Fitting Methods for Inferring Selection Probabilities from Hindsight Utility Values

Kenneth W. Regan
Department of CSE
University at Buffalo
Amherst, NY 14260 USA
regan@buffalo.edu

Tamal Biswas
Department of CSE
University at Buffalo
Amherst, NY 14260 USA
tamaltan@buffalo.edu

March 13, 2013

Abstract

We consider the problem, given a vector of possible actions a_i whose utilities $u_i = u(a_i)$ may not be fully apparent, and an agent Z, infer probabilities p_i for Z to choose a_i. For instance, can we gauge the probability of a trader making the best investment choice, or of a student giving the best answer on a multiple-choice exam, given extensive training data from subjects with comparable skill sets according to parameters that determine Z? We compare various applicable fitting methods, in the model of [Regan and Haworth, AAAI 2011] where the choices are possible chess moves, the utilities are values given by chess programs stronger than the human players, and the parameters for Z correspond to the Elo rating scale of skill at chess. In this application there are metrics for quality of fit that are orthogonal to the measures operated on by the various fitting methods, including the "percentile fitting" method introduced in the above reference. In this application, percentile fitting markedly out-performs maximum-likelihood estimation under these metrics, as does simple solving of equations involving the parameters.

Keywords. Computer games, chess, decision making, probabilistic inference, machine learning, statistics, fitting methods, maximum likelihood.

1 Introduction

Given ℓ-many possible actions with utilities $(u_1, \ldots, u_\ell)$, can we infer probabilities $(p_1, \ldots, p_\ell)$ for an agent Z to choose the respective actions? We assume the listed actions are mutually exclusive and exhaust all options, so $\sum_i p_i = 1$. In a simple rational-choice situation with perfect information about utility, and ignoring for now the possibility of two or more actions with equal best value, we would project $p_i = 1$ for the action a_i with highest u_i, and $p_j = 0$ for all other actions. In real-world situations of bounded rationality, however, we may expect to see substantial probability on a variety of reasonable options, and also on poor but deceptively attractive options. Can we model this so that over sufficiently many *turns* t at which an action must be chosen, and

given parameters z quantifying the ability of Z to perceive the stipulated values u_i, we can make accurate projections of aggregate statistics?

We interpret the utility values u_i as given by design, by hindsight, or by some recognized authority, but not directly furnished to Z. We call the general problem, "converting utilities into probabilities." Here are three instances:

(1) *Multiple-choice tests.* Suppose we wish to infer the probability not only of the best answer being selected, but also the probabilities of other answers being chosen, in terms of parameters z representing the education and preparedness level of test takers. It is interesting to ask how accurately one can make this projection given only single numerical values u_i representing the value or attractiveness of each answer. This may be especially helpful when the u_i themselves represent partial-credit values for the $\ell - 1$ inferior answers.

(2) *Portfolio management and trading.* Here the a_i are investment choices presented at a given time, and the u_i represent the returns that would have accrued at a specific later time when the actual transaction is evaluated, had a_i been chosen. The u_i may also include measures of the overall health of the portfolio at that point, and so depend on choices and outcomes in other transactions. For now we emphasize that these other measures too are computed with hindsight, even those that concern hedging against risk. The problem is to predict how well a trader Z will "sniff out" what prove to be the best investments.

(3) *Games of strategy.* In chess and other games, the actions a_i are legal moves at a given turn t, and the u_i are values given by some strong computer program for these moves. The parameters comprising z correspond to aspects of playing skill. In chess, skill is measured in terms of the Elo rating system [Elo78].

The experimental results reported in this paper are for application (3). We argue that they have sobering implications for procedures that may be employed in applications (1) and (2). The main issue is the poor performance of maximum-likelihood estimation (MLE), where the above setting already ensures that MLE is the large-data limit of Bayesian inference. This is exacerbated when there are differences in entropy between the inferred distributions $(p_i)_t$ at different turns t.

