Intrinsic Chess Ratings

AAAI 2011

Kenneth W. Regan\(^1\)    Guy McC. Haworth\(^2\)

University at Buffalo (SUNY)    University of Reading (UK)

August 8, 2011

\(^1\)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/ (all data, not yet linked).

\(^2\)Various other projects—Google Haworth Reading CENTAUR to find http://centaur.reading.ac.uk/view/creators/90000763.html
Fallible Agent Model – Inputs and Outputs

- Player skill parameters $s, c, \ldots$
- At each decision point (game turn), ‘objective’/‘hindsight’ utility values $v_0 \geq v_1 \geq v_2 \ldots$ for the available options (moves) $m_0, m_1, m_2, \ldots$
- Probabilities $p_0, p_1, p_2, \ldots$ of each option.
- Skill assessment of the option $m_j$ that was actually chosen.
- Also generates projected confidence intervals for various statistics.
Intrinsic Chess Ratings

Why? What to do with it?

Intrinsic skill based on decisions made rather than outcomes.

- In chess, outcomes of games subject to opponent’s play, ‘luck’.
- 50-odd games per year is a small sample—
- —but 1,500 moves from those games is a healthy sample.

1. Have outcome-based ratings been consistent over time? (In chess, has there been ‘Elo Inflation’?)

2. Compare players of different historical eras.

3. Measure effect of thinking time on skill.

4. Evaluate statistical claims of cheating with computers, or alternatively, ‘sandbagging’.

5. Human player training.
Main Principle

The probability $p_i$ of an option $m_i$ depends primarily on its value $v_i$ in relation to the values of other options.

Other principles/working assumptions:

1. Decisions at different turns $t$ are independent.
2. For agents of all skill levels, the higher $v_i$, the higher $p_i$.
3. Weaker agents prefer weaker moves.
4. Values $v_i$ need not be omniscient, only high enough above the agents being modeled to represent an authority figure (in chess: 200–400 Elo higher).

Well, 1. is contradicted by human players having multi-turn plans, while 2. and 3. contradict each other. But we argue 1. gives “local-limited dependence” while our full model harmonizes 2. and 3. via linear combinations—though it violates 4.
Skill Parameters for Agents $P$

- **Sensitivity $s$**: How well can $P$ perceive small differences in value?
  \[ x_i = \frac{v_0 - v_i}{s} \]
  Lower $s$ is better. (Necessary to convert from utility units to dimensionless?)

- **Consistency $c$**: How (in)frequent are mistakes?
  \[ p_i \sim \frac{1}{1 - (1 - x_i)^c} \]
  Higher $c$ is better.

*Full model—not this paper—not as general?*

- **Depth of calculation**: Applies to alternating-move games, transactional decision making, ‘looking ahead’.
Basic Model

General form: for certain relation function $R$ and curve family $g = g_{s,c}$,

$$R(p_i, p_0) = g_{s,c}(v_i, v_0).$$

Simplest in the sense that the dependence on $v_j$ for $j \neq i$ is only thru $p_0$ and the constraint $\sum_j p_j = 1$.

This paper: First scale down differences in value according to the overall imbalance in the position, defining

$$x_i = \delta(v_i, v_0)/s.$$

Then take $R$ to be a ratio of logs and $g$ an inverse-exp curve, namely:

$$\frac{\log(1/p_0)}{\log(1/p_i)} = e^{-x_i^c}. $$
Full Model—better? needs attention?

- Values $v_i^d$ from chess engines at different search depths $d$.
- Apply basic model at each $d$ to get $p_i^d$.
- Additional skill parameters $w_d$ called weights, with $\sum_d w_d = 1$. (Hopefully specifiable by one or two scalars.)
- Overall probability then given by the linear combination

$$p_i = \sum_d w_d p_i^d.$$  

Idea: A move whose goodness appears only at higher depths ("swing-up" move) should be less probable than a move whose good value is evident at all depths. Likewise a "swing-down" move (a good ‘trap’ in chess) should be more probable than a less-deep error, even though both have the same value at the reference depth.

