David Leblanc

Assignment #0

CS773C Machine Intelligence
09/05/2012

Goal:

Design a Connect4 game program which can play against a human. The "AI" should use a minimax tree with alpha-beta pruning, and must have an adjustable ply depth. This program should ultimately be challenging to beat.

Design:

The program has been designed to be played just like the actual Connect4 game, using a simple command like interface. The "black" player starts first, followed by the "red" player. Each player can either be an AI player or Human. Both players can be set to "AI" so they can play each other. Once an "AI" is selected, the ply depth can be chosen. For the purpose of this assignment, the ply depth is limited to the range of 1-8, since beyond that depth, the computational cost becomes too high.

Strategy:

The goal of the game is to connect four tokens in a line. Lining up tokens in groups of two, or three is a good strategy and should be rewarded (or punished for the opponent). Another good strategy is to take control of the center of the board, so the static evaluator should take into account position of tokens relative to the center. Finally, it should be noted that it is better to tie, then risk opponent winning, therefore the focus of the AI is to play slightly more defensive, trying to deny opponent from winning.

Static Evaluator:

The main features used for the static evaluator to describe the game state are as follows:

- Count of groups of two (for both red and black)
- Count of groups of three (red and black)
- A successful group of four (red and black)
- Average distance from the center column of the board (red and black)

The groups of two and three are only counted if it is possible to make a straight line of four tokens using that group. In other words, for example, if black tokens are found between, or at the end of a line of red tokens, or if the line is on the edge of the board, this group is not accounted for. The average distance from center essentially means the closer a token is from the center, the higher it's score is. Tokens further from the center column (on the edge of the board) yields a lower score.

The state evaluation is done as follows:

$$f(GameState) = w_1 \cdot c_{2R} + w_2 \cdot c_{2B} + w_3 \cdot c_{3R} + w_4 \cdot c_{3B} + w_5 \cdot c_{4R} + w_6 \cdot c_{4B} + w_7 \cdot d_{c,R} + w_8 \cdot d_{c,B}$$

Where w are the weights, c the counts for groups of two, three, and four, d the distance from center, and R/B stands for red/black. For the purpose of this assignment, the weights w are constant, fixed, and manually chosen. The values for the weight have been chosen somewhat arbitrarily, based on

experience playing the game. The weights used are as follows (assuming the “AI” player is red):

$$w_1 = 0.4, w_2 = -0.6, w_3 = 0.5, w_4 = -1.5, w_5 = 0.9, w_6 = -5.5, w_7 = 0.5, w_8 = -0.5,$$

As shown above, the fourth and sixth weights are heavily “punished” which makes sense considering that is the count of triplets and win condition for the opponent, black. The fifth weight (the winning case for red) is not as favored, since the goal of these weights is to deny the opponent a victory. As mentioned previously, these weights are not the most exact and unique weights, they are simply designed for this type of strategy.

Results:

A few results can be shown in the sample run cases of playing “AI vs AI”. The main strength of this AI is that it attempts to deny opponent from winning, which is a decent goal in connect4. Since connect4 is a somewhat simple game (much simpler than checkers or chess, but slightly more complex than tic-tac-toe), the expected result should be more often than not a tie, or slightly favoring the first player.

From the videos, we can see that when the ply depth is set to 2, the game results in a tie. But for any ply depth greater than two, we see that the “AI” that plays first is favored to win. Although, if the second player's ply depth is set slightly higher, it can overcome the odds. Refer to videos and code for further detail regarding this.