In (1) the utility values are given by design, in (2) by hindsight, and in (3) by authority—the chess programs have published ratings [B$^+$13] far in excess of the best human players. In all cases, the agent Z does not know the values u_i, and the problem is to determine how well-aligned the agent's fallible ranking of the options is with the authoritative one. Here are some aggregate statistics by which to measure such alignment:

(a) *Best-Choice frequency* (BC). On a multiple-choice test, this is the score without partial credit, expressed as a percentage. In chess it is the frequency of move agreements with the computer; it is called MM in [RH11].

(b) *Aggregate Error* (AE). On an exam with partial credit, this is the total number of points lost. In trading it is the total monetary shortfall from optimal investment choices. In chess it is the sum, over all moves (where the computer judges the player chose an inferior move), of the difference in value between the optimal move and the chosen one. Chess programs standardly give these values in units of *centipawns*—figuratively hundredths of a pawn, whereupon AE represents the total number of pawns "lost" by the players in the set of positions t. Where there is no confusion we also use AE for the per-move *average error*.

(c) *Ordinal Ranks* (OR). Besides the BC frequency f_1, these give the frequency f_2 of choosing the second-best option, f_3 for third-best, f_4 for fourth-best, and so on. Of course there may be no parity between second-best choices: some may be "close" while others are large errors. Projections of OR may take the difference in value into account, so that it is permissible to mix these kinds of data points. Likewise permissible is to assume all turns have the same number ℓ of options, padding those having fewer with "dummy options" having large negative values, which will translate to essentially-zero probability in projections.

Note that OR entails indexing choices by their values: $u_1 \geq u_2 \geq \cdots \geq u_\ell$. Also note that the projected probabilities $p_{t,i}$ over $t \in T$ alone suffice to generate projected values for these statistics, namely

$$\hat{f_k} = \frac{1}{T}\sum_t p_{t,k}; \qquad \hat{ae} = \frac{1}{T}\sum_t p_{t,k}(u_1 - u_k).$$

We consider models in which the projected probabilities $p_{t,i}$ are given by a function P of the utility values $u_{t,i}$ and the parameters comprising $\vec{z}$; we assume the latter are independent of t. Three properties, at least the first two desirable, are:

(i) *Independence.* For all t, the generated values $(p_{t,1}, \ldots, p_{t,\ell})$ depend only on $(u_{t,1}, \ldots, u_{t,\ell})$ and $\vec{z}$.

(ii) *Extensionality of utility.* For all t and Z, $u_{t,i} = u_{t,j} \implies p_{t,i} = p_{t,j}$.

(iii) *Monotonicity.* For all t and i, if $u_{t,i}$ is replaced by a greater value u', then for all Z, $p'_{t,i} \geq p_{t,i}$.

Note that (iii) is different from saying that always $p_{t,1} \geq p_{t,2} \geq \cdots \geq p_{t,\ell}$, though that follows from (ii) and (iii) as-stated. The reason for doubting (iii) is the application to Z of all skill levels—it says that improving the hindsight quality of an answer makes it no less attractive, which runs counter to the idea in chess that "weaker players prefer weaker moves." One reason independence is desirable is that it yields inherent standard deviations and hence confidence intervals for projections of these statistics:

$$\sigma_{BC} = \sqrt{\sum_{t=1}^{T} p_{t,1}(1 - p_{t,1})} \tag{1}$$

$$\sigma_{AE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\sum_{i\geq 2} p_{t,i}(1 - p_{t,i})(u_{t,1} - u_{t,i})}. \tag{2}$$

We regard the components of $\vec{z}$ as fittable parameters whose values are determined by regression over training sets giving actual choices made by agents. After $\vec{z}$ are fitted to values z, the fitted model computes

$$(p_{t,1}, \ldots, p_{t,\ell}) = P_z(u_{t,1}, \ldots, u_{t,\ell}).$$

Issues that arise in training, including the choice of fitting method, are the focus of this paper.

