Adversarial Search in Gomoku

Chris Betsill
Westmont College
cbetsill@westmont.edu

1. INTRODUCTION

In this challenge, we were asked to create a Gomoku player to implement a Search method with Alpha-Beta Pruning.

Alpha–beta pruning is a search algorithm that seeks to decrease the number of nodes that are evaluated by the minimax algorithm in its search tree. It is an adversarial search algorithm used commonly for machine playing of two-player games. In addition my player also implements a machine learning algorithm to influence gameplay.

2. Procedure

Not wanting to reinvent the wheel,I began using the same basic framework of a Gomoku player which implemented the Least Mean Square learning algorithim. This gave me the benefit of already having methods in place for communicating with the game server, reading game states, evaluating possible moves, and reading and writing to files.

Since the search algorithm cannot exhaustively search the game tree, it searches to a maximum depth which is determined by the size of the board. This means that the smaller the board, the deeper the search can go. It also ensures that the 2 second time limit for a move is not reached.

The player then applies a heuristic to determine the strength of a move. The heuristic looks at what the effect of a potential move could be. It does this by building an attribute list for the potential move, which includes features like the number of consecutive pieces in a row in all adjacent directions for both the player and the opponent.

The weighted values of each feature are determined by the learning algorithm. This allows the player to over time determine what types of features are most significant. For training and testing, the player mainly played games against a version of itself which does not implement the learning algorithm.

It then aggregates these features down to more important combinations, such as how many 3 or 4 in a row in any direction (for both the player and opponent), and if the potential move will either result in victory, or prevent a loss at the next turn.

3. Results & Conclusion

The combination of using Alpha-Beta Search and the LMS learning algorithm creates an agent that, with sufficient training, is far superior to a player only implementing one or the other. The search helps to provide the player strategy, while the changing weight of heuristics (due to the LMS learning algorithm) provide the player with adaptability, and the ability to improve performance