**Working assumption of using basic model:** Swing-up and swing-down cases offset over large-enough sets of moves.
Implementation Details

1. Values $v_i$ by champion chess program РYВКА 3 run in 50-PV mode to reported depth $d = 13$.
   - Estimated 2650–2700 strength.
   - Usually 6–8 hours per x64 CPU thread per game.
   - РYВКА 3 led field by 200 points in 2008; only program with “Multi-PV cap” feature saving much time; now under cloud for GPL violations.

2. Scaling derived from huge amounts of Single-PV mode data also serving as scientific control:
   $$\delta(v_0, v) = \int_{v_i}^{v_0} d\mu(x) \quad \text{with} \quad \mu(x) = \frac{1}{1 + x}.$$ 

3. Eliminate turns 1–8 of any game, turns where РYВКА 3 judges more than a 3-pawn advantage, and turns involved in repetitions.
A Weird Fact, and its Correction

• In upwards of 10% of moves, Rybka 3 gives equal-top values to two or more moves, i.e. \( v_0 = v_1 = \ldots \)
• All of our models would give equal probabilities to such moves.
• However, in cases of exactly two equal-top moves, the move first-listed by Rybka is preferred almost 60% of the time!
• Likewise the second move of an equal-top triad is preferred to the third, although triads etc. are rare enough to ignore.
• Believed cause: The first move gains a higher value at some low depth and stays, since engines keep order of tied moves stable.
• Alas, Rybka 3 hides the lowest plies, so it’s hard to tell.
• Hope is that the “full model” will reflect this naturally by \( w_d \) for low \( d \) picking up the higher value.
• Basic model adjusted by deducting 0.03 from \( v_1, v_2, \ldots \) in tied cases, which nearly equalizes the probabilities across training sets.
Training the Model

Objective: Find a robust relation between Elo ratings and values $s, c, \ldots$

2800 World’s best players
2700 World-class players
2600 Strong grandmaster (GM)
2500 Typical GM
2400 Typical International Master (IM)
2300 Typical FIDE Master (FM)
2200 National Master
2000 Expert (U.S.)
1800 Class A (U.S.)
1600 Class B, etc...
Intrinsic Chess Ratings

Training Test Sets

- All available games under *standard round-robin tournament conditions* where both players were rated within 10 points of an Elo milestone $R = 2700, 2600, \ldots, 2200$, played in three different time periods.
- Use statistical fitting to derive $s_R, c_R$ for each $R$ in each period.
- Main finding: Little or no difference across time periods for each $R$, hence no “rating inflation.”
- Since the FIDE rating system was extended below 2200 in the past decade, test sets were compiled for 2100, 2000, \ldots, 1600, widening the interval from ±10 to ±15 or ±20 to get enough games.
- Gamescore errors in these test sets needed manual correction. Frequency about 1% of games, but cause about 20% additional error in master-level sets.
Two-Parameter Fitting: $c$-fit Convention

- While $s$ varied widely from 0.08 to 0.16 and higher, $c$ was observed to stay in a narrower band, about 0.52 down to 0.43, fairly linearly as $R$ from 2700 to 1600.
- Moreover, the $s$, $c$ values trade off against each other in long ridges of near-optimality for various fitting methods.
- Hence decided to impose a linear fit to determine $c'_R$ first. (By happenstance almost exactly 0.07 per 100 Elo.)
- Given $c'_R$, can do one-parameter fit to get $s'_R$. Quality of fit is reasonably close to that of original $s_R$, $c_R$ fit.
- Future: use a tool like Neil Sloane’s gosset to investigate $s$, $c$ space further.
Fitting Methods—which to use?

1. **Maximum Likelihood**: Maximize the product of probabilities $p_j$ over the moves $m_j$ that were actually played. Gives markedly inferior results.

2. **Bayesian Update**: Used by Haworth in his models, not yet here—should it approach ML values?

3. Solve the two equations that set $\sum p_0$ equal to the actual number of times move $m_0$ was played, and set the projected error $\sum_i p_i \delta(v_0, v_i)$ summed over all game turns equal to the actual error. (Called ‘FF’ in tables.)

4. Fit expected vs. actual percentages of playing the second-best, third-best, fourth-best moves $m_1, m_2, m_3, \ldots$ of the analysis engine, as well as $m_0$. Problem: heteroskedasticity.

5. Fit *probability percentiles* instead, hopefully solving the problem of 4. (Called ‘PF’, or ‘CF’ in tables below when $c$ is fitted first.)

6. Some other fitting methods?
The Percentile-Fitting Method

- Pick a percentile grid, e.g. \( q = .02, .04, .06, \ldots, .96, .98 \) (used in paper).
- A game turn \( t \) is a “hit” (scoring 1) if the played move \( m_j \) satisfies
  \[
  q \geq c(j) = \sum_{i=0}^{j-1} p_i.
  \]
- Partial hit if \( c(j-1) \leq q \leq c(j) \), scoring \( \frac{q - c(j-1)}{c(j) - c(j-1)} \).
- Let \( r_q \) be the proportion of hit scores for \( q \) over all turns \( t \).
- Minimize \( \sum_q (r_q - q)^2 \).
- (Alternatives: multiply by \( H(q) \) and/or minimize some other \( l_p \) distance instead of least-squares.)
Can Basic Model Be Tuned Better?

- PF and CF are observed fairly regularly to:
  - over-predict $p_0$ by about 0.003 (i.e., 0.3%),
  - under-predict $p_1$ by about three times as much, and
  - predict probabilities of $m_2, m_3, \ldots$ fairly closely.
- Thus PF and CF turn out to be biased estimators of $p_0$...
  - ...but helpfully hedge against false positives in cheating testing.
- FF is an unbiased estimator of $p_0$, but so-far seems to give less control on other $p_j$.
- This holds for all reasonable curve families $g$ tried thus far, such as
  
  $g_c(x) = \frac{1}{1 + x^c}$ or $g_c(x) = \frac{1}{(1 + x)^c}$.

- Changing relation $R$ to simply $p_i/p_0$ rather than ratio of logs warps the model markedly the other way. Use some $R$ intermediate between them?
### Results For 2006–2009

<table>
<thead>
<tr>
<th>Elo R</th>
<th>$s$</th>
<th>$c$</th>
<th>$c'_R$</th>
<th>$s'_R$</th>
<th>$mm_p/mm_a$</th>
<th>$ad_p/ad_a$</th>
<th>$Q_{fit}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2700</td>
<td>.078</td>
<td>.503</td>
<td>.513</td>
<td>.080</td>
<td>56.2/56.3</td>
<td>.056/.056</td>
<td>.009</td>
</tr>
<tr>
<td>2600</td>
<td>.092</td>
<td>.523</td>
<td>.506</td>
<td>.089</td>
<td>55.0/54.2</td>
<td>.063/.064</td>
<td>.041</td>
</tr>
<tr>
<td>2500</td>
<td>.092</td>
<td>.491</td>
<td>.499</td>
<td>.093</td>
<td>53.7/53.1</td>
<td>.067/.071</td>
<td>.028</td>
</tr>
<tr>
<td>2400</td>
<td>.098</td>
<td>.483</td>
<td>.492</td>
<td>.100</td>
<td>52.3/51.8</td>
<td>.072/.074</td>
<td>.016</td>
</tr>
<tr>
<td>2300</td>
<td>.108</td>
<td>.475</td>
<td>.485</td>
<td>.111</td>
<td>51.1/50.3</td>
<td>.084/.088</td>
<td>.044</td>
</tr>
<tr>
<td>2200</td>
<td>.123</td>
<td>.490</td>
<td>.478</td>
<td>.120</td>
<td>49.4/48.3</td>
<td>.089/.092</td>
<td>.084</td>
</tr>
<tr>
<td>2100</td>
<td>.134</td>
<td>.486</td>
<td>.471</td>
<td>.130</td>
<td>48.2/47.7</td>
<td>.099/.102</td>
<td>.034</td>
</tr>
<tr>
<td>2000</td>
<td>.139</td>
<td>.454</td>
<td>.464</td>
<td>.143</td>
<td>46.9/46.1</td>
<td>.110/.115</td>
<td>.065</td>
</tr>
<tr>
<td>1900</td>
<td>.159</td>
<td>.474</td>
<td>.457</td>
<td>.153</td>
<td>46.5/45.0</td>
<td>.119/.125</td>
<td>.166</td>
</tr>
<tr>
<td>1800</td>
<td>.146</td>
<td>.442</td>
<td>.450</td>
<td>.149</td>
<td>46.4/45.4</td>
<td>.117/.122</td>
<td>.084</td>
</tr>
<tr>
<td>1700</td>
<td>.153</td>
<td>.439</td>
<td>.443</td>
<td>.155</td>
<td>45.5/44.5</td>
<td>.123/.131</td>
<td>.065</td>
</tr>
<tr>
<td>1600</td>
<td>.165</td>
<td>.431</td>
<td>.436</td>
<td>.168</td>
<td>44.0/42.9</td>
<td>.133/.137</td>
<td>.129</td>
</tr>
</tbody>
</table>

The $s'_R$ column shows a reasonable progression.
Results Over Time

With $c'_R$ fixed, corresponding $s'_R$ values:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2700</td>
<td>0.513</td>
<td>0.080</td>
<td>0.084</td>
<td>n.a.</td>
</tr>
<tr>
<td>2600</td>
<td>0.506</td>
<td>0.089</td>
<td>0.087</td>
<td>0.087</td>
</tr>
<tr>
<td>2500</td>
<td>0.499</td>
<td>0.093</td>
<td>0.092</td>
<td>0.091</td>
</tr>
<tr>
<td>2400</td>
<td>0.492</td>
<td>0.100</td>
<td>0.103</td>
<td>0.103</td>
</tr>
<tr>
<td>2300</td>
<td>0.485</td>
<td>0.111</td>
<td>0.117</td>
<td>0.116</td>
</tr>
<tr>
<td>2200</td>
<td>0.478</td>
<td>0.120</td>
<td>0.122</td>
<td>n.a.?</td>
</tr>
</tbody>
</table>

**Beyond Paper:** Supporting evidence (of “no inflation”) from the Single-PV control data that Average-Difference ($ad$) levels for round-robin tournaments of each given FIDE category are consistent over time.
Operating the Model: 1. Skill Assessment

To compute an Intrinsic Performance Rating (IPR) for a player $P$ over a set of games:

- Fit $s, c$ for those games.
- Iterate to find a stable point on the $s'_R, c'_R$ line.
- Output the corresponding $R$.

Beyond Paper—Some Old Masters:

<table>
<thead>
<tr>
<th>Player</th>
<th>Years</th>
<th>IPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Howard Staunton</td>
<td>1841–1858</td>
<td>1990</td>
</tr>
<tr>
<td>Adolf Anderssen</td>
<td>1851–1878</td>
<td>2060</td>
</tr>
<tr>
<td>Paul Morphy</td>
<td>1857–1859</td>
<td>2310</td>
</tr>
<tr>
<td>Wilhelm Steinitz</td>
<td>1860–1876</td>
<td>2220</td>
</tr>
</tbody>
</table>

Figures are somewhat subject to change.
**Beyond Paper:** Can do IPR’s for entire tournaments too.

