

AiPoker - Implementation of a Poker Agent

John Hollén, Robin Berntsson, Simon Bergström

Abstract— We will probably write this section the last thing we do.

Index Terms—Monte Carlo Tree Search, Texas Hold'em

1 BACKGROUND

Texas Hold'em is a popular variant of poker, not just to play with friends but also online against other people for money. Poker belongs with the most difficult kind of card game to solve discretely since it is an stochastic game with imperfect information. The challenge of creating the best poker agent has attracted a lot of people and is a popular subject in AI development.

The theory behind Monte Carlo search trees will be studied within this project and tested as a part of the logic that will determine the actions of the poker agent. Smaller algorithms to improve the logic for the AI will be tested and implemented in combination of the Monte Carlo Search Tree to create the best possible poker agent.

2 THEORY

2.1 Characteristics of a Poker Player

Like previously described poker is a stochastic game, that makes it very hard to control and to define what a good poker player is. A good poker player does not always need to be the one who is a winning player. But in this project a aim has been set to try to create a poker agent who wants to win and will not take too much risk when playing. But to not make the agent to predictable the agent shall randomly switch behavior between aggressive and passive playing.

3 METHOD

3.1 Basic Game Engine

In order to even begin implementing an AI, the core poker game had to be implemented first. For simplicity the game implemented in this study is a two player game where a human player faces a computer AI. The player who starts is chosen at random when the game is started. The player that starts then puts one dollar in to the pot. The other player puts two dollars in the pot. Then the game is turn based like ordinary Texas Hold'em, the players can bet, check, call, raise and fold. And after each betting round cards are dealt to the table and the players combine the cards on hand with the cards on the table in order to get the strongest hand possible.

To be able to test the game engine the computer was first set to do moves completely at random. When all this was done the implementation of the AI could start.

3.2 AI - techniques

3.2.1 Monte Carlo Tree Search

Monte Carlo Tree Search (MCTS) is just like an algorithm like minimax based on a tree structure. By simulating a number of different outcomes a search tree is built based on the outcome of the simulation. More specifically the algorithm is made in four different stages which are repeated numerous times when the algorithm is being run.

MCTS only starts with the root node of the tree and incrementally build the tree by repeating the following 4 steps. One of the pros with MCTS is that it is anytime, that is it will give a valid solution to the problem even if it is interrupted before it ends. However the quality of the solution is an estimate and is expected to be better the more time

the MCTS keeps running.

Selection

Selection is made by starting from the root node and then recursively select nodes until a leaf node is reached (this does not have to be a leaf of the game tree). To determine which node is the optimal one, the following upper bounds confidence formula is used:

$$v_i + C \cdot \sqrt{\frac{\ln N}{n_i}} \quad (1)$$

4 RESULT

5 CONCLUSION

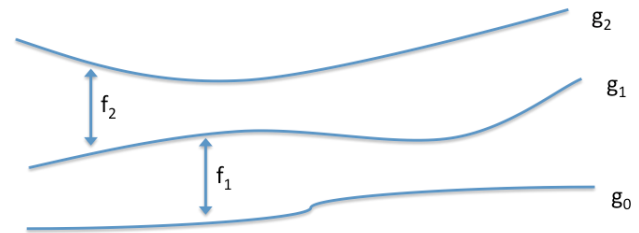


Fig. 1. En bild för att se hur man laggar till bilder.

REFERENCES

- [1] Byron L., Wattenberg M. *Stacked graphs - Geometry & Aesthetics*. IEEE Transactions on Visualization and Computer Graphics archive Volume 14 Issue 6, Pages 1245-1252. IEEE Educational Activities Department Piscataway, NJ, USA ISSN: 1077-2626. 2008.
- [2] Reynolds A. P., Richards G., Rayward-Smith V. J. *The Application of K-medoids and PAM to the Clustering of Rules*. In: Intelligent Data Engineering and Automated Learning - IDEAL 2004. Lecture Notes in Computer Science, 3177 . Springer-Verlag, pp. 173-178. ISBN 978-3-540-22881-3, 2004.