CS182 Final Project Proposal

Amy Kang, George Zhang

October 28, 2016

1 Problem

Blackjack is a popular casino game with a very simple premise: given two cards out of a standard card deck and knowledge about one of the dealer's cards, act optimally to beat the dealer without busting. The rules of the game:

1. The player must decide whether to "stand" (to not take another card) or to "hit" (take another card) with the goal of getting as close to, but not exceeding, a hand value of 21.
2. The player can continue to hit until he decides to stand, or until his hand value exceeds 21, at which point he goes "bust," losing the game.
3. The dealer then turns over her face-down card, and if the card is less than 17, must hit. Otherwise, she stands. If the dealer busts, she loses.
4. If the player's hand exceeds the dealer's, the dealer pays the player an amount equal to his bet. If equal, the player keeps his bet. If less, the player loses his bet.

Despite its simplicity, however, it is difficult to win consistently due to the stochastic nature of the game. We would like to investigate whether a reinforcement learning algorithm, such as Q-learning, can learn to play the game and win on a consistent basis. Here, we will define "win" to mean monetary value, and not number of hands won.

2 System Details

We intend to build a blackjack simulator on which to train our agent. We believe that due to the simple nature of the game (i.e. only requirements is a deck of cards), we should be able to build a platform for playing blackjack fairly quickly. Initially, we will restrict a user's actions to hitting and staying, but may expand the options (doubling down, splitting, etc.) as we progress. After doing this, we can then build the agents needed to train and play the game. We will likely build a random agent (i.e. "stands" and "hits" at random) as a benchmark, and then implement other algorithms and strategies aimed at winning. We will then be able to compare the performances of all the agents to see which performed the best.

3 Topics Covered

With this project, we'll be drawing on concepts from reinforcement learning and MDPs. Blackjack can easily be modeled as an MDP. The set of actions are predetermined by the game, and the transition function is determined by the set of cards remaining in the deck. One of our tasks, then, will be determining the appropriate state space. One example state space which we intend to implement first looks only at the value of the cards for the player and the value of the dealer's face-up card. This simple definition of a state produces state space of roughly 21x11, which is very manageable from a Q learning perspective. The rewards for this game is also built into blackjack itself, due to its system of betting.

We believe reinforcement learning to be the best way to approach this game. We intend to explore different sets of state spaces, potentially including additional information about the game and various strategies (card-counting, for example). As the state space grows, we believe a RL approach will find a good solution more effectively than other approaches (such as state search, for example).

4 Algorithms

As stated above, we intend to use Q learning on a simple state space to begin with. We may also consider taking a Monte Carlo approach and allowing the hand to play out before updating all the values to more quickly propagate information back to lower values that the user has (i.e. that hitting on a 9 produces high future utility). We will then expand our analysis to different state spaces to see if we can produce a more effective strategy using additional information about the game, such as which cards have already been seen (which can be approximated using various card counting approaches). Finally, we will want to introduce the idea of betting different amounts, and see if the agent can learn which circumstances are ideal for higher betting. We expect that learning how to bet efficiently will greatly increase the probability that the agent will win more than it will lose.

We also want to investigate various exploration / exploitation policies. We will likely start with epsilon-greedy, but we expect that tuning the parameters of epsilon-greedy and looking at other selection policies may allow the agent to learn better in a larger state space.

5 Expected Behavior

We expect that the basic Q learning agent will learn a policy that should do better than the random agent. We also expect that increasing the state space to provide more information will provide more specific scenarios, allowing the agent to make more informed decisions. We hope that by giving the agent more control over its betting amount will also lead it to win more often than not.

6 Issues

One issue we want to avoid is the curse of dimensionality. If we start giving the agent too much information, we risk having such a large state space that learning on it will be difficult. We chose blackjack because it can be represented in a relatively small state, but improving on the performance may expand the state space. We will have to consider what the ideal tradeoff will be, and whether we should experiment with something like approximate Q-learning and look at specific features of the game.

We also expect that introducing betting options will be tricky. It makes little sense to add them to the state space and learn the results, as this just expands the state space without being particularly effective in terms of learning, since each betting amount will likely have the same learning trajectory, just with different Q values in them due to the different betting values. We instead may want to learn a function that takes certain features of the current game and produces an optimal bet.

7 Resources

We will, of course, use the textbooks from class (Sutton and Barto, AIMA). We may also look into additional papers with the goal of finding various strategies that work in blackjack and previous research done in this area. The following are examples of the resources we plan on using:

- A. Perez-Uribe and E. Sanchez, "Blackjack as a Test Bed for Learning Strategies in Neural Networks," *1998 IEEE International Joint Conference on Neural Networks Proceedings*, vol 3, p. 2022-2027, 1998.
- D. Abel, A. Agarwal, F. Diaz, A. Krishnamurthy, R. Schapire, "Exploratory Gradient Boosting for Reinforcement Learning in Complex Domains," *Cornell University Library*, 2016.
- M. Schiller and F. Gobet, "A Comparison between Cognitive and AI Models of Blackjack Strategy Learning," *KI 2012: Advances in Artificial Intelligence: 35th Annual German Conference on AI, Saarbrücken, Germany, September 24-27, 2012. Proceedings*, vol 7526, p. 143-155, 2012.