<table>
<thead>
<tr>
<th>Event</th>
<th>Year</th>
<th>IPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>St. Petersburg quad</td>
<td>1896</td>
<td>2390</td>
</tr>
<tr>
<td>St. Petersburg prelims</td>
<td>1914</td>
<td>2370</td>
</tr>
<tr>
<td>St. Petersburg finals</td>
<td>1914</td>
<td>2570(!)</td>
</tr>
<tr>
<td>New York</td>
<td>1927</td>
<td>2620</td>
</tr>
<tr>
<td>AVRO</td>
<td>1938</td>
<td>2620</td>
</tr>
<tr>
<td>The Hague WC</td>
<td>1948</td>
<td>2640</td>
</tr>
<tr>
<td>Curacao CT</td>
<td>1962</td>
<td>2580(!)</td>
</tr>
</tbody>
</table>
Let \([m_{Et}]\) be a sequence of moves over game turns \(t\) that are liked by some engine \(E\).

Let \([m_{jt}]\) be the sequence of played moves. How close ‘should’ these sequences be? for a non-cheating player?

Plug in \(s_R, c_R\) for the player’s rating \(R\), or an upper bound on it.

Then \(M = \sum_t p(E_t)\) is the expected number of matches to \(E\).

And \(\sigma^2 = \sum_t p(E_t)(1 - p(E_t))\) is the projected variance.

Output the \(z\)-score \(z = \frac{\hat{M} - M}{\sigma}\), where \(\hat{M}\) is the player’s actual number of matches.

Is this a reliable way of testing null hypotheses of no cheating?
Beyond Paper: Projected vs. Actual $\sigma$

- How close are $z$-scores from player performances in events to a normal distribution?
- Are they robust enough for use in court cases?
- Is the model capturing a significant enough aspect of skill (at chess)?

**Distributional Testing Method:** For each Elo milepost $R$,

1. Generate 10,000 random subsets of 9 games each from the training set for $R$. [Why 9? Typical for major events.]
2. Also randomly choosing Black or White in each game. Pretend the resulting set $S$ of moves is a performance by one ‘player.’
3. Tabulate the $z$-scores $z_S$. 

Intrinsic Chess Ratings
## Distributional Results: FF Method

<table>
<thead>
<tr>
<th>$z$-rng</th>
<th>-4.x</th>
<th>-3.x</th>
<th>-2.x</th>
<th>-1.x</th>
<th>-0.x</th>
<th>+0.x</th>
<th>+1.x</th>
<th>+2.x</th>
<th>+3.x</th>
<th>+4.x</th>
</tr>
</thead>
<tbody>
<tr>
<td>Targ.</td>
<td>0.3</td>
<td>13</td>
<td>214</td>
<td>1359</td>
<td>3413</td>
<td>3413</td>
<td>1359</td>
<td>214</td>
<td>13</td>
<td>0.3</td>
</tr>
<tr>
<td>2700</td>
<td>0</td>
<td>23</td>
<td>248</td>
<td>1350</td>
<td>3223</td>
<td>3266</td>
<td>1508</td>
<td>353</td>
<td>28</td>
<td>1</td>
</tr>
<tr>
<td>2600</td>
<td>1</td>
<td>44</td>
<td>236</td>
<td>1313</td>
<td>3127</td>
<td>3220</td>
<td>1708</td>
<td>321</td>
<td>28</td>
<td>2</td>
</tr>
<tr>
<td>2500</td>
<td>3</td>
<td>55</td>
<td>430</td>
<td>1604</td>
<td>3059</td>
<td>2960</td>
<td>1415</td>
<td>345</td>
<td>46</td>
<td>3</td>
</tr>
<tr>
<td>2400</td>
<td>1</td>
<td>30</td>
<td>309</td>
<td>1484</td>
<td>3128</td>
<td>3121</td>
<td>1510</td>
<td>377</td>
<td>39</td>
<td>1</td>
</tr>
<tr>
<td>2300</td>
<td>0</td>
<td>36</td>
<td>322</td>
<td>1606</td>
<td>3133</td>
<td>2996</td>
<td>1465</td>
<td>389</td>
<td>52</td>
<td>1</td>
</tr>
<tr>
<td>2200</td>
<td>1</td>
<td>47</td>
<td>364</td>
<td>1451</td>
<td>3231</td>
<td>3136</td>
<td>1420</td>
<td>324</td>
<td>26</td>
<td>0</td>
</tr>
<tr>
<td>2100</td>
<td>1</td>
<td>24</td>
<td>299</td>
<td>1418</td>
<td>3131</td>
<td>3281</td>
<td>1477</td>
<td>338</td>
<td>29</td>
<td>2</td>
</tr>
<tr>
<td>2000</td>
<td>0</td>
<td>40</td>
<td>344</td>
<td>1410</td>
<td>3102</td>
<td>3212</td>
<td>1500</td>
<td>350</td>
<td>41</td>
<td>1</td>
</tr>
</tbody>
</table>

