Monte-Carlo Tree Search in Poker using Expected Reward Distributions*

Guy Van den Broeck [a] Kurt Driessens [a] Jan Ramon [a]

[a] *Dept. of Computer Science, K.U.Leuven*

1 Introduction

Poker playing computer bots can be divided into two categories. There are the *game-theoretic bots*, that play according to a strategy that gives rise to a Nash equilibrium. These bots are impossible to beat, but are also not able to exploit non-optimalities in their opponents. The other type of bot is the *exploiting bot* that employs game tree search and opponent modeling techniques to discover and exploit weaknesses of opponents. In this paper, we focus on the second type of bot.

Research in computer Poker has been mainly dealing with the limit variant of Texas Holdem. In limit Poker, bet sizes are fixed as well as the total number of times a player can raise the size of the pot. In fact, all exploiting bots developed so far, deal with the heads-up limit version of the game. We investigate the use of Monte-Carlo Tree Search (MCTS) in multiplayer no-limit Texas Hold'em Poker. To extend to use of MCTS to incomplete information games, we propose new backpropagation and selection strategies that take into account the consequences of having non-deterministic nodes in the search tree.

2 The Poker Game Tree

A game tree is used to reason about future states of the game. The expected reward in each state is computed through a set of backpropagation rules. Because cards are dealt randomly, the basic MINIMAX game tree has to be extended with chance nodes to deal with the nondeterminism in the game.

It is impossible to make claims about the optimal action for an opponent without knowing what cards he is holding. Therefore, we need an opponent model to predict the behaviour of the opponents. The hidden information in Poker thus results in a so called MIXIMAX game trees [1] where opponent decision nodes have to be modeled as chance nodes. The model that we use is a set of M5P regression trees [5] that predict the probability of each possible action in a given game state and the rank of each player's hand at showdown. We learn the model using the Weka toolkit [6] from features of the game state, such as the number of raises, individual player statistics and the pot odds.

3 Monte-Carlo Tree Search

The game tree for 2 player no-limit holdem Poker consists of an estimated 10^{71} nodes [3]. Traversing the entire tree is impossible and an incomplete search procedure is required. We employ Monte-Carlo Tree Search [2] which is a best-first search strategy that revolutionized research in computer-Go.

MCTS incrementally builds a subtree of the entire game tree in memory. For each node, standard MCTS stores the expected value of the reward of that node together with a counter that stores the number of sampled games that gave rise to the estimate. The algorithms starts with only the root of the tree and repeats the following 4 steps until it runs out of computation time: **(1) Selection:** Starting from the root, the algorithm selects in each stored node the branch it wants to explore further until it reaches a stored leaf. This is not necessarily a leaf of the full game tree. **(2) Expansion:** One (or more) leafs are added to the tree as child(ren) of the leaf reached in the previous step. **(3) Simulation** A sample game starting from the added

*A longer version of this paper will appear at the 1st Asian Conference on Machine Learning (ACML), Nanjing, China, 2009

leaf is played until conclusion. **(4) Backpropagation** The estimates of the expected values and the counters on the explored path are updated with the new information. Finally, an action-selection strategy chooses a good action to be executed based on the stored values in each of the root's children.

4 Expected Reward Distributions

Because Poker is a nondeterministic incomplete information game, we are faced with the problem that the propagated rewards will never converge to a single value, but instead are drawn from a probability distribution. Existing strategies were not designed with this type of game in mind and, as a consequence, fail to exploit this fact. In the worst case, they continuously sample from uninformative branches of the game tree. We propose a modification of the standard MCTS selection and backpropagation strategies that explicitly model and exploit the uncertainty of sampled expected values. This improves the performance of MCTS in nondeterministic games or games with hidden information.

First, we can estimate the expected value and variance of the rewards obtained in the simulation step. From these values, we build the expected reward distribution. This distribution models the uncertainty of the expected reward of an action. In contrast to the reward distribution, the expected reward distribution does converge to a single value when more samples are taken.

Second, better sample selection strategies can be used that fully exploit the information in the expected reward distributions. We propose the UCT+ strategy that is based on the existing UCT strategy [4]. It samples more in regions of the game tree that both have an uncertain estimate of the expected reward and an expected reward that is high enough to be worth considering, thereby striking a balance between exploration and exploitation.

5 Results

The new strategies are evaluated as a part of a complete Poker bot that is, to the best of our knowledge, the first exploiting no-limit Texas Hold'em bot that can play at a reasonable level in games of more than two players. The experiments show that the MCTS-based approach enables strong exploitative behaviour against weaker rule-based opponents. Preliminary experiments show that the bot can hold its own when playing against experienced humans. In a number of experimental evaluations, we studied some of the conditions that allow the proposed selection and backpropagation strategies to result in a measurable advantage over existing strategies. We discovered that an increase in available computation time translates in additional profit for the new approach. We expect a similar trend as the complexity of the opponent model surpasses the complexity of the backpropagation algorithm.

References

[1] Darse Billings. *Algorithms and assessment in computer poker*. PhD thesis, Edmonton, Alta., Canada, 2006.

[2] G.M.J. Chaslot, M.H.M. Winands, H. Herik, J. Uiterwijk, and B. Bouzy. Progressive strategies for Monte-Carlo tree search. *New Mathematics and Natural Computation*, 4(3):343, 2008.

[3] A. Gilpin, T. Sandholm, and T.B. Sørensen. A heads-up no-limit Texas Hold'em poker player: discretized betting models and automatically generated equilibrium-finding programs. In *Proceedings of the 7th international joint conference on Autonomous Agents and Multiagent Systems*, volume 2, pages 911–918. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, 2008.

[4] L. Kocsis and C. Szepesvari. Bandit based Monte-Carlo planning. *Lecture Notes in Computer Science*, 4212:282, 2006.

[5] Y. Wang and I.H. Witten. Induction of model trees for predicting continuous classes. 1996.

[6] H. Witten Ian and F. Eibe. Data Mining: Practical machine learning tools and techniques. *Morgan Kaufmann, San Francisco*, 2005.