GLICKO-BOOST
Deloitte/FIDE Chess Rating Challenge

Prof. Mark E. Glickman
Boston University

1 Overview

The rating system I entered into the Deloitte/FIDE Chess Rating Challenge, the "Glicko-boost" system, is a substantial extension of the Glicko system. The Glicko system (Glickman, 1999) is a rating system I invented in the mid-1990s in which each player is characterized at any time by two parameters: a rating, and a "ratings deviation" (RD). The RD, the main innovation of the Glicko system, is a measure of the uncertainty in a player's rating. The greater a player's RD, the more uncertainty exists about the player's ability. The implication for rating updates is that players with large RDs typically incur large rating changes, and opponents of players with large RDs tend not to be impacted much by the game results against such players. The Glicko system has been in wide use on various chess servers, online gaming systems, as well as for non-gaming applications.

The Glicko-boost system addresses several arguable weaknesses in the most commonly used form of the Glicko system. The approach I constructed has the following features:

- The system allows for an advantage to white.
- If a player's performance was exceptional in a given month, then the pre-month RD would be increased (that is, "boosted") to a higher value and the player would be re-rated. This accounts for the possibility a player with a low RD but who is improving quickly over time may be under-rated, and needs an extra increase to his/her rating. This concept is the foundation of the so-called Glicko-2 system, but has been modified for a simple implementation in the Glicko-boost system.
- The system updates players' ratings once, and then updates the initial ratings a second time using the opponents' ratings and RDs from the first update. Thus, if a player's opponent had a strong performance during a given month relative to his/her rating, the player's rating would be updated relative to a higher opponent rating because of this 2-pass algorithm. Analogously, if the player's opponent had a weak performance, the player's rating would be updated relative to a lower opponent's rating. In other words, a player's rating change would be based on more information about the opponents' ability than simply the opponents' pre-month ratings, as with the Glicko system.
- Instead of the RD increasing over time by a constant amount, as in the Glicko system, the magnitude of the increase can depend on the value of the RD itself, and also on the player's rating. This allows for a more flexible way in which rating uncertainty increases over time.

In particular, because stronger players may (on average) have more stable strengths, the RD increase for higher-rated players is, in general, slightly smaller than for lower-rated players.

An outline of the rating computation is as follows. The details of the computation are described in Section 2. For the steps below, we assume that each player has a rating and an RD prior to competing for a given month.

1. Perform Glicko updating (with a white advantage) for all players during the current month.

2. Perform Glicko updating (with a white advantage) again for each player using the pre-month ratings and RDs, but use the *opponents'* ratings and RDs from Step 1.

3. Based on the resulting ratings and RDs from Step 2, determine for each player whether the total score exceeds the expected score by a "statistically significant" amount (as described in Section 2.2). If so, then reset the player's pre-month RD to a larger value specified in the formulas; otherwise, make no changes to the RD.

4. Re-perform Step 1 of the algorithm (using the RDs determined from Step 3).

5. Re-perform Step 2 of the algorithm based on the results of Step 4. The resulting ratings and RDs are the post-month parameter values based on the game outcomes.

6. Update the players' RDs due to the passage of time by increasing the RD as a function of the RD itself and the player's post-month rating.

The sequence of six steps is applied as each month of game data is accumulated.

2 Details of the algorithm

The Glicko-boost system depends on three main component modules, which are discussed in this section. A slight revