Game Playing for Dots and Boxes

by Evan Liu & Debnil Sur

Motivation

The advancement of artificial intelligence has resulted in fundamental questions about the differences between the human mind and Turing computation in elementary cognitive tasks. Simple games taught to young children serve as testbeds for the difference of human analysis—chess problems rapidly solved by novices are incorrectly handled by Deep Thought, a chess computer of grandmaster rank [1]. We will study the application of artificial intelligence techniques to the game Dots and Boxes. This game starts with an $m \times n$ grid of dots; players connect adjacent dots via horizontal or vertical lines; the player who draws the final connecting line to create a box is rewarded with that box and gets to repeat her turn; and the game ends when all possible boxes have been created. Though simple, the game is analytically unsolved for any dimension greater than a 4×5 grid. Comparing human and computer play can thus provide greater insight into game playing strategy at large.

Project Scope

Thus, we aim to build a general game playing agent for a solved dimension of dots and boxes with performance superior to existing game-playing agents. Though extremely simple, this game has sparse literature. The most successful results in this field have utilized Markov decision processes, with alpha-beta pruning, to characterize the massive state space of the game [1]. We will begin with a similar approach and hope to improve upon the performance of previous game playing agents. "Solving" this game can be seen in two senses: (1) whether a player has a winning strategy or (2) by what margin a player would win in an optimal strategy. We will focus the development of our AI on the first question but, if time permits, extend our efforts to quantifying the consistent margin of victory on various sized boards.

Challenges

This game's rules are quite simple; after all, it is designed for children. The primary challenge thus arises in its massive state space. For instance, one of the largest solved problems, the 4×4 game, has 40 edges, a state space of 2^{40}, and a naive search space of 40! [1]. Because edges can be filled in any order, the set of available moves depends only on the board configuration. The large state and search spaces thus open themselves well to the application of the computational search techniques applied to games and discussed in class, such as minimax with alpha-beta pruning. Thus, we will test variations upon these search-based strategy to find a computationally tractable approach.

Metric

To measure the effectiveness of our approach, we will compare its playing strategy with existing AI, namely Dabble. Dabble is the most well-known publicly accessible AI for this game at the moment. While a game playing agent, its approach is not based upon the same search and state space algorithms we have discussed in class [1]. Consequently, it serves as an effective comparison of mass look-ahead gameplaying techniques with previously utilized approaches. We will play a set of 20 games of our AI against Dabble and record the win percentage. Then, we will adjust our approach, test performance, and continue until we have beaten Dabble a majority of times played.

Baseline

The baseline for performance will be playing our AI against a human player. Even a rudimentary AI, with its ability to look ahead in the huge amount of potential states, should be able to handily defeat a human player. Therefore, defeating a human in the vast majority of plays out of 20 games should be a trivial task for an artificial intelligence. Moreover, an inability to do so signifies major issues in our approach. Thus, it serves as a suitable benchmark.

Oracle

For any game, the oracle is naturally perfect play. In this case, this entails effectively traversing the search space to find optimal play at any given position. At lower dimensions (4×5 and below), the game is solved, demonstrating the feasibility of perfect play. Thus, although perfect play is the logical oracle, we are also aware that it is fully attainable at the dimensions we are studying. For this reason, the goal of our project will be to find a method that is consistently more successful than those of current game-playing agents. This will indicate that even though the game may be solved, our method is preferable to those currently used.

Input & Output

We model the game in accordance with the general game-playing approach discussed in class. The input to this general model is the grid size, represented by how many dots are on each side. many dots are on each axis). As we traverse the game space, or simulate moves that play the game, we update the state representing the current state. Here, we will emphasize again that the only move that matters is the one that completes a box; i.e. Player A could draw the first three edges but Player B draws the final edge and therefore receives the point for that box. Moreover, it does not matter which box has been scored by which player, just each player's absolute score. Therefore, we simply need to store the edges drawn thus far and the score of each player for the game-playing agent and the opponent to decide the next move. Thus, at each step of the game, the input will be the prior state and the output will be the optimal next state. This optimal state consists of the prior state, plus one optimally placed edge, and the score, potentially updated based on where the edge was played.

Conclusion

Simple for toddlers and complex for machines, games serve as a fascinating case study of potential difference in human analysis. Dots and Boxes has been analytically solved at lower dimensions by alternative game-playing techniques. As such, deeper study of it using principles from class serve as an interesting endeavour in game playing strategy at large.

References

[1] Barker, J. K., & Korf, R. E. (2012, July). Solving Dots-And-Boxes. In *AAAI*.