Skill and Prediction and Cheating at Chess

Université de Montréal

Kenneth W. Regan[1]
University at Buffalo (SUNY)

January 29, 2009

[1]Various projects in progress, co-workers named orally in-context. Sites: http://www.cse.buffalo.edu/∼regan/chess/fidelity/ (my homepage links), http://www.cse.buffalo.edu/∼regan/chess/ratings/ (not yet linked).

Skill Via Results

Skill in chess and several other games is standardly measured by the Elo Rating System, named for Arpad Elo. The old Elo formula for the change in a player's rating R_P after games g with opponents rated R_O was

$$\Delta R_P = K(W - L) + \sum_g c(R_O - R_P)$$

except that $c|R_O - R_P|$ was capped at $K - 1$ or $K - 2$ so that a win always netted 1 or 2 points. The "coefficients" K, c were generally chosen so that a player rated 200 points higher than opponents expected [needed] to score 75% [to preserve R_P]. (The new system approximates this using a logistic formula that also avoids the cap.)

Salient Features of Elo Ratings

- Points are zero-sum: what P gains O loses.
- Only rating differences matter—absolute numbers have no intrinsic meaning.
- Hence, ratings are subject to time drift—most believe they have *inflated*, but population models predict *deflation* since players gain points then leave (even Kasparov!).
- The coefficients—only K in the new system—control the speed and magnitude of updates.
- What rating to give a beginning player? A "rusty" player?
- Skill assessment is based ultimately only on results of games.
- Games are few, and subject to "luck."

Desired Constant Skill Spectrum

2800	World champ: Kasparov max 2851; Anand, Fischer $\sim$ 2800
2700	"Super-GM"
2600	"Strong GM"
2500	Grandmaster (GM)
2400	International Master (IM) (KWR, D. Levy, H. Berliner)
2300	FIDE Master (FM)
2200	National Master (USCF, CCF, et al.)
2000	Expert
1800	Class A (Richard Karp)
⋮	⋮
1000	Class E, "bright beginner"
⋮	⋮
600?	True beginner with "sight of the board"?
⋮	Kindergarten surge pressures USCF floor of 100!

Class Units and Game Depth

- With Elo 600 as floor, Chess has 11 class units from beginner to champion. (Others use 0 or 100 and say 14.)
- Laszlo Mero, *Ways of Thinking...* adapted this as a general measure of the human depth of a game—even estimating Soccer = 11 versus Basketball = 20.
- Criticism: $G' = G +$ coinflip has depth ≤ 1, but is no less complex than G.
- Chance enters more subtly in Backgammon—while for Poker, what is the length of a "game"?
- However, IMHO the concept and this chart I recall from 1996 (cannot find reference) are justified, and lead to a general yardstick for the intellectual capacity of computers' processing power.

Game Depth and Computers

10	Checkers (solved!)
11	Chess (8x8, "Western")
14	Shogi (9x9 board, captured pieces are returned to play...)
$\gg$ 20	Go

With 32-processor Hydra slaughtering super-GM Michael Adams 5.5–0.5 in 2005, and Deep Fritz 10 ($100) beating then-World Champion Vladimir Kramnik 4–2 on a standard-issue 4-core PC in Dec. 2006, computers are 12–13 on this scale. Hiroyuki Iida cites a panel predicting computer Shogi supremacy in 2012.

Cheating Alert

A $50 program on a sub-$1,000 2-core laptop can effectively collude even at the highest levels.

Rating System Controversies

- Are (chess) Elo ratings inflating?
  - Jan. 2009 FIDE list: 31 players ≥ 2700.
  - Jan. 1979: 1. Karpov 2705, 2. Korchnoi 2695...
  - Average of top 100 players: 1/09 2684, 7/06 2665, 7/01 2645.
- How to rate historical players before the Elo system was adopted in the 1960s?
- How much do faster time controls decrease the quality of play?
- Can we rate skill at various facets of the game? attacking? defense? endgame?
- Should we have "rating floors" to prevent sandbagging? Can we detect players throwing games or getting illicit help?

Proposed Answer

- Measure skill intrinsically based on quality of moves chosen rather than results of games.
- Use computer chess programs to judge quality of moves, then adjust for human factors.
- Model players as fallible decision-making agents (with Guy Mc.C. Haworth, Giuseppe DiFatta), with skill and tendency "profiles".
- Requires separating skill assessment from behaviour prediction.
- Confidence ranges on profiles enable statistical detection of fraud. . . hard!. . .

Virtues: a "standard candle"; more robust statistically because *moves are 30–40x more plentiful than games*, per player!

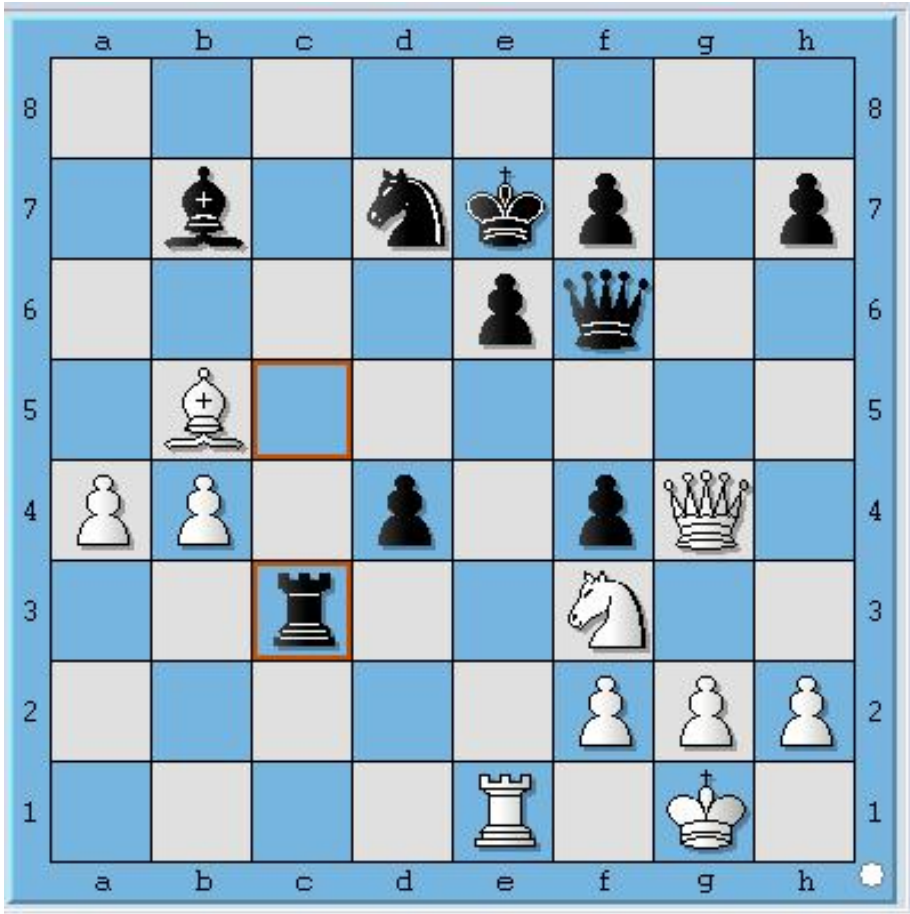
Chess Engines

- - essentially all work by iterative deepening: having completed the ply-$(d-1)$ round, begin the ply-d round. (One ply = one move by one player, also called one "half-move" in chess.)
- - all use some kind of search pruning to avoid full search of (ostensibly!) inferior moves. *Alpha-Beta* pruning reduces the *branching factor* ($\simeq$ time for ply-d vs. ply-$(d-1)$) from the chess average of 35 to a best-case factor of 2; actual programs achieve about 2.5.
- - apply their evaluation function to leaves of the search tree, then minimax evaluations back up to the root.
- - use a hash table to store evaluations of positions already seen.
- The ply-d round does a basic search of one additional ply at each leaf, then extensions are applied for lines of play flagged as promising or critical.

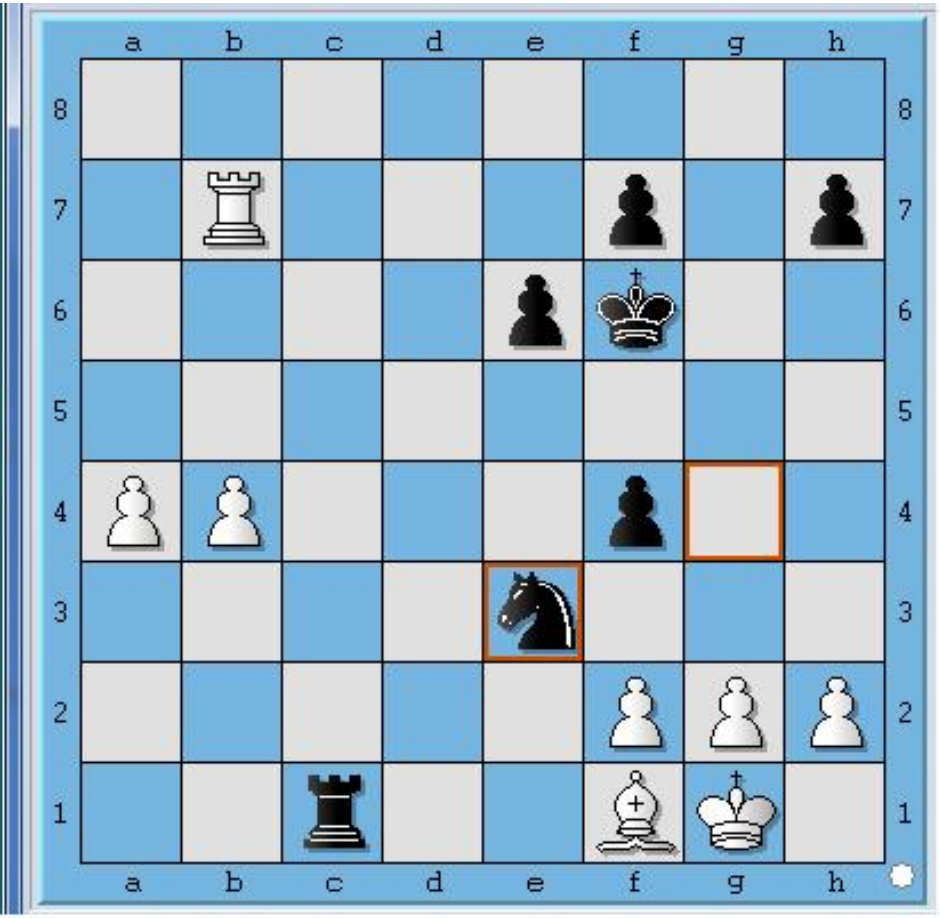
Sample Engine Output (Toga II 1.4b7), with "Swing"

```
 d/ext.  Time      Total nodes   Eval.    PV move
 ...
 12/37   00:02     1.089.963     +1.07    29.Nxd4
 13/37   00:03     2.196.479     +1.05    29.Nxd4
 14/38   00:05     4.323.312     +1.06    29.Nxd4
 15/47   00:10     8.786.279     +0.95    29.Nxd4 Qxd4 30.Rd1
Nf6 31.Rxd4 Nxg4 32.Rd7+ Kf6 33.Rxb7 Rc1+ 34.Bf1 Nxh2 (?)
 16/49   00:46    45.329.575     -1.27    29.Nxd4 Qxd4 30.Rd1
Nf6 31.Rxd4 Nxg4 32.Rd7+ Kf6 33.h3 avoiding 34...Ne3!!
 16/54   01:39    93.245.470     -0.94    29.Nh4
 16/57   04:23   258.742.003     -0.14    29.Bxd7
 17/57   06:14   363.133.546     -0.09    29.Bxd7
 18/57   06:54   403.700.038     -0.27    29.Bxd7
```

2008 WC Decisive Example



Kramnik-Anand, 2008 WC match game 8
Position before Kramnik's 29.Nxd4



Position after Anand's 34...Ne3!! Ouch!

Champion program Rybka sees it sooner

```
Rybka 3 1-cpu w32:  ...
   6     00:00          5.139     +0.65    29.Nxd4 Qxd4 30.Rd1
   7+    00:00          7.196     +0.85    29.Nxd4
   7+    00:01          8.825     +1.05    29.Nxd4
   7     00:01         10.452     +1.07    29.Nxd4 Qxd4 30.Rd1
Nf6 31.Rxd4 Nxg4 32.Rd7+ Kf6 33.Rxb7 Rc1+ 34.Bf1 Nxh2 (?)
   8     00:01         16.085     +1.07    29.Nxd4 Qxd4 30.Rd1
Nf6 31.Rxd4 Nxg4 32.Rd7+ Kf6 33.Rxb7 Rc1+ 34.Bf1 Nxh2
35.Kxh2 Rxf1 36.f3 Rf2 37.b5
   9     00:02         88.567     -0.19    29.Ng5 Ne5 30.Qh5
   9     00:07        272.886     +0.04    29.Nd2 d3 30.Bxd7
Kxd7 31.a5 Qg6 32.Qh3 Bd5 33.Rd1 Ke8
  ...
  16     04:50     13.680.534     -0.10    29.Nd2 d3 30.a5 Rc2
31.Bxd7 Kxd7 32.Rd1 Ke7 33.h3 Rb2 34.Qh5 Bd5 35.Nf3 Rxb4
36.Rxd3 Ra4 37.Ne5 Qg7 38.Qh4+
```