2 Training Data and Fitting Metrics

Every piece of training data is an *item*

$$I = (u_1, \ldots, u_\ell; i; e)$$

where i is the index of the option that was chosen, and e gives supplementary information about the person or agent who made the choice. In examinations, e may be prior information about past grades or test results, or alternatively may be posterior information such as the overall score on the exam itself. In chess, e can be the Elo rating of the player making the move, either before or after the games by that player included in the training set. We index an item and its components by the turn t, and sometimes write just t in place of I, calling the item a *tuple*.

In this paper we postulate a mapping $E(Z)$ from the agent space to the values e that come with the training data. This mapping need not be invertible—indeed when two or more scalar parameters comprise $\vec{z}$ this is not expected. Its main point is that when regressing on a subset of items whose values e are all equal (or close to equal) to obtain z, comparing $E(z)$ and e acts as a check on the results. They need not be equal—perhaps E itself has been fitted by a final linear regression against e and the particular e is an outlier in this fit—but lack of closeness is a reason to doubt the method used to fit z.

When the estimation method does not guarantee that the projected means agree with the sample means, for BC and AE in particular, then the difference from the sample becomes a measure of goodness (or rather badness) of fit. We express the deviations from the sample means as multiples of the inherently projected standard deviations, that is as multiples of σ_{BC} and σ_{AE}. Technically this assumes independence of turns, but this assumption is not a strong consideration because we are not using them to infer z-scores for hypothesis tests. They are mainly a convenient choice of units when comparing results between training sets of different sizes.

Our main independent metric for goodness of fit involves the projected ordinal frequencies $\hat{f}_k$ for $1 \leq k \leq \ell$ and the sample values f_k. The *ordinal rank fit* (ORF) is given by

$$\sum_{k=1}^{\ell} (\hat{f}_k - f_k)^2.$$

In tables, the frequencies are expressed as percentages—equivalently, the ORF score is multiplied by 10,000. We do not weight ORF by the number of items with $i = k$ (i.e., by f_k itself), but we consider a fitting method that tries to minimize this score.

The metrics and training and fitting methods all extend naturally when items I_t for different turns t are given different weights w_t. The weight w_t gives some measure of the value or difficulty of the decision.

In the multiple-choice test setting, when the u_i represent points values in scoring the exam as well as predictors for choosing answers, the value can also be represented as a multiplier on the u_i themselves. In the portfolio trading setting, the u_i already reflect the total dollar value being invested. In the chess case, however, the corresponding notion of "value" of a move depends on extraneous information about the state of the game. Hence we instead consider the idea of weighting decisions by difficulty. The most difficult case, intuitively, is when there are many options of near-optimal value. This applies equally to multiple-choice exam questions, investment choices, and chess moves. This is reflected by the *entropy* of the projected distribution $(p_1, \ldots, p_\ell)$.

The only demurral is that there might not be much at stake, if *all* options are close to equal, or if there is a sharp jump from near-optimal choices to ones that are clearly markedly inferior. In chess this can happen if a position is "dead drawn," except for recognizably silly moves that give away a piece for nothing. In the present paper we do not try to avoid this objection, as it arises rarely in the training data. Hence we stay with the *entropy weights*

$$ew_t = \sum_{k=1}^{\ell} p_{t,k} \log_2(1/p_{t,k})$$

as a measure of difficulty. Since the ew_t factor through as multipliers on the above scores, we do not need to change the notation for them.

3 Fitting Methods

In all cases we are given training data consisting of items. Since the data are fixed, we can regard the probabilities $p_{t,j}$ as functions of Z alone.

3.1 ML and Bayes

The *Maximum Likelihood* method is now to fit $\vec{z}$ to maximize the probability of the selected options i_t in the training data. By independence this means to maximize

$$\prod_t p_{t,i_t},$$

which is equivalent to minimizing the log-sum

$$\sum_t \ln(1/p_{t,i_t}).$$

We write z_{ML} for some value of Z that minimizes this logsum, and call $P(z_{ML})$ the *max-likelihood probabilities*.

For completeness, we derive the result that Bayesian iteration approaches the ML estimator in this setting. Let $A(z)$ denote the event that the agent in the training data with chosen options $\vec{i} = i_1, i_2, \ldots$ arises from $Z = z$. By Bayes' Theorem, assuming the space Z is finite,

$$\begin{aligned}
\Pr(A(z) \mid \vec{i}) &= \frac{\Pr(\vec{i} \mid A(z))\Pr(A(z))}{\Pr(\vec{i})} \\
&= \frac{\Pr(\vec{i} \mid A(z))\Pr(A(z))}{\sum_z \Pr(A(z))\Pr(\vec{i} \mid A(z))} \\
&= \frac{\prod_t \Pr(i_t \mid A(z))\Pr(A(z))}{\sum_z \Pr(A(z)) \prod_t \Pr(i_t \mid A(z))} \\
&= \frac{\prod_t p_{t,i_t}(z)\Pr(A(z))}{\sum_z \prod_t p_{t,i_t}(z)\Pr(A(z))}.
\end{aligned}$$

The standard "know-nothing prior" assumption $\Pr(A(z)) = 1/|Z|$ lets us simplify this even further to

$$\Pr(A(z) \mid \vec{i}) = \frac{\prod_t p_{t,i_t}(z)}{\sum_z \prod_t p_{t,i_t}(z)}.$$

Note that the global independence assumption not only creates a simple product over t but also makes the value independent of the order of presentation of the data for each t. Thus the Bayesian probability of $Z = z$ is just the normalized likelihood function.

Write N_z for $\prod_t p_{t,i_t}(z)$. Upon iterating the data d times, we get

$$\Pr(A(z) \mid \vec{i}^d) = \frac{N_z^d}{\sum_z N_z^d}.$$

Because $a^d = o(b^d)$ whenever $a < b$ and the vector of values N_z is finite, the right-hand side as $d \to \infty$ converges pointwise to the Dirac delta function for the value z maximizing N_z, which is just $z = z_{ML}$ as before. (This also holds true under any fixed prior $A(z)$.)

Thus the peaks of the Bayesian probability curves approach the ML estimators. Large homogeneous training sets can be expected to behave like d-fold iterations of a smaller training set. Thus in this application, it seems that ML already captures the objectives of Bayesian iteration.

Note that in both cases, only the probabilities of the selected options m_{t,i_t} are involved in the formulas. The ordinal information in i_t is not used. The basis for the critique in this paper is that the ML and Bayesian approaches are not using all available information.

We move on to simple frequentist approaches. We define $d(x, y)$ to be (the square of) a distance function, not necessarily supposing $d(x, y) = (x - y)^2$ for use with least-squares estimation. Since there is no notion of "same outcome," the issue becomes how best to preserve the intuition of building frequencies for the outcomes. One idea is to impose a percentile grid on them.

3.2 Percentile Fitting

The "Percentile Fitting" method of [RH11] attempts to avoid these choices and weighting issues. The method is to minimize a distance integral of the form

$$\int_{q=0}^{q=1} d(q, f_q(z))$$

where $f_q(z)$ is the *hit score for percentile* q defined as follows. The hit score is the average of the hit scores for each tuple t, so suppressing t we need only define $f_q(z)$ for one vector of projected probabilities $P(z) = (p_1(z), p_2(z), \ldots, p_\ell(z))$. Here is where the fixed ordering of outcomes is used. Let $i = i_t$ be the selected outcome for that tuple. Define

$$p = \sum_{j=1}^{i-1} p_j(z); \qquad r = p + p_i(z),$$

$$f_q(z) = \begin{cases} 1 & \text{if } q \geq r \\ \frac{q-p}{r-p} & \text{if } p \leq q \leq r \\ 0 & \text{if } q \leq p. \end{cases}$$

