Considering Uncertainty in Chess Engine Evaluation

Sebastiano Rebonato-Scott

1 Abstract

We should prefer a position which we are certain is +2 to a position which we think is +3 but we are not sure. What we should aim to maximise in a chess game is not the evaluation with respect to our side, but our expected score from the game. I define the expected score as the number of points (0 for loss, 0.5 for draw, 1 for win) which we predict to obtain from this game.

Considering our uncertainty in the evaluation is important because the expected score from a game does not increase linearly with evaluation. For instance, the difference between the expected score from a position which is +0.5 and a position which is -0.5 is very significant, while the difference in expected score from a position which is +9 and one which is +10 is tiny. Therefore, the expected score from a position which is +2 ($\pm$ 0.2) to a position which is +3 ($\pm$2.5).

2 Implementation

The aim is to create a function W(p) which is inputted certain features of a position p, and outputs a weight $w \subset (0, \infty)$.

2.1 Obtaining Data

Suggested in this paper are two methods of obtaining data as to the uncertainty of various positions.

2.1.1 Method 1: WDL prediction

Positions in which the static evaluation is a poor predictor of the result of the game are high in uncertainty.

2.1.2 Method 2: Low / High Search Depth

Positions in which the evaluation returned from a low depth search is very different to the evaluation returned from a high depth search are high in uncertainty.

2.2 Loss Function

Let WDL(x) be a function which maps the evaluation of a position to its expected score. This function must:

- be strictly increasing
- be S-shaped
- have range [0, 1]

Let S(p) be the static evaluation of a position p.
Let H(p) be the evaluation returned by a high depth search of p.

To compute the loss of a function W(p) with method 2:

$T \leftarrow 0$
for all positions in dataset **do**
  $a \leftarrow H(p)$
  $w \leftarrow W(p)$
  $b \leftarrow S(p) \cdot w$
  $\Delta \leftarrow WDL(b) - WDL(a)$
  $T = T + \Delta$
end for
  return T

If method 1 is used instead then H(p) should be replaced with the eventual result of the game which the position came from.

2.3 Creating W(p)

I will first experiment with a handcrafted approach to implementing W(p), tuned using the loss function above. Some features which should be inputted are:

- king safety of both sides
- total material of both sides
- mobility of pieces (trapped pieces can lead to big misevaluations)
- whether or not the position is check