Skill Versus Prediction

- Give "extra" skill points for avoiding moves with *downswing*—and for playing moves with *upswing*! (Arbitrary?)
- Nevertheless, predict that even players of WC calibre will fall into traps nonnegligibly often, or miss the game's highest-complexity secrets.
- Both goals need context information:
  - what other options are available in the position?
  - how attractive/tempting are they?
  - how much time does the player have to decide?
  - how critical is the situation? the game?
  - what are the player's *forte* and *limitations*?

Context from Multi-PV output

```
...
 15/50   01:21      39.522.643    −0.01    29.Nd2   ...
 15/50   01:21      24.791.506    +0.40    29.Nxd4  ...
-------------------------------------------------------
 16/73   02:24     117.929.404    −1.07    29.Nxd4  ...
 16/73   02:24     113.331.703    −0.50    29.Qg8   ...
 16/73   02:24     117.926.589    −0.41    29.Qh5   ...
 16/73   02:24     102.819.455    −0.17    29.Bxd7  ...
 16/73   02:24      95.230.189    −0.13    29.Nd2   ...
-------------------------------------------------------
```

Delta $= (0.00, 0.04, 0.28, 0.37, 0.94, \ldots)$

Prediction Goal

Estimate $D_P(\tau)$ = the distribution of move choices made by player(s) of skill profile P at game turns τ, with confidence intervals.

Obstacles:

- Heterogeneous positions/choices/players...
- "The experiment is not repeatable..."
- Human brains are not aleatory (?)
- Choices between turns τ, τ' are not independent.

Main Modeling Assumptions

Main Modeling Assumptions

1. For players P at all skill levels, $D_P(\tau)$ depends predominantly on the intrinsic values of the moves available at turn τ.

Main Modeling Assumptions

1. For players P at all skill levels, $D_P(\tau)$ depends predominantly on the intrinsic values of the moves available at turn τ.
2. Chess programs strong enough to be *sensei* for P will estimate the values closely enough to yield robust statistics.

Main Modeling Assumptions

1. For players P at all skill levels, $D_P(\tau)$ depends predominantly on the intrinsic values of the moves available at turn τ.
2. Chess programs strong enough to be *sensei* for P will estimate the values closely enough to yield robust statistics. *Sensei* $= 2$ class units.

Main Modeling Assumptions

1. For players P at all skill levels, $D_P(\tau)$ depends predominantly on the intrinsic values of the moves available at turn τ.
2. Chess programs strong enough to be *sensei* for P will estimate the values closely enough to yield robust statistics. *Sensei* $= 2$ class units.
3. Choices are predominantly independent.

Main Modeling Assumptions

1. For players P at all skill levels, $D_P(\tau)$ depends predominantly on the intrinsic values of the moves available at turn τ.
2. Chess programs strong enough to be *sensei* for P will estimate the values closely enough to yield robust statistics. *Sensei* $= 2$ class units.
3. Choices are predominantly independent.
4. The higher a particular move is valued, the more likely it is to be played, regardless of skill level.

No. 4. is NOT the same as saying that for players at all skill levels, the best moves are the ones more likely to be played, which is ridiculous.

Basic Model's Main Assumption

5. For players of all skill levels, the best moves are the ones most likely to be played.

Indeed, upon ordering legal moves $m_1, m_2, m_3, \ldots$ in nondecreasing order of deltas $(\delta_0 = 0.00, \delta_1, \delta_2, \ldots)$, we posit that $p_i = \Pr_P[m_i]$ is inversely monotone in δ_i.

Simplicity Assumptions

6. For $i > 0$, p_i depends only on δ_i and p_0 by a simple mathematical relation R.

Just having δ_i would lose all context, but this says the only needed context goes into the probability of the best move. Note also that $\sum_i p_i = 1$ removes the apparent degree of freedom on p_0. So this is as strong a simplicity assumption as possible.

