Artificial Intelligence for Texas Holdem Poker Utilizing Machine Learning Techniques

Isaac Lewis
Advised by Professor Brian Mak

Introduction

Texas Holdem Poker is a popular card game where players compete to form the best set of cards possible, while making bets on their own likelihood of winning. It is a ***stochastic, imperfect information*** game: the cards are dealt randomly, and the opponent’s cards are kept hidden. These factors make holdem poker a ***difficult problem for artificial intelligence*** (AI).

A strong poker AI needs the ability to form ***an accurate model of its opponent’s behavior***. This problem is known as ***opponent modeling***, and was the focus of our project. We attempted to solve this problem using ***machine learning***.

Objectives

- Implement a proof-of-concept pokerbot (***TeaBot***) that used a tried-and-tested algorithm, ***miximax***
- Implement an improved pokerbot (***CoffeeBot***) by ***applying machine learning to the problem of opponent modeling***

Methodology: Miximax Algorithm

Our bots implemented the ***miximax algorithm***, a generalization of ***the minimax algorithm***. *Minimax* assumes both players always choose the action that maximizes their potential gain. In holdem poker, we cannot be certain which action is optimal for our opponent, as we cannot see their cards. Therefore, *miximax* assumes ***the opponent follows a mixed strategy*** – their action choices are made probabilistically.

We therefore need to ***estimate the probability of an opponent taking a particular action*** in a given situation.

Methodology: Supervised Learning

Our approach was to train a ***supervised learning classifier*** on observations against a particular opponent. Supervised learning classifiers take a set of labeled training data, and try to find a function that maps the input features to the class labels.

We trained a classifier using ***past observations of an opponent for training data***, where each observed situation was labeled with the action the opponent took in that situation. We could then pass a new situation to our classifier, which would output a ***probability estimate of the opponent's next action*** in that situation. We experimented with using ***naïve Bayes classifiers*** and ***neural networks*** for this task.

Results

We compared *TeaBot* and *CoffeeBot* with ***RulesBot***, a bot we implemented for testing purposes. *RulesBot* follows a ***rules-based strategy*** that considers its own cards and the current betting round. To reduce variance, each pair of opponents played 2 matches, each match consisting of 10,000 hands of poker.

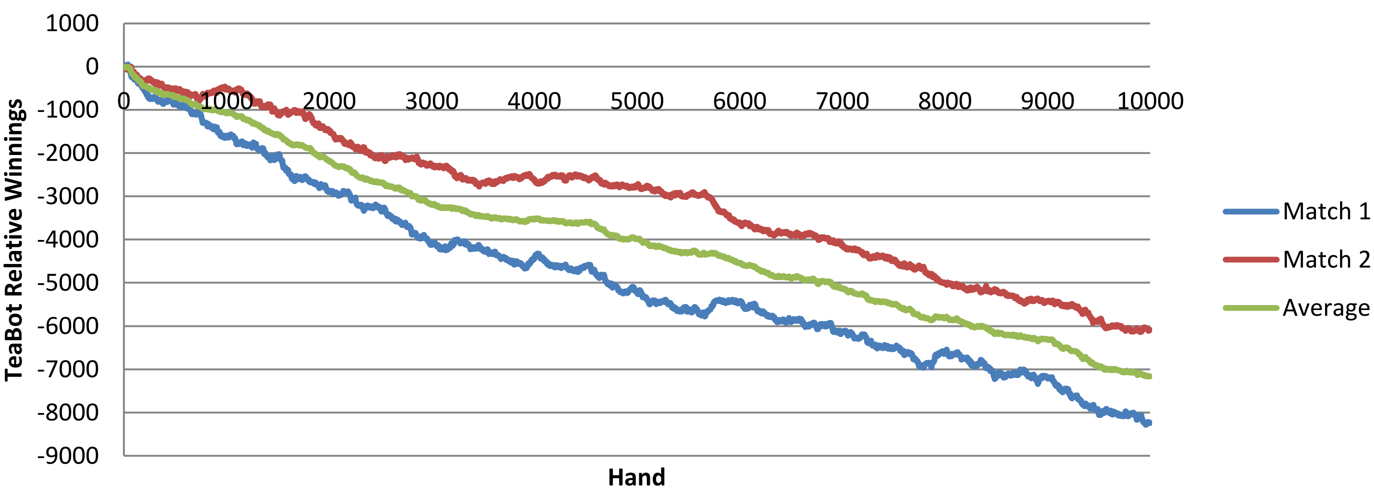

Figure 1. TeaBot vs RulesBot

TeaBot loses to RulesBot by a large margin, despite the fact that RulesBot follows a relatively simple strategy. This is because TeaBot ***lacks a strong opponent modeling function***.

