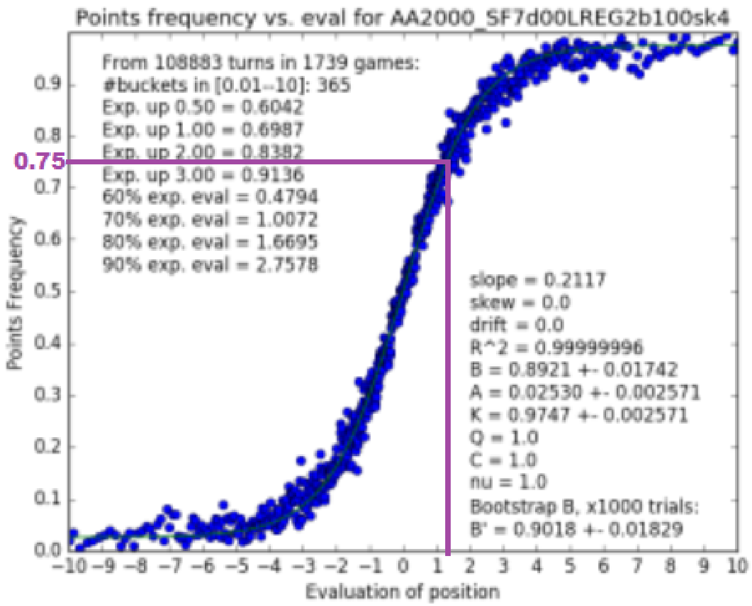
CSE702 Fall 2025 Week 3 Tuesday: Value and Points Expectation

Suppose you are $v = +1.00$ ahead as judged by a chess engine. What are your chances p_W of winning the game? And taking draws D into account, what is your expectation $e = p_W + 0.5p_D$?

This appears to have a natural answer that comes out of charts like the following, with value v on the horizontal axis and e vertical:



Here over 100,000 positions played by players rated 2000±10 against players rated 2000±10 were ordered by value for the player to move and divided into blocks of 100 each. The score % achieved on those positions is plotted as a blue dot (using Python tools this time). This graph fits a generalized logistic curve

$$A + \frac{K - A}{1 + e^{-Bv}}$$

with $R^2 > 0.9999999$. If you were impressed by R^2 values above 0.99 before, this takes the cake. The only difference from a simple logistic curve $\frac{1}{1+e^{-Bv}}$ is that the asymptotes are "nudged" away from $e = 0$ and $e = 1$ by the amount $A \sim 0.025$. (Note that A is the curve value as $v \to -\infty$, K as $v \to +\infty$, and $K = 1 - A$ here.) This means that from about 1-in-40 cases a player who was down hugely nevertheless won---or maybe 1-in-20 cases the player escaped with a draw (something between those extremes). There is however a statistical "swiz" here: positions from the same game are counted separately and treated as independent, even though their game outcomes are autocorrelated.

Segue to "Sliding Scale Problems" article.

Then the "Turkey Part 2" article.

Summary:

- Average Centipawn Loss/Average Scaled Difference are in units of **centipawns**.
- But those are particular to a given chess program. Stockfish notoriously used to give weirdly high evaluations to positions with moderate advantage.
- The "Turkey Part 2" issue extends to say that *chess programs do not need to respect the logistic relationship at all*---they can postprocess evaluations in any way that preserves the ordering of moves. The flaw in the arguments of Amir Ban.
- AlphaZero used units of **expectation**: $e = Pr[win] + 0.5Pr[draw]$.
- How does e correspond to the centipawn value v?
- Answer: As a Logistic Curve.
- Same kind of curve as for expectation given difference in ratings.
- Which justifies the idea of "giving odds" of material to equalize chances between players.

How to Handle?

The nasty problem---for me---is that the *slope* of the v-to-e curve depends on the absolute rating level R, in a way that the *diff*-to-e curve does not.

AlphaZero does not have this problem because it works toward its own single value of R.

It would be nice to dispense with the v-to-e issue by using expectation units directly. For a long time, I tried to use a direct conversion of every engine's v scale to the scale of the Rybka 3 chess program (which was the undisputed best program from 2008 until it was convicted of plagiarism in 2011). But the sliding-scale issue bit here too. In brief: *whose* win expectation will you use?

Other technical issues:

- Single-PV concordance in the T1 and EV metrics (and T3 and its variants) is about 2-3% higher than Multi-PV concordance.
- Equal-optimal moves do not have equal probabilities: https://rjlipton.wpcomstaging.com/2012/03/30/when-is-a-law-natural/

The explanations for these are related. The third parameter in my model was tuned to handle them.

On to Predictive Analytics