Here is the frequentist intuition. Consider any fixed value of q, say $q = 0.60$, and consider any projected tuple $(p_1(z), p_2(z), \ldots, p_\ell(z))$. The parameter(s) z represent a way of stretching or compressing sub-intervals of width $p_k(z)$ in the unit interval. Let us suppose first that q is exactly at the upper end of interval p_k, meaning that $p_1(z)+p_2(z)+\cdots+p_k(z) = q$. Then we interpret z as representing the assertion that the probability of one of the first k options being chosen is exactly q. That is to say, if $i \leq k$ then we call the tuple a "hit," else it is a "miss." So this particular z is asserting that the probability of a hit is q, and that is the z that we wish to find.

If q sits midway inside interval p_k, then we must consider how to score z in the case $i = k$. To interpolate correctly, let b be the ratio of the real-number distance of q from the left end of the interval to its width p_k. Then score this case as b of a hit. Thus z and q represent the assertion that the expected hit score for the tuple *at* percentile q is none other than q itself.

For each z and q, this prescription defines a criterion for scoring a hit for each tuple, and asserts that this expectation is q. Since the expectation is the same for each tuple, we have intuitively achieved the effect of the simple-frequency case, and can aggregate over the tuples. The frequency function $f_q(z)$ defined above tabulates the actual hit scores from the data. The degree of fit given by z for percentile q is then quantified as the distance between q itself and $f_q(z)$.

Treating q itself as a continuous parameter leads to minimizing the above integral. The one arbitrary choice we see is whether this should be weighted in terms of q. Minimizing

$$\int_{q=0}^{q=1} H(q) d(q, f_q(z))$$

instead is natural because having $H(0) = H(1) = 0$ reinforces the idea that the hit percentage projections are automatically correct at the endpoints $q = 0$ and $q = 1$. Apart from this, our intuitive point of using percentiles is that they skirt issues of skedasticity. We abbreviate this method as PF.

3.3 Fitting to Equate Means for BC and AE

The projected probabilities also yield a *projected utility* $u(z) = \sum_j u_j p_j(z)$. This can readily be summed or averaged over all tuples. Thus one can also fit z by equating the projected $u(z)$ with the actual utility u achieved in the training data. This is affinely related to minimizing the AE metric defined above.

In cases where the objective is to see how often the agent makes the optimal choice, as well as modeling its average (falloff in) utility (from optimal), one can write two equations in the parameters Z. When Z comprises just two parameters, one can fit by solving two equations in two unknowns, equating $u = u(z)$ and the first-choice hit frequency $h_1 = |\{\, t : i_t = 1 \,\}|/T$ with the average of $p_{t,1}(z)$. This hybrid fitting method bypasses all of the above options, and hence acts as a helpful check on them. We abbreviate it FF for "first-choice and falloff."

3.4 Fitting the Ordinal Ranks

We can fit to minimize the ORF statistic, or alternatively its frequency-weighted version $\sum_k f_k(f_k - \hat{f}_k)^2$. For data such as below where about half the "mass" is on index 1—that is, when the best answer or trade or chess move is found at least half the time (at least when items have unit

weights)—the latter policy is a compromise on simply solving to make the BC projection agree with the sample mean. The compromise policy avoids overfitting, and avoids heteroskedasticity issues with ORF itself.

Still, we study the fitting policy of minimizing ORF to emphasize the importance of getting accurate projections for the "tail" of the distribution of choices: bad mistakes on exams, disastrous trades, "blunders" at chess. We call this policy IF for "index fit," and the frequency-weighted version IM for "index mass." We could also consider hybrids of FF and IF or IM, but stop short of doing so in order to highlight the simpler comparisons.

3.5 Model Tuning

There is a larger issue of whether differences between fitting methods and metrics is an indication of the underlying model itself being out-of-tune. One may be trying to fit a curve that simply does not have the right "shape." In cases where there is no clear-cut notion of "tuning," these considerations can become subjective. However, when there is a fitting method that provides reasonably low (here meaning good) scores on all the fitting distance metrics being considered, then we can take that as a measure of "goodness of tune." Poor scores for a different fitting method then can be regarded as pertaining more to the method than to the model. We intend this paper to address those situations in which perfection of modeling is far from being attained, yet reasonably accurate results are obtainable under suitable choices of methods.

