Card Game AI Comparisons

Monte Carlo Tree Search vs. Heuristics Methods

Greg Daniels CS5600-Intelligent Systems

Description

- This project is an implementation and comparison of game playing algorithms for an Italian card game called Scopa. It is a two player game in which players attempt to gain cards from the center board. If unable to gain a card, a player must discard a card to the center board. The game is played in rounds of 6 hands of 3 cards per hand. At least 1 point is awarded at the end of every round, but as many as 22 points are theoretically possible in every hand.
- Rules of the game
  - Players alternate capturing cards from the center or discarding to the center.
  - In order to capture cards from the center of the board a player must find cards (1 or more) in the center that sum to the same value as 1 card in the players hand. These cards then go into the players discard pile used to determine the score at the end of the game
  - If a player takes all of the cards from the center this is a “Scopa” (translated as sweep) and is worth 1 bonus point. The other player must then discard one of his cards on his next turn.
  - Four points are assigned as the end of every game (Ties result in no point being awarded)
    - 1 to the player with the most cards.
    - 1 to the player with the most sevens.
    - 1 to the player with the sette bello (seven of the money suit)
    - 1 to the player with the most of the money suit.
- Monte Carlo Tree Search is an algorithm that has recently been applied to many games where the search space is large. It works by sampling the search space rating each sample and returning the move that yielded the best sample.
- The main area of study is a Compare the results between a Monte Carlo Tree Search and Heuristics for playing the game.

Algorithms

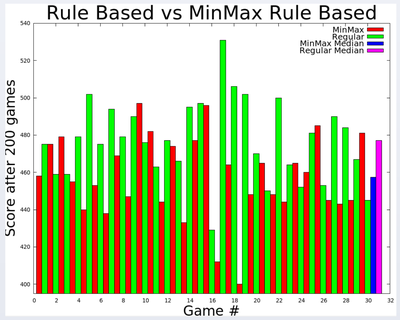
- MCTS with Random Playouts
  - This method will perform rollouts on every move available to the Player. Each move of the rollout will be made randomly. This will yeild a sampling of the seach space and an approximation of the quality of the move.
  - Two variations were implemented
    - The value returned is the number of points scored by the player at the end of the game
    - The value returned is the difference in the points of the two players
- MCTS with Rule Based Playouts
  - This algorithm will use the same rules as the heuristic player to sample the search space. Because of the overhead of applying the heuristics it wont be able to take as big of a sample as the random MCTS but the sample should be of a higher quality.
- Rule Based (Heuristic)
  - This Algorithm will work on a simple set of heuristics. The basic idea of these heuristics is to assign a value to each possible move based upon the values of the cards that the player will capture with that move. For example if a player can has the option of capturing a 1 with a 1 from his hand or capturing two 1s and a 5 with a seven. The second move will be chosen because he gains 4 cards and a seven. The suit of the cards also comes into play here as well.
- Rule Based With MiniMax End Game
  - This will be implemented exactly the same way as the Rule based player except that for the last hand complete game knowledge is available. Every card to this point has been revealed and we know exactly is in the other players hand and thus we have all the necessary information available to implement a Min Max Algorithm for the end game. This should yield slightly better performance that the rule based player.

Methods

All algorithms were tested against the Rule Based or Heuristic algorithm and were measured based upon the score after 200 games. A sample size of 30 was taken for every trial.

