games, with policy snapshots recorded every 100 games. The results, visualized in the provided plot, show how the probabilities of various actions changed over time for the hand "AA".

Early Evolution: Initially, the probability of folding and checking/calling was higher, indicating a conservative strategy among agents.

Mid Simulation: As the simulation progressed, agents began to favor more aggressive strategies, particularly increasing the probability of raising by 5% and 3%.

Late Simulation: Towards the end of the simulation, there was a notable shift towards aggressive raising, with the probability of folding decreasing significantly. This suggests that more successful strategies involved taking calculated risks with strong hands.

The analysis of the simulation data, provided in the CSV file, confirmed these trends across multiple agents, indicating that the genetic algorithm successfully evolved more optimal strategies over time.

Discussion and Conclusion

The results of this project demonstrate the potential of genetic algorithms to evolve complex strategies in stochastic environments like poker. The significant policy changes observed indicate that agents were able to adapt and optimize their strategies based on their performance outcomes. Future work could extend this project by incorporating more sophisticated evaluation functions or by applying the evolutionary framework to other complex games or decision-making scenarios.

Limitations

The project faced limitations in terms of computational resources and the simplicity of the initial strategy. The simulation might benefit from more extensive training, a larger pool of agents, and more sophisticated initial strategies to explore even more complex evolution patterns.