4 Some Theoretical Considerations

In case the parameter space for Z allows a dense set of probability vectors, the simple case of repeated data (or equal utility vectors) allows exact fitting, and gives the same optimal z under any method.

Theorem 4.1 *Percentile Fitting agrees with Maximum Likelihood for homogeneous data and free parameterization of probabilities.*

Proof. Take $f_1, \ldots, f_\ell$ to be the frequencies from the data. The logsum minimand for ML then becomes

$$e(z) = f_1 \ln(1/p_1(z)) + f_2 \ln(1/p_2(z)) + \cdots f_\ell \ln(1/p_\ell(z)).$$

This is known to be minimized by setting $p_j(z) = f_j$ for each j, in accordance with the basic frequency idea, and the freedom assumption on z allows this to be realized. It remains to show that PF achieves the same minimum z.

For this z, let q be any percentile. If q falls on the endpoint r of any interval $p_k = p_k(z)$, then as $r = p_1 + p_2 + \cdots p_k = f_1 + f_2 + \cdots f_k$, the training data gives $f_q(z) = r = q$. Since other values of q occupy the same medial position in the same interval over all of the equal tuples, the interpolation gives $f_q(z) = q$ for all q, so the PF minimand is zero.

Also $h_1 = f_1 = p_1(z)$ and $u = \sum_j u_j f_j = \sum_j u_j p_j(z) = u(z)$, so the FF equations hold.

4.1 Differences Among ML/Bayes and PF

Now we give an example showing that this equivalence can be disturbed by constraining the parameters. Indeed it seems to be the simplest example that can possibly show a difference. Let each tuple have outcomes m_1, m_2, m_3, and let the probabilities be given by $p_1(z) = p_2(z) = z$, $p_3(z) = 1 - 2z$, with one numerical parameter $z \in [0, 1]$. Consider training data with $t = 2$ equal tuples, in which m_1 and m_3 are chosen once each.

The ML maximand is $z(1 - 2z)$ and is maximized at $z = 1/4$.

Percentile fitting gives a different answer, however. The PF minimand is a three-piece integral. The first piece integrates $d(q, \frac{1}{2}\frac{q}{z})$ from $q = 0$ to $q = z$. The second piece integrates $d(q, \frac{1}{2})$ from $q = z$ to $q = 2z$. The third piece integrates $d(q, \frac{1}{2} + \frac{1}{2}\frac{q-2z}{1-2z})$ from $q = 2z$ to $q = 1$. For $d(x, y) = (x - y)^2$, symbolic computation with *Mathematica*^{TM} shows this is minimized for $z = 3/10$, not $z = 1/4$.

5 Experimental Domain

We adopt the chess model of Regan and Haworth [RH11], for several reasons:

1. Chess games and results are public—there are no copyright or privacy considerations.
2. The decisions are taken under conditions of actual competition, not simulations.
3. The human subjects have similar training and backgrounds, and the games have no systematic or substantial outside influences (put another way, the modeling can be free of "nuisance terms").
4. There is a well-defined skill metric, namely the chess Elo rating system.
5. The utility values are assigned by a recognized human-neutral authority, namely the champion chess program Rybka 3 [RK08].
6. The data sets are unique—comprising *all* games recorded between players with ratings e near the same Elo century mark in official chess events under "standard individual tournament conditions" in a specified time period. There is no arbitrariness of choosing data from certain kinds of exams or kinds of financial markets.
7. The data sets and statistical analyzing code are freely available by request, though they are not (yet) public.
8. The training sets each contain over 5,000 data points, some over 20,000.