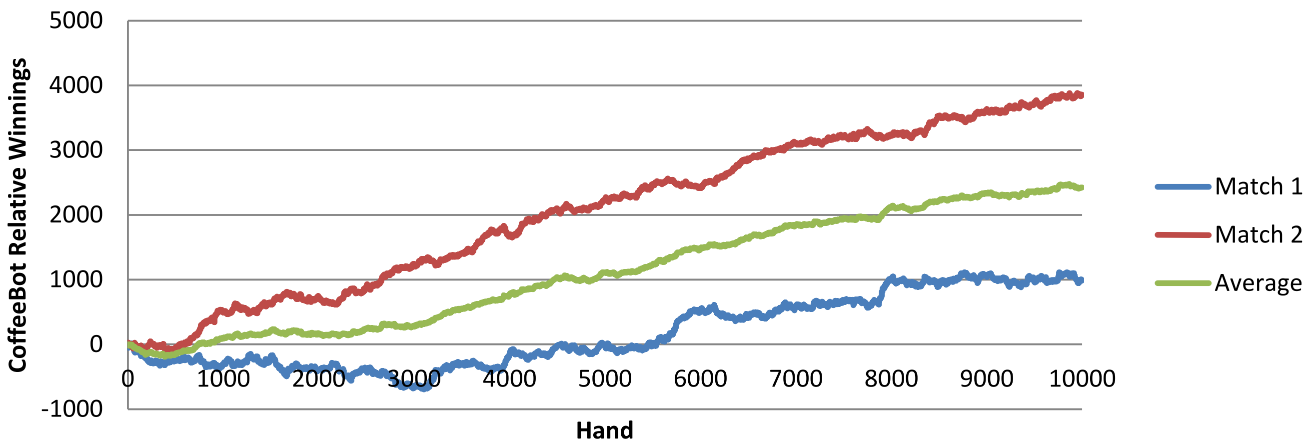

Figure 2. CoffeeBot (using a naïve Bayes classifier) vs RulesBot

CoffeeBot, on the other hand, ***defeats RulesBot***, showing that our approach of applying supervised learning to opponent modeling ***led to improved performance***.

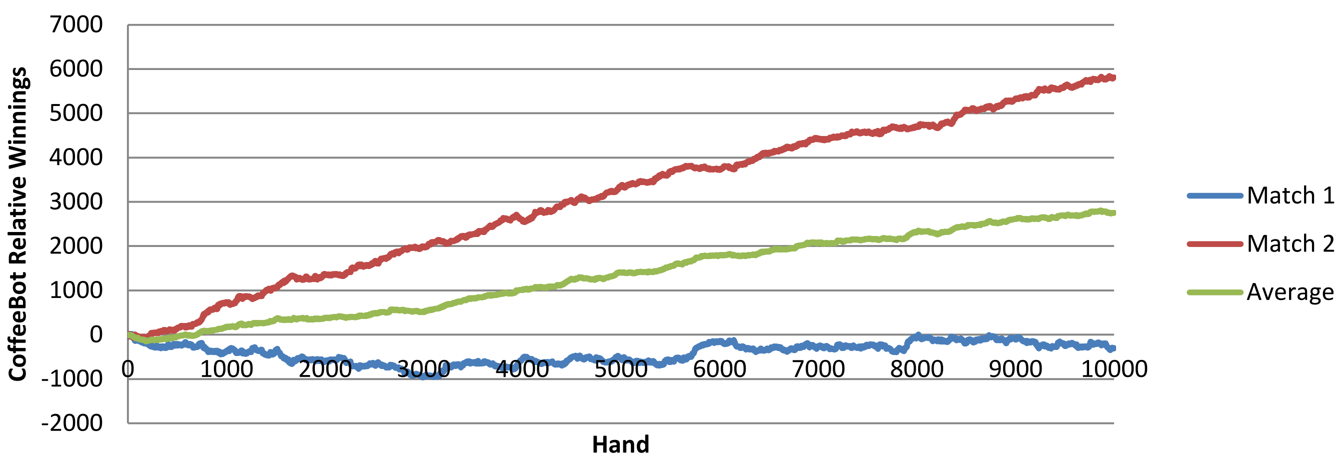

Figure 3. CoffeeBot (using a neural network classifier) vs RulesBot

Neural networks gave ***slightly better overall performance*** than naïve Bayes classifiers, but with ***more inconsistency*** between the two matches.

Conclusions

- The *miximax* algorithm gives ***weak performance*** when not accompanied by a strong opponent modeling function
- Our approach of using machine learning for opponent modeling was successful, and ***led to improved performance***
- ***Neural networks slightly outperform naïve Bayes*** classifiers, but are more inconsistent