Simplicity Assumptions

6. For $i > 0$, p_i depends only on δ_i and p_0 by a simple mathematical relation R.

Just having δ_i would lose all context, but this says the only needed context goes into the probability of the best move. Note also that $\sum_i p_i = 1$ removes the apparent degree of freedom on p_0. So this is as strong a simplicity assumption as possible.

> *Entia non sunt multiplicanda praeter necessitatem*
> (William of Occam)

Simplicity Assumptions

6. For $i > 0$, p_i depends only on δ_i and p_0 by a simple mathematical relation R.

Just having δ_i would lose all context, but this says the only needed context goes into the probability of the best move. Note also that $\sum_i p_i = 1$ removes the apparent degree of freedom on p_0. So this is as strong a simplicity assumption as possible.

> *Entia non sunt multiplicanda praeter necessitatem* (William of Occam), *sed lineares combinationes ipsi ad-libidum valitur* (Schrödinger).

Simplicity Assumptions

6. For $i > 0$, p_i depends only on δ_i and p_0 by a simple mathematical relation R.

Just having δ_i would lose all context, but this says the only needed context goes into the probability of the best move. Note also that $\sum_i p_i = 1$ removes the apparent degree of freedom on p_0. So this is as strong a simplicity assumption as possible.

> *Entia non sunt multiplicanda praeter necessitatem* (William of Occam), *sed lineares combinationes ipsi ad-libidum valitur* (Schrödinger).

The full model intends to be a weighted linear combination of basic-model $R_d(- - \delta_{i,d} - -)$ over different ply-depths d, with "Swing" providing non-monotonicity by linear means.

Skill Parameters

Two parameters seem natural and unavoidable:

- A conversion σ from the (1/100)-pawn units of δ_i to dimensionless magnitude: $x = \delta_i/\sigma$.
- An exponent c that magnifies high and low values of x.

Smaller $\sigma \implies$ higher $x \implies$ smaller differences in δ_i are more sharply felt $\implies$ higher skill. Hence called σ for *sensitivity*, also because it is linear in the standard deviation of various curves.

Higher $c \implies$ higher $(--x--)^c$ when $x > 1 \implies$ denominators with high δ_i vanish rapidly $\implies$ blunders are less likely $\implies$ higher skill. Called c for *competence*.

Make a 1-dimensional model?

In tests, σ and c trade off in an annoying "long high ridge." Why not declare just one parameter?

Make a 1-dimensional model?

In tests, σ and c trade off in an annoying "long high ridge." Why not declare just one parameter?

> *Dim-entia non sunt dividenda praeter necessitatem*
> (Mailliu of Macco).

Make a 1-dimensional model?

In tests, σ and c trade off in an annoying "long high ridge." Why not declare just one parameter?

> *Dim-entia non sunt dividenda praeter necessitatem* (Mailliu of Macco).

- The full model also has weight parameters w_d, with weights toward higher d meaning higher skill.

Make a 1-dimensional model?

In tests, σ and c trade off in an annoying "long high ridge." Why not declare just one parameter?

> *Dim-entia non sunt dividenda praeter necessitatem* (Mailliu of Macco).

- The full model also has weight parameters w_d, with weights toward higher d meaning higher skill.

Given player data Δ_P, the object is to find a robust statistical best-fit of some family of curves $g_{c,\sigma}(\delta_i)$ (or $g_c(x)$) over σ and c [and the w_d]. Use the fit to compute the probabilities p_i and confidence intervals for test quantities derived from them.

Some Modeling Relationships R

All cases involve normalizing $g_c(0) = 1$.

- "Shares" (recall $x_i = \delta_i/\sigma$):

$$\frac{p_i}{p_0} = g_c(x_i), \qquad \text{so} \qquad p_i = \frac{g_c(x_i)}{\Sigma_i g_c(x_i)}.$$

Too Cold.

- "Power Shares"

$$\frac{\log(1/p_0)}{\log(1/p_i)} = g_c(x_i), \qquad \text{so} \qquad p_i = p_0^{1/g_c(x_i)}.$$

Too Hot? Or Just Right?