The model has two parameters, called s for "sensitivity" and c for "consistency." The items use the error quantities $\delta_i = u_1 - u_i$ for fitting (they index from $i = 0$ rather than $i = 1$). The probabilities are projected by implicitly solving the equations

$$\frac{\log(1/p_i)}{\log(1/p_1)} = e^{-(\frac{\delta_i}{s})^c}$$

along with $\sum_i p_i = 1$. Further details about how the δ_i values are obtained and about the composition of the training sets are available in [RH11, RMH11]. The regressions published in those papers to compute an "Intrinsic Elo Rating" $E(z) = E(s, c)$ from a training set are not current with the provided software as of 12/18/2012, even besides the change from PF to FF as the reference fitting method between the above papers. However, the differences are not large enough to be a concern—and the regressions for $e = E(z)$ in these papers are weak anyway with only six points $e = 2200, 2300, 2400, 2500, 2600, 2700$ (projections below 2200 are extrapolated). Again we are using $E(z)$ only as a rough check.

We used the training sets for the years 2006–2009, for ratings E from 2700 all the way down to 1600. This ensures some variety in the levels of human subjects, which gives more weight to common phenomena. We ran each of five fitting methods FF, PF, ML, IF, and IM, each with unit and with entropy weights, on the set for each century point. We have room to report the results for 2700, 2400, 2100, and 1800; the others were substantially the same.

6 Comparison of Fitting Methods: Empirical Results

The legends on fitting methods are FF for solving the two equations $\hat{f}_1 = f_1$ and $\hat{ae} = ae$ in the two unknowns s, c (guaranteeing unbiased estimators for BC and AE), PF for percentile fit, ML for maximum likelihood, IF for index-fit without skedasticity adjustment, and IM for index-mass (with adjustment). The Nelder-Mead minimization method of [G$^+$09] was applied for minimization to precision 0.0001; hence the tables report four decimal places.

Table 1: Comparing fits with unit weights

2700 set: 6,892 turns				
Fit	$E(z)$	ORF	$\times\sigma_{BC}$	$\times\sigma_{AE}$
FF	2689	0.0499	-0.0004	0.0008
PF	2580	0.029	1.0815	-4.6792
ML	2418	0.2665	6.0971	-10.8041
IF	2466	0.0025	0.0393	-8.2837
IM	2459	0.0026	0.0066	-8.471
2400 set: 19,929 turns				
Fit	$E(z)$	ORF	$\times\sigma_{BC}$	$\times\sigma_{AE}$
FF	2421	0.0249	-0.0006	0.0018
PF	2335	0.0182	1.3522	-5.4644
ML	2087	0.1496	7.8135	-18.5049
IF	2275	0.0105	0.1571	-8.6114
IM	2243	0.0112	0.0422	-10.1977
2100 set: 9,728 turns				
Fit	$E(z)$	ORF	$\times\sigma_{BC}$	$\times\sigma_{AE}$

FF	2027	0.0217	-0.0019	-0.0034
PF	1969	0.0205	0.0727	-2.1879
ML	1653	0.1367	4.6025	-12.437
IF	1978	0.0204	0.085	-1.8592
IM	1941	0.0212	0.0375	-3.136
1800 set: 15,930 turns				
Fit	$E(z)$	ORF	$\times\sigma_{BC}$	$\times\sigma_{AE}$
FF	1729	0.0183	-0.0029	0.0049
PF	1686	0.0303	-1.2289	-1.7197
ML	1331	0.1074	3.1323	-14.6344
IF	1823	0.0145	-0.0363	4.2969
IM	1790	0.015	0.0256	2.7243

What stands out most is the poorness of the maximum-likelihood estimator as implemented above, especially for estimating the aggregate error (AE) statistic. The two index fits are also poor on this. Whereas, the FF test, simply solving two equations, achieves ordinal-rank fit scores that are close to the optima achievable by the IF (or IM) method. The percentile fit (PF) is fairly close in quality, but gives biased estimates. The differences are even greater under entropy weighting.

