Thomas Lenz

Matthew Pivoney

Fidel Ramirez

Texas Holdem Project Proposal (Unrevised)

Texas hold em’ is a variant of poker, a game of chance with limited information given at any time. Each player has two private cards in their hand, or their hole cards. In addition, the game can be split up into several sections: first, the game can be split up into its two most major stages: preflop and postflop. The “flop” is the first group of cards to be revealed in the community pool. The term “preflop” describes the stage of the game where the agent has the least information. In this stage, the only information available to the agent is the cards in its hand. These cards can be looked at by the agent at any time during the game. After the hole cards are dealt, there’s only one round of betting before the flop is revealed. Because of a limited amount of information, any bets made here tend to be tenative at best, and no success is guarenteed. Everything that occurs after the flop is added to the pool is part of the postflop stage. Since the flop is where the largest amount of objective information is gained, a heavy amount of pruning will take place, and the possibility of analyzing betting habits will become exponentially more effective postflop. Postflop, the game can be broken down further. After the flop is revealed, another round of betting occurs. Subsequently, the fourth card in the pool, known as “the turn”, is revealed, followed by another round of bets. Once all players are even, the river, or the final of the five cards, is dealt into the pool. The final round of betting finally occurs, followed by the showdown, where all the players’ hole cards are revealed, and the best hand is found for each. At each of these stages, save for the showdown, an agent has to make decisions based off of prior experiences or based off of what it has been taught previously in order to maximize its winnings while minimizing its losses during any hand. By applying mathematics we can decided when it is best to take any of our actions.

The first thing we have to do is identify the environment and all the properties of it. The first property is that this is partially visible, since while the agent is not able to see what other agents have, it can see its own hand, as well as the community pool. The next property is that this is a multi-agent environment. Specifically, this is a competitive environment in a zero sum game as we are learning right now, and there are differences in how agents should behave according to each other. This leads us into considering game theory under the assumption that our opponent is rational as well. Since we are working with probability, the environment that we are working in is going to stochastic. This means that even if we are rational and make no mistakes in our decision making based on statistics and probability we may still lose the goal though is to win more than we lose. The environment is also episodic, since the agent should be making decisions based off of the current state of the board, its hand, and the pot. While an argument could be made that the amount of money invested into the pot should, under certain circumstances, affect what the agent does in response to a bet, it’s hard to justify explicitly programming in the lost-cost fallacy. The environment can also be considered to be Dynamic, since the way we act gives the opponent information about what we have, and the way the other agents behave could give our agent information on how it behaves. This can be used in a few ways meaning that we can bluff the opponent if we know that they tend to be passive, though this may be too complex for us to handle since that includes a lot more factors. Since we have already broken the game into several phases the environment is then discrete.

As a result of the environment, there are several issues that will needed to be worked around or solved. The first of which being that are working with limited information. The fact that we will not be able to fully see the game will mean that there will be a certain amount of uncertainty no matter what we do. However, that can be somewhat reduced by using statistics and making some assumptions. Another way to make decisions is to perhaps somehow teach the agent. This can be done in a few different ways, but currently we are undecided how we would go about it and how we would implement it. There was a study done by that actually used both concepts mentioned previously, where they taught an AI how to make choices depending on how the game state is. In that study, they used previously recorded results of games in order to teach the AI. 1 They were able to do this by clustering according to certain attributes that were recorded. After being trained long enough, for the right amount as well, when given a state, the AI was able to make choices based off its previous knowledge. The example given in the study, however; was only concerned with pre-flop choices, not for the whole hand. That being said, that could be further expanded. This leads us to another problem: actually learning how to implement an AI which can learn from information that we would already have somehow. Thankfully, there are sources that are available to help us learn in depth of how such an AI would learn and how to teach it, one source being the book that is assigned to this course.

To conclude, after examining the environment that we could be working in, this should be a rather challenging project to undertake. However, this is nothing that has not been done already, and there are more sources that exist covering the same topic and all of those can still be referenced if need be. In addition, most of what this seems to be is statistics, probability, and data mining, none of which is too difficult to learn on our own. While we may not cover it in class, we should be confident enough to tackle the problem. If we do find issues relatively early on the project, though, we may choose to somehow alter the environment. One suggested bit of feedback that we already recieved was to perhaps make the environment fully visible (let us see what the agent has in their hand). This would be a huge alteration to the project and would remove a significant amount of uncertainty in the game, giving more consistent results for a proof-of-concept. There probably are other options or routes that we could do in order to perhaps make the project more manageable as well, but for now, we're going to be working in the given environment and attempt to make an AI which would have learned from prior games, which the method of how we would attempt to having already been explained.

1.