Training a Linear Heuristic on Chess Positions for the Minimax Algorithm

Tae Hyung Kim

August 5, 2018

1 Introduction

First, we make clear the notation that we shall use. We shall represent the type of chess pieces by the following abbreviations:

$$\{\mathbf{K} \leftrightarrow \text{King}, \mathbf{Q} \leftrightarrow \text{Queen}, \mathbf{R} \leftrightarrow \text{Rook}, \mathbf{B} \leftrightarrow \text{Bishop}, \mathbf{N} \leftrightarrow \text{Knight}, \mathbf{P} \leftrightarrow \text{Pawn}\}.$$

In addition, we shall use the standard method of identifying location of a piece. That is, each column shall be lettered a, ..., f while each row will be numbered $1, \ldots, 8$. Each position will be written with the column letter and the row number consecutively; for example, we would say that the white king initially occupies the position a4. Finally, we represent a piece, which includes its position, by the piece type followed by its position. For example, the king is denoted by **K**a4.

Recall the algorithm of minimax with alpha-beta pruning. Suppose that we represent a `Board` as an object (assume Python for simplicity) with the method `get_moves`. In addition, suppose that we have a heuristic function $h : \texttt{Board} \to \mathbb{R}$ which evaluates how good a board is.

Algorithm M. Suppose we have a board `board`.

M1. If

Define variables θ_{P_1,P_2} where P_1, P_2 are any two pieces. Now, the heuristic of interest in this paper is $h_\theta : \texttt{Board} \to \mathbb{R}$ defined by

$$h_\theta(\texttt{board}) = \sum_{P_1,P_2 \in \texttt{board}} \theta_{P_1,P_2} \cdot \chi_{\texttt{board}}(P_1)\chi_{\texttt{board}}(P_2)$$

where $\chi_{\texttt{board}} : \texttt{Piece} \to \{0,1\}$ is the indicator of whether a certain piece is in the board or not. Note that the product $\chi_{\texttt{board}}(P_1)\chi_{\texttt{board}}(P_2)$ simply returns 1 if the pair of pieces P_1, P_2 is in the board, and 0 otherwise.

The ultimate goal is to find the θ which will provide the best heuristic for the minimax algorithm. Given that the data we will be given is whether a certain board is "good" or "bad", the problem of computing θ is a logistic regression problem, as we wish to minimize the cost $\log(g(h_\theta(\texttt{board}))$ where g is the sigmoid function.