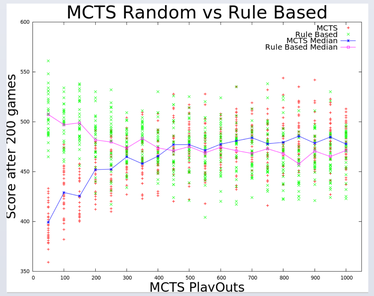
The MCTS algorithms were tested by comparing their performance based on the number of playouts performed to determine the best move

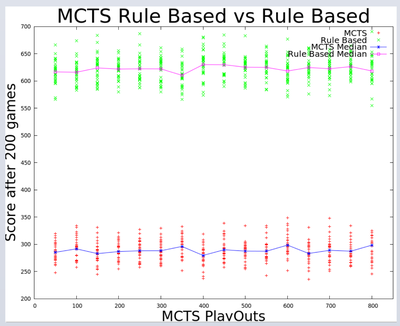
Results



Left are the results from running the Rule Based algorithm against the MinMax Endgame algorithm. Opposite from expected the Rule Based outperformed the one with min max endgame support by an average of 20.4 points. A T-test of the results gives a p value of .00635 which shows that the RuleBased Algorithm should continue to outperform the MinMax if more samples were taken.

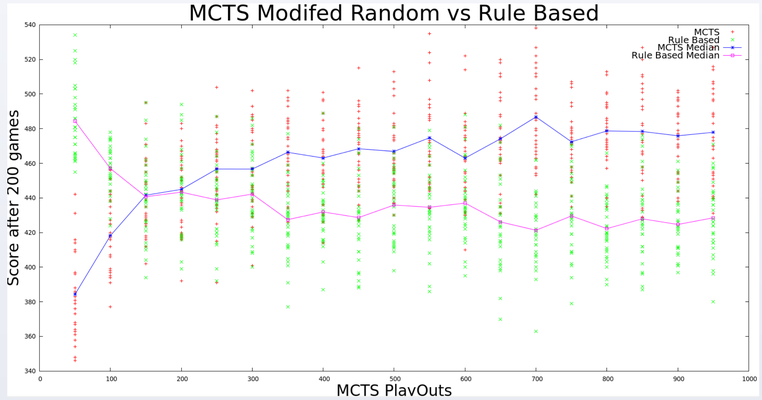
Right are the results from running the first version of randomMCTS against RuleBased. There is a point around 450 playouts where there are diminishing returns for performing more playouts. T-tests for the data sets above 400 playouts yield values ranging from .06 up to .39. Although the median is higher there is little difference between the performance of these two algorithms.





Left are the results from running the Rule Based algorithm against RuleBasedMCTS opposite from expected Rule Based greatly outperformed the MCTS even though they use the same set of heuristics to determine moves. This result is likely happening because the rules cause the algorithm to always explore the same branches which causes it to miss some branches that initially ‘look’ bad but in reality would be a preferable choice.

Results(contd.)



Above are the results from running the Rule Based algorithm against the second version of RandomMCTS. As previously stated the modification was to value the playout as the difference in the scores instead of the number of points scored. This proved to be a significant improvement it converged at 150 playouts instead of 450 and outperformed the rule based by significant margin. Above 300 playouts all of the P values from T-tests were below 4x10^-5, Showing that this is indeed an improvement over the rulebased.

Conclusions

The best algorithm for playing scopa was the Modified Random Monte Carlo Tree Search. While the heuristics work well they are based on my experience with the game of scopa and there is the possibility of improvement there.

Scopa is usually played to a set number of points typically 11 or 16. These results do not take that into consideration. They simply sum all of the points that a player recieves for 200 games. With that in mind there could be vast differences in performance of the algorithms if that were considered especially since the difference in points scored depending on the number of playouts was between 30 and 50 points. When averaged out over 200 games this only yields a slight advantage.

Refferences

- www.mcts.ai
- http://www.doc.ic.ac.uk/~sgc/papers/browne_ieee12.pdf
- http://www.informatik.uni-freiburg.de/~ki/teaching/ws0910/gamesem/bjarnason-et-al-2009.pdf

Contact

Questions or Comments? Contact me at gdaniels13@gmail.com

All of the code and complete results are available on github at https://github.com/gdaniels13/AI/tree/master/Scopa2

Or

RESEARCH POSTER PRESENTATION DESIGN © 2012
www.PosterPresentations.com