Design for the Three Player Limit Texas Hold’em Bot

Ben Kreuter
Computer Science Dept
U. Virginia

Ben Rodes
Computer Science Dept
U. Virginia

Samee Zahur
Computer Science Dept
U. Virginia

Colin Braley
Computer Science Dept
U. Virginia

April 14, 2011

1 High Level Design

For our final project we have joined the teams of Ben K. and Ben R. with the team of Samee and Colin. Ideally we would combine both our strategies from the West Virginia Hold’em project to create a three player limit Texas hold’em bot, however both of our previous bots were near nash equilibrium approaches. Instead, our approach for the final submission is to combine the counterfactual regret minimization used by Samee and Colin for their West Virginia Hold’em submission with a Monte Carlo tree search (MCTS) algorithm.

We were attracted to MCTS given the lack poker bots taking this approach and the success MCTS has had in the game GO. MCTS for poker incrementally builds a miximax game tree where the expected value at each node is the result of Monte Carlo simulation. MCTS works by using a selection algorithm to find nodes of the partial game tree to expand, taking into account exploration vs. exploitation. Once a node is selected, a new node is added to the tree and a Monte Carlo simulation is performed starting at the newly added node to the end of the game where the result is then stored in the new node and backpropagated up the tree. After a number of rounds building the game tree, given the current game state (a node in the tree), a maximizing move is selected based on the simulated expected values.

To simulate the game we must know something about our opponents to predict their future betting behaviour, which also affects the distribution of cards in the deck for any remaining community cards we must also simulate. Initially, we have no model of our opponent to base a simulation so our approach is to to start with a CFR generated near nash equilibrium as our opponent models and deviate from these to a more accurate model of our opponents as data becomes available.

2 Implementation

There are still some implementation details to work out, involving how exactly will we model opponents and combine the CFR data with the MCTS tree. We also have yet to choose what selection and back propagation MCTS algorithms we will use, and if and how we will periodically
update the MCTS tree to reflect changes in opponent strategy. We must also determine when it is ideal to deviate from the near nash equilibrium, however this also based on what our opponent models will be.

3 Initial Submission

Our initial submission is a simplified CFR bot. Ideally a CFR strategy would be to evaluate the player’s hole cards and select a tree to traverse, where each node contains a probability distribution of actions to make based on the bucket the hole cards fall in to. Our current submission buckets all hole cards into the same category.