- "Entropy Shares"?

$$\frac{p_i \log(1/p_i)}{p_0 \log(1/p_0)} = g_c(x_i).$$

or as cross-terms

$$\frac{p_i}{p_0} \cdot \frac{\log(1/p_0)}{\log(1/p_i)} = g_c(x_i)?$$

Inverse Exponential

- $$g_c(x) = e^{-x^c}, \qquad \text{i.e.,} \qquad g_{\sigma,c}(\delta_i) = \frac{1}{e^{(\delta_i/\sigma)^c}}.$$
- For $c = 2$, half of a bell curve.
- For $c = 1$, p_j/p_i depends only on $\delta_j - \delta_i$, i.e. the relative probabilities of two moves depends only on their difference in value, irrespective of other moves (except that their absolute probabilities depend on them). Like saying sales of a Stephen King (horror) and Kathy Reich (mystère) stay in ratio even when a Sue Grafton (ABC mystery) is added to the best-seller list. **False in chess.**
- For $c \leq 1$, no point of inflection—discernment of advantage is sharpest between the best move and all others. **True, at least for grandmasters.**

Inverse Polynomial

A.

$$g_c(x) = \frac{1}{1 + x^c}.$$

B.

$$g_c(x) = \frac{1}{(1 + x)^c}.$$

Both approximate inverse-exponential for the same c, for small x, but have fatter tails.

Logistic/SecantH-Related Curves

A.

$$g_c(x) = \frac{2}{e^{x^c} + e^{-x^c}}.$$

B.

$$g_c(x) = \frac{2}{(e^x + e^{-x})^c}.$$

C.

$$g_c(x) = \frac{2^c e^{-x}}{(1 + e^{-x})^c}$$

This is a *Hubbert curve* for $c = 2$.

D.

$$g_c(x) = \frac{4}{e^x + 2 + e^{-x}}.$$

More-Extreme Curves

- Inverse Logarithmic:

$$\begin{aligned} g_c &= 1/(1+\ln(1+x^c)) \\ g_c &= 1/(1+(\ln(1+x))^c) \\ g_c &= 1/(1+\ln(1+x))^c \end{aligned}$$

- Inverse Double-Exponential:

$$\begin{aligned} g_c &= 1/e^{e^{x^c}-1} \\ g_c &= 1/e^{(e^x-1)^c} \\ g_c &= (1/e^{e^x-1})^c. \end{aligned}$$

(The last one causes a "race" between σ and c.)

Are there other natural and simple curves to consider?

Nasty Scaling Issues

The "metric" $\mathbf{x}$ of chess-engine evaluations is scaled only linearly by σ. However, the one-pawn difference between $\delta_i = 5.00$ and $\delta_i = 6.00$ is relatively insignificant, because a blunder of either magnitude will usually have an equal effect on (instantly losing) the game.

Handled by defining various line-elements $\mathbf{dx}$ that drop off sharply with x, or rather with unscaled δ, then integrating (by Simpson's Rule) to map the engines' recorded deltas δ_i to "effective deltas" δ'_i.

Obviously sensible, but so far I get better results with unit scale. Worse, although this mapping is monotone decreasing, it often yields a higher player-error expectation $E = \sum_i p_i \delta'_i$ ("Failure of Convexity").

Simpler is to cut off all cases of $\delta_i > C$ to C. My data-preprocessing script (in Perl) forces $C \leq 10$. Rybka 3 provides C as an automatic feature, but I reported it buggy for $C \geq 5$, and $C = 4$ used in my data runs may be too low, especially when a player is "only" 2 pawns ahead.

Program

- 8000+ lines of C++ code, from seminar student's 500-line original.
- That's about 150 dense single-spaced A4 pages.
- Object-orientation used to filter subsets of moves by player, date, etc., etc.; to allow modular substitution of different curves, relationships, fitting methods, scale metrics...; to build a makeshift but reusable menu system; to log user selections as well as results to files for automatic replay. Hence not written in Matlab or Maple...
- Allows standard max-likelihood fitting, my homegrown "Percentiles" method, or least-squares of simpler derived quantities.
- Needs to combat numerical instability, especially for the "Power Shares" model—when it does millions of Newton iterations taking noticeable time even for data sets of size $< 10,000$.
- Primitive binary-search minimization—N.J.A. Sloane's `gosset` would be nice, but...

