Project Proposal
CS 221
Prof. Liang

Ryan Worley
Teammates: None
Due Date: 10/25/2018

Introduction

In 1962, Edward Thorpe, published his book, Beat the Dealer, summarizing a system that could be used to gain advantage for the player in the game of blackjack. Since then, casino-goers have been scheming, trying to implement this strategy in order to make easy money in this supposed game of chance. The most well known success story associated with card counting stems from the notorious MIT blackjack committee, which is rumored to have won more than $500,000 dollars in one night counting cards as a team. While counting cards sounds like an enticing strategy, its not an easy thing to pull off. The ability to keep track of the card count and manage the current state of the game is no easy task, and one that only the most cool-headed players could accomplish.

The process of counting cards is as follows:

- For any card with a value of 6 or under, add one to the count.
- For any value 7-9, keep the count the same.
- For any value of 10 or higher, decrease the count by one.

Larger count values means that there are more high cards in the deck, which gives the player an advantage over the dealer.

Purpose

The purpose of the project will be to create an engine that learns the optimum policy for playing blackjack while using a card counting strategy. At the end of the learning process, the user will be able to enter a starting state (card count, player cards, dealer card shown) and be returned with the optimum policy and expected return by which to play with, even if the optimum policy is to get up and leave the game. If the program works correctly, the user should generally see positive returns while playing. If there is enough time, a graphical interface will be created to display all of these values for the user for ease of access.

Preliminary Data

Data sets used to test this algorithm are small and will consist of deck layout inputs. One instance of data would include how many of each card are in the deck. From this data, we can create code that will generate a randomized order of these cards (shuffling) and create an associated true deck count associated with the permutation of cards from the data set. From this data set, we can play various game of blackjack, and come up with an optimum policy for playing a game at a certain state (card count) through a reinforcement learning algorithm.

Baseline and Oracle (Bridging the Gap):

A baseline algorithm was created that play blackjack without incorporating any card counting strategies or letting the player split cards. To do this, a blackjack class was created in python, and a value iteration was run on the problem to compute the optimum policy of a single round with a full 52 card deck. We found that the expected reward value was negative -6% of the bet size. However, I am getting mixed results with this algorithm, and need to look more closely at the value iteration and process and how the state updating process is handled, especially in relation to the probability of the dealers cards. This shows that when the player plays only one game against the dealer with a full fifty-two card deck, the odds are against them, and a negative return is expected. I believe that I can produce much greater returns than this and develop a strategy that allows the player to actually gain money in the game process.

To find an oracle (upper bound) associated with this project, some background research was conducted. In many studies, card counting does increase your expected value. From a non-research perspective, individual claim to have won over $100,000 dollars a year playing advantage blackjack. One team out of Seattle even claim that they won over 3.5 million in just a couple of years. While these total value numbers are amazing, it doesnt give us a good estimate of the expected value of the game, since we dont know how much was invested in the game. However, there is documentation of card counting providing a 1.2% advantage to the player, with a standard deviation of 3.5 (very high). We see here that there is a significant gap that can be bridged between my baseline and oracle, and the project has the opportunity to make its way into positive reward.

Challenges

Card counting dramatically increases the number of states associated with the MDP. Add this in with incorporating the full game of blackjack with doubling down, splitting cards and this problem will get a lot tougher to implement. Defining the state for the MDP will be a difficult decision and keeping only the necessary information will be important in order to minimize our state space and keep the problem efficient.

Additionally, reinforcement learning will be implemented, since value iteration isnt totally realistic showing of the game. In a real game of blackjack, the player will have knowledge of the count (the state), but no knowledge of the exact probabilities of the reward associated with each action. While the count gives the player a general knowledge of whether a high or low card is drawn, it does not tell the player exactly what cards are left in the deck. Creating sufficient features for the reinforcement learning algorithm to converge will also be a challenge itself.

Goals for Next Submission

For the next submission, I want to minimize the state space associated with this problem to keep the MDP efficient. I also want to work towards creating the MDP blackjack process, complete with doubling down, splitting, and keeping track of the count. MDP should also be modified to incorporate variable betting strategies. These strategies would allow for the expected returns to be greater, as a player could adjust his bet based on the likelihood of winning in a specific state.

References

[1] Jensen, Kamron, *The Expected Value of an Advantage Blackjack player*. All Graduate Plan B and other Reports. 524

[2] Wong, Stanford 1994. *Professional Blackjack*, page 31, 1994 ed.