Table 2: Comparing fits with entropy weights

2700 set: 6,892 turns				
Fit	$E(z)$	ORF	$\times\sigma_{BC}$	$\times\sigma_{AE}$
FF	2681	0.069	-0.0005	0.0067
PF	2612	0.0667	0.5396	-3.9494
ML	1863	0.3832	5.9989	-28.5197
IF	2598	0.0449	-0.1216	-4.5738
IM	2608	0.0495	-0.0189	-4.0811
2400 set: 19,929 turns				
Fit	$E(z)$	ORF	$\times\sigma_{BC}$	$\times\sigma_{AE}$
FF	2400	0.0245	0.0001	0.0114
PF	2369	0.0256	0.4614	-2.5458
ML	1554	0.3268	8.924	-47.0187
IF	2399	0.0248	0.043	-0.021
IM	2371	0.0225	0.0477	-2.331
2100 set: 9,728 turns				
Fit	$E(z)$	ORF	$\times\sigma_{BC}$	$\times\sigma_{AE}$

FF	2000	0.0256	0.0009	0.004
PF	2000	0.0209	-0.9911	0.1092
ML	1138	0.3118	5.1245	-31.4804
IF	2127	0.0337	-0.0714	6.9611
IM	2105	0.0317	0.0299	5.6754
1800 set: 15,930 turns				
Fit	$E(z)$	ORF	$\times\sigma_{BC}$	$\times\sigma_{AE}$
FF	1704	0.0242	-0.0017	-0.0022
PF	1731	0.0225	-3.1544	1.8705
ML	820	0.3323	6.7039	-38.5982
IF	1989	0.0267	-0.4083	19.0187
IM	1984	0.0274	-0.0184	18.5923

7 Conclusions

The mystery that needs further elucidation is, why does Maximum Likelihood perform so badly here? Note that the results it gives are nevertheless self-consistent across the data spectrum. This is shown by the progression of $E(Z)$ figures for ML: 2700: 2418, 2400: 2087, 2100: 1653, 1800: 1331, and similarly for the unreported runs in-between. If used in isolation, with its own confidence intervals, it would provide a workable model of skill unto itself, re-calibrating the $E_{ML}(Z)$ figures accordingly. One might not be aware of how it is "out-voted" by the other fitting methods in absolute terms. This owes to the general setting presented in this paper being amenable to a wealth of fitting techniques and separate metrics for evaluating their quality.

We hope to spur from this a deeper comparative examination of methods used in psychometric test scoring (such as [WH01, ON06], and financial analysis, among other applications. We also speculate that the skill assessment methods used in chess–which expressly correspond to statistical win-loss results in competition—can be carried over to these other domains, exploiting the rigor and definiteness of the chess model and its data.

References

[B$^+$13] Graham Banks et al. CCRL rating lists. http://www.computerchess.org.uk/ccrl/, 2013.

[Elo78] Arpad Elo. *The Rating of Chessplayers, Past and Present*. Arco Pub., New York, 1978.

[G$^+$09] M. Galassi et al. *GNU Scientific Library Reference Manual (3rd Ed.)*. GNU, 2009.

[ON06] R. Ostini and M. Nering. *Polytomous Item Response Theory Models*. Sage Publications, Thousand Oaks, California, 2006.

[RH11] K. Regan and G.M^{c}C. Haworth. Intrinsic chess ratings. In *Proceedings of AAAI 2011, San Francisco*, 2011.

[RK08] V. Rajlich and L. Kaufman. Rybka 3 chess engine, 2008. http://www.rybkachess.com.

[RMH11] K. Regan, B. Macieja, and G. Haworth. Understanding distributions of chess performances. In *Proceedings of the 13th ICGA Conference on Advances in Computer Games*, 2011. Tilburg, Netherlands.

[WH01] F. Wichmann and N. Jeremy Hill. The psychometric function: I. Fitting, sampling, and goodness of fit. *Perception and Psychophysics*, 63:1293–1313, 2001.