Data

- Data in 10-PV mode to depth 18 by Toga II 1.4 beta7, a version privately made for me by engine author Thomas Gaksch.
- 9,500 moves (skipping turns 1–8) from world-championship match games beginning with Fischer-Spassky, 1992.
- Toga II versions are free and open-source, and based on Fruit 2.x (ditto) by Fabien Letouzay.
- Nevertheless, this and the latest public Toga II (1.4 beta 5c) beat all commercial engines except Rybkas and HIARCS 12 in my tests.
- The private version cuts down extensions to promote basic search, and tests equivalent to depth 16–17 of earlier ones—and depth "12.5" of Rybka 3.
- Earlier data set ran Toga II 1.2.1a to depth 15 on top-level games from 2005–07. Re-running with Rybka 3 to depth 13...
- Estimate Toga II 1.2.1a depth 15 to be about Elo 2900–2950, Toga II 1.4b7 depth 18 to hit 3,000. Still short of 2 class units.

Fitting

- ML: find σ, c to maximize $\sum_\tau \log(1/p_\tau)$, where p_τ is the computed probability of the move actually played at turn τ. Seems to discard information about the moves not played—(?)
- Suppose all "spreads" $\Delta = (0.00, \delta_1, \delta_2, \dots)$ were equal. Then we could fit by building a histogram for each i of the % of time h_i that move m_i was played, and fit σ, c to make $g_{\sigma,c}(\delta_i)$ closest to $R(h_0, h_i)$.
- For example, in the "Shares" model, if we have N cases of $(0.00, 0.10, 0.30, 0.30)$ with the top move were played 50%, the next 25%, and the other two 12.5% each, then we fit $1/(1+\delta/\sigma)^c$ exactly with $c = 1$ and $\sigma = 0.1$, upon normalizing the "shares" $(1, 1/2, 1/4, 1/4)$.
- How to handle when spreads vary widely?

My "Percentile Fitting" method...

- For any particular (σ, c) and datum Δ_τ, the model computes probabilities $(p_0, \ldots, p_{\ell-1})$.
- For each percentile q, let j_q be the index of the probability interval p_j in which q falls.
- If the index i_τ of the move played in turn τ gives $i_\tau < j_q$, then τ is a "hit" for q.
- If $i_\tau > j_q$, then τ is a "miss."
- If $i_\tau = j_q$, then τ is a partial hit proportional to how far along p_i the value q actually falls.
- For each q let r_q be the proportion of hits. Minimize $\sum_q (q - r_q)^2$ over σ, c.

This reduces to the histogram method for the case of all spreads equal, when it also agrees with ML. Assuming independence, it yields simple mixed Bernoulli trial confidence intervals.

is vastly unsound!

- The previous reasoning applies for any ordering of the δ_i and corresponding p_i, for any turn τ.
- But if the orderings are uniformly and independently randomized, every curve $g_{c,\sigma}$ fits, nearly perfectly! (The percentiles are always assigned so that uniform permutations agree with p_i-weighted random choices of move.)
- One can contrive small examples where certain percentile sets give perfect fits for markedly different σ, c.
- Nevertheless, the method seems to work in practice...
- ...under moderate perturbations of data? Not sure...

An example of an application where one does not wish to randomize data, at least not too much.

Results...

- Demonstrate program, show results...

Reliability of Top Players

Whose view is closer to reality?

- Capablanca: went 10 years without losing a game; vs.
- Tartakower: *"Les fautes sont là, sur l'échiquier, attendant d'être commises."*

Mindful of Nassim Taleb's *The Black Swan* and my own 2400-ish play, I expected Tartakower to be reflected in the simple-ratio ("Shares") model fitting best with a fat-tailed curve.

However, Capa was the better player—and the powering model fits best, with an inverse-exp or logistic/sech curve!

(Still, a single blunder in 200+ moves can cost 1 game out of 5...)

Alleged Cheating With Computers

- Allegations typically enumerate that Player X matched the first choice of computer program Y on $r\%$ of the moves, with $r \gg$ the "40–45%" I was told humans achieve at most.
- Many problems—no such accusation in over-the-board play has been sustained without physical evidence.
  - "the first choice" is not well-defined: at what depth or time? Hash-dependent, not reproducible...
  - Logs of tests with Y not kept.
  - My runs show super-GMs regularly hit $r = 56\%$. "Anyone" can play like a super-GM for short periods.
  - Under forcing attacks, which limit options to stay in the game, players will match Y more often.
  - "Littlewood's Law"—in 1 out of every 20 games you will think your opponent cheated at the 95% confidence level.
- Nevertheless, given r and a sequence $[m_\tau]$ of Y-moves, my program will output an expectation $E[r' \mid X]$ for the number of matches to Y, along with confidence intervals.

