Senior Project – Computer Science – 2015

Modelling Opponents in Board Games

Julian Jocque
Advisor – Prof. Rieffel

Abstract

Modelling opponents in a game has many potential applications. The goal of opponent modelling is to build a model of an opponent to try to predict future moves the opponent will make. A possible application would be building a computer Chess player that plays similarly to famous chess players, even in game situations which that player had never seen before.
To model opponents in board games I chose to use an algorithm called the Estimation Exploration Algorithm, hereafter referred to as the EEA. This algorithm has been shown to solve similar problems with a high degree of success. The algorithm works by creating and evolving a set of models and a set of tests then using the tests it created to iteratively increase the accuracy of the models. I created a system which uses the board game Konane and the EEA to model opponents.

Approach

Example Konane Board in starting position
from http://www.konanebrothers.com/march_2010_043_op_690x517.jpg

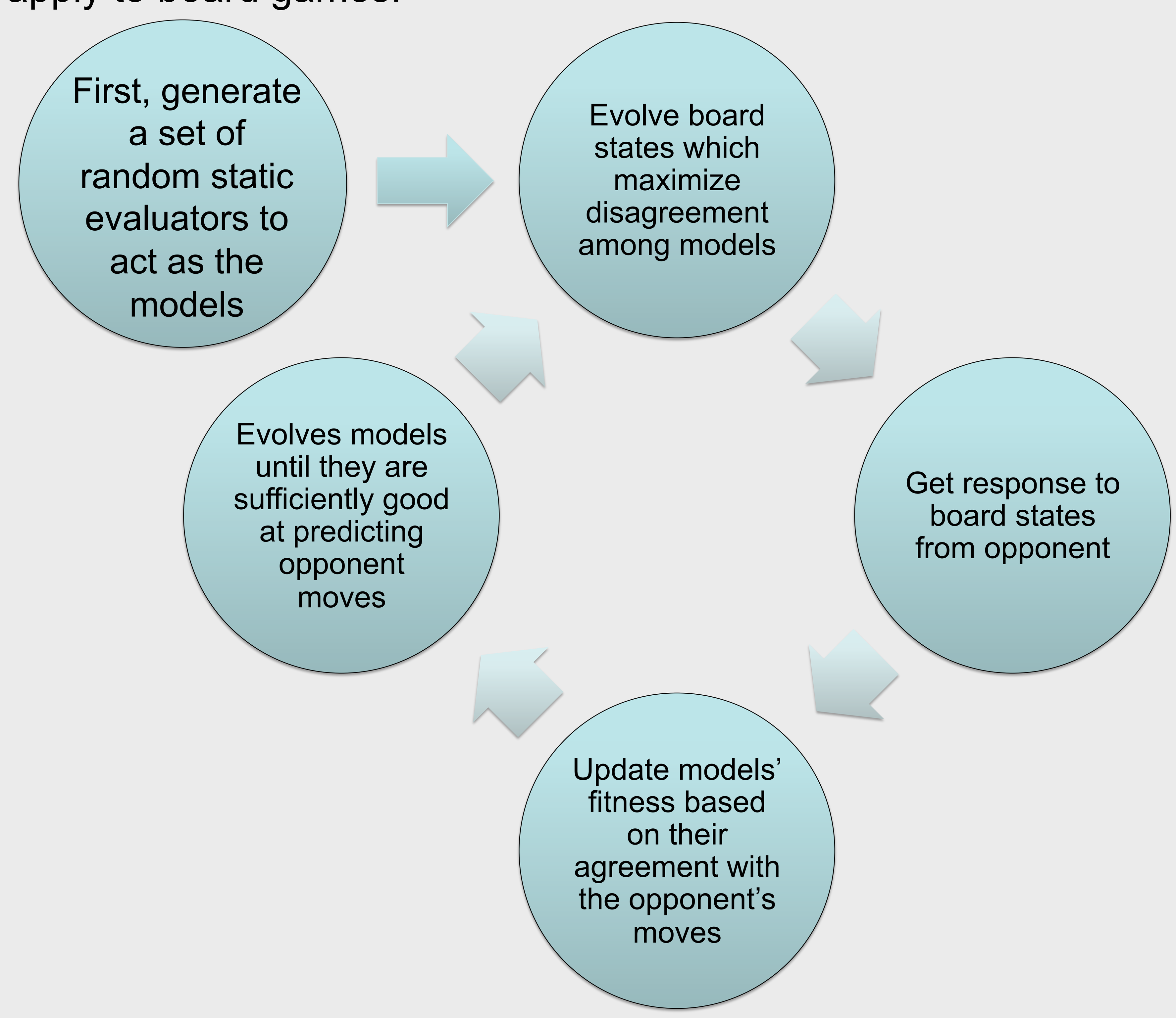
I chose to approach this problem using the simple game of Konane. Konane is a game similar to checkers and as such is both easy for a computer to model and for a computer to play.

The EEA is an evolutionary algorithm which continually evolves models and tests. The models for my approach are static evaluators for a player using the minimax strategy. The tests are board states for the opponent to respond to. The models are evolved to correctly predict the responses the opponent gives to the tests and the tests are evolved to maximize disagreement among the models. This disagreement is key as it minimizes the number of responses we need to get from the opponent because if a test causes great disagreement among a set of models, only the very best models will survive evolution. The fitness of the models is then judged by their ability to agree upon the moves made by the opponent for the evolved tests.

This approach allows for very rapidly finding models which accurately predict moves an opponent makes.

System

The system was written entirely in Python. All code is original with the exception of the Konane board game and rules which came from Prof. Rieffel.
The system works by following a modified EEA which is made to apply to board games:



Results and Evaluation

I was able to complete all objectives and get the system fully built. I was also able to get a large amount of data out of this system. One graph of the minimum, maximum and median fitness for a run of the system is below. Each black line indicates that the number of tests the models must accurately predict has increased. Although model fitness seems to stay the same it is actually rising to solve the harder and harder problem it is faced with.

The actual accuracy of models was found to vary wildly across different runs of the system. When the settings of the models and the opponents were similar, I found upwards of 90% agreement among them. However, when the settings differed I didn't find accuracy higher than 65%, which indicates there may be limits to how well this system can model an arbitrary opponent.