Somewhat flattened compared to normal distribution, indicating some loss of modeling fidelity. But not too bad.
### CF Method—negative skew is deliberate

<table>
<thead>
<tr>
<th>z-rng</th>
<th>-4.x</th>
<th>-3.x</th>
<th>-2.x</th>
<th>-1.x</th>
<th>-0.x</th>
<th>+0.x</th>
<th>+1.x</th>
<th>+2.x</th>
<th>+3.x</th>
<th>+4.x</th>
</tr>
</thead>
<tbody>
<tr>
<td>Targ.</td>
<td>0.3</td>
<td>13</td>
<td>214</td>
<td>1359</td>
<td>3413</td>
<td>3413</td>
<td>1359</td>
<td>214</td>
<td>13</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>2700</th>
<th>2600</th>
<th>2500</th>
<th>2400</th>
<th>2300</th>
<th>2200</th>
<th>2100</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>29</td>
<td>63</td>
<td>97</td>
<td>45</td>
<td>63</td>
<td>97</td>
<td>38</td>
<td>73</td>
</tr>
<tr>
<td>330</td>
<td>418</td>
<td>658</td>
<td>392</td>
<td>520</td>
<td>678</td>
<td>678</td>
<td>407</td>
<td>505</td>
</tr>
<tr>
<td>1585</td>
<td>1838</td>
<td>2069</td>
<td>1699</td>
<td>1991</td>
<td>2117</td>
<td>1734</td>
<td>1862</td>
<td>1962</td>
</tr>
<tr>
<td>3350</td>
<td>3373</td>
<td>3199</td>
<td>3269</td>
<td>3389</td>
<td>3467</td>
<td>3423</td>
<td>3304</td>
<td>3304</td>
</tr>
<tr>
<td>3105</td>
<td>2947</td>
<td>2609</td>
<td>2953</td>
<td>2612</td>
<td>2601</td>
<td>2958</td>
<td>2918</td>
<td>2918</td>
</tr>
<tr>
<td>1304</td>
<td>1171</td>
<td>1095</td>
<td>1318</td>
<td>1103</td>
<td>886</td>
<td>1186</td>
<td>1113</td>
<td>1113</td>
</tr>
<tr>
<td>278</td>
<td>174</td>
<td>243</td>
<td>294</td>
<td>247</td>
<td>138</td>
<td>233</td>
<td>205</td>
<td>18</td>
</tr>
<tr>
<td>18</td>
<td>13</td>
<td>24</td>
<td>29</td>
<td>32</td>
<td>9</td>
<td>17</td>
<td>18</td>
<td>0</td>
</tr>
</tbody>
</table>

Flattening still makes +2.x and +3.x columns higher than desired, but again not too bad.
Conclusions and Future Work

- Model captures enough chess skill to show steady progression of parameters with regard to Elo ratings.
- And to approximate a normal distribution of actual versus expected matching, reasonably closely.
- Scientifically effective on cheating-allegation cases. [Show Elista 2006 demo if time.]

To-do list:

1. **Analysis Interchange Format** (AIF) standard so others can compile analyzed games.
2. Implement full model, to see if it is really better.
3. Tune model(s) better to eliminate the $p_0, p_1, p_2$ slight warping.
4. Further application to chess and other games.
5. Extend to (non-game) decision-making settings.