Recent Cases and "Indications"

- The 10/2006 Topalov-Kramnik "WC Scandal" (Y = Fritz 9): show demo.
- Topalov at San Luis 9/2005: 7.5/8 start, but statistically indistinguishable from Anand's unimpeached Mexico 2007 victory.
- Eugen Varshavsky, World Open 7/06 (Y = Shredder 9): made to remove big floppy hat and trenchcoat, worked like Kryptonite.
- Umakanth Sharma, Asian Team Ch. 12/06 (Y = HIARCS 10): found with Bluetooth device in cap, banned for 10 years, no games released(?).
- D.P. Singh, 2006, roomed with Sharma (Y = Junior 9): show.
- Krzysztof Ejsmont, Poland 7/07 (Y = Rybka 2.3.2a): ejected from event after matching "98%" (I get at least 88%), later acquitted "for want of proof."
- M. Sadatnajafi, Turkey 4/08 (Y = ??): set up and seen consulting cellphone during game, Farsi text messages found, ejected and banned. too small data.

Accusations–unfounded?

- Topalov at Corus, 1/07 and 1/06: article in the *Sueddeutsche Zeitung* by IM Martin Breutigam, Y not specific, no hearing, only 2 games...
- Anna Rudolf, France 1/08: regarded as false accusation.
- More?

Human Nature? (show demo)

"Big" (+−) is >=2.00 and "Def" (+/−) is 1.00 to 2.00

Category	#Turns	Predic. / Actual	Falloff
Overall :	9850	976.2 / 1003.1	= 0.97
Ahead :	1006	120.4 / 205.4	= 0.59
Behind :	783	77.1 / 149.1	= 0.52
Up Big :	394	48.3 / 117.7	= 0.41
Up Def :	612	72.1 / 87.7	= 0.82
Evenish :	8061	778.8 / 648.6	= 1.20
Down Def:	508	47.8 / 70.7	= 0.68
Down Big:	275	29.3 / 78.4	= 0.37

Inflation? Deflation? Historical Ratings?

- Show demo.
- (So far): This decade's champions look better when regressed against 1970s–1980s standards; past champions look worse when judged by today's standards.
- Hence, deflation? Real human skill improvement?
- An effect of extensive training with computers?

Fitting Humans Below WC Level...

...will require hundreds of thousands of processor-hours of high-depth data!

With DiFatta and Haworth: can low-depth data be reliable?

Note: All our Toga versions, run only to depth $d = 10$, soundly beat the Crafty program run to depth 12, which was touted as a "world championship gold standard" in the only refereed published predecessor study, by Matej Guid and Ivan Bratko.

Rating Strong Programs on Human Scale

- Too strong for direct intrinsic measurement?
- Disturbed by computer-similarity effect?
- Idea: Measure programs at low fixed depths d intrinsically, thus fitting *them* to human scale.
- Then play matches between the same engine at depths d' versus d, to fit a curve of the marginal strength gained by an extra ply of search. (Published analyses hint this decreases with d, but by how much?)
- Extrapolate ratings to high d as/if needed.
- Finally observe which depths the engine playing under standard human time controls typically reaches. (Problem: endgames)

Not Just Chess?

- Search-engine rankings: "top 10 choices" evaluated by mix of automated and human juries, at great cost...
- Is an executive favoring one advisor more than others?...on the least-obvious decisions?
- Can we elucidate general patterns of human fallibility?

A numerical math/complexity problem

Given $a_1, a_2, \ldots, a_n \geq 1$, how quickly can we approximate the number p, $0 < p \leq 1$, such that

$$p^{a_1} + p^{a_2} + \cdots + p^{a_n} = 1?$$

Challenging even for $n = 2$, $a_1 = 1$, i.e. given a, find p such that $p + p^a = 1$. Can one improve on my program's use of Newton's method, and avoid considerable numerical-decay problems?

Other opportunities to contribute:

- Better minimization method, such as `gosset`.
- Explore σ, c tradeoff—maybe one parameter can be eliminated after all.
- Help improve data-gathering. Do quicker shortcuts work?
- Extend program to automate reading of single-PV mode data used in cheating tests.
- Other fitting methods and statistical tests. . . [FIN]