Research Paper for Use of Machine Learning to Develop a Better Artificial Intelligence for Playing Tic-Tac-Toe

Rachel Miller

December 2, 2004

Abstract

Machine learning is a tool that allows an Artifical Intelligence to learn from past sucesses and failures, increasing its ability over time. Though there are many different types of machine learning AIs, there is no concurence on the 'best' type of machine learning algorithm; different algorithms have different strengths, such as speed of learning, adaptability, or eventual skill. My project hopes to create several algorithms for a relatively simple game, Tic-Tac-Toe, and then have them compete to discover their relative advantages. Ideally, these algorithms would be modified according to these results to create better algorithms.

Contents

Acknowledgements

I would like to acknowledge the Thomas Jefferson Computer Systems Lab, for providing the hardware, software, and support to make this project possible. I would particularly like to thank Randy Latimer for his on going mentorship.

1 Introduction

Computers have enormous amounts of memory and processing available, but are limited by the logic they follow. Programs are only as useful as the logic they follow. This logic is given to the computer in the form of code. The use of Artifical Intellegence hopes to make computers better at difficult tasks that typically require a human. Machine learning, however, allows the computer to create its own logical rules, and learn from its past experiences. Machine Learning allows an AI to get smarter, even without additional dirrect programmer input. My project hopes to be able to play effectively for several two player games, primarily Tic-Tac-Toe.

1.1 Machine Learning

This project will collect data over time, as the program creates its own library of board position values. It will run a game to its finish between two computers, two humans, or a computer and a human. Each position in the board will be considered for its proximity to the final outcome. For example, a position right before a win would be considered valuable, while a position right before a loss would be considered undesirable. Multiple algoritms will have to be developed for judging the relative merits of a position and assigning a value. These must be tested against eachother, or against a third party algorithm, to judge their relative effectivnesses. If the position allready exists in the library, the value of the position in this game will be added to

the total value for the position. If the positioin does not allready exist, the algorithm will add it to the library. Board positions are considered as a one dimensional array of values of 0, 1, and 2. Board positions in the library will be sorted, with primary decision given to the first array value, and secondary to the second, etc. This will make the library easy to search through a binary algorithm, looking first in the middle of the library, and then going higher or lower, accordingly. As the computer moves, it considers each possible move. It looks for each move in the library, and chooses the one it thinks has the greatest value.

2 Background

There are multiple standard AI learning algorithm's. These algorithms include decission trees, neural networks, or genetic learning, amoung others. Many AIs are based upon a one-ply search of a library of board positions. These algorithms are most traditionally applied to two player games, such as checkers or chess. I therefore found this library and search technique most applicaple to my program, which would also be designed to play a two player game. Its relative simplicity was also desired, so potentially multiple algorithms could easily be employed, creating new algorithms and test them against eachother.

3 Theory

An algorithm should be able to increase its ability at a given task by learning through experience. Even without initial code related to strategy, over time, it should be able to predict which strategies lead to the greatest win rate. It should even be able to adapt to changing situations. My game playing algorithm hopes to fufill all of these components by gathering information about past games, and organizing it to find the most winning board positions.

4 Design Criteria

My project hopes to develop a new algorithm for machine learning. Machine learning comprises some of the most recent AI developments. Concurrently with my coding, I hope to research machine learning, and part of the more general field of AI as well. Hopefully, my new algorithm will lead to competitive AIs, that show distinct playing progress over time.

Though these algorithms will be developed through common games, the results areapplicable to computer science, particularly current AI researchers.

5 Materials used

My program is coded in C++, due to the languages ease of use for loading and storing libraries, and using large matrices.

My program does not require any extra external hardware, as its inputs

and outputs are all digital information.

6 Procedure and Workplan

I plan to finish the tic-tac-toe programing in one week. This will include programming the board, visuals, and the basic rules. I plan to create the library and sorting infastructure in six weeks. This will create a value for each board position, and add it to the library. The library moust be sorted for easy searching, as the number of possible board combinations is great. Sorting should minimize run time for the algorithm, making sure it can run in real time. I plan to create several algorithms in six weeks. This may include modifying a single algorithm with various weights, or even whole new methods of calculation. I will test the algorithms for two weeks. This will include more research, and developing values for board positions. I will adapt the algorithms for additonal uses for the next eight weeks. This might include adaptation to chess, checkers, a sliding puzzle, or other games or applications.

7 Results

Though my code and program are not complete, and thus no results have yet been created, I hope that my program will show increased win rates (or at least decread loss rates) against humans or other algorithms over time.

8 Discussion

9 Conclusion

10 Further Recommendations

Machine Learning algorithms are very powerful, and are applicaple to much more than games. As well as employing my algorithm for similar game situations, it could be used for other situations where strategies would need to be numerically ranked. These could deal with orginization tasks or modeling tasks.

http://www-2.cs.cmu.edu/satirist.org/learn-game http://theory.lcs.mit.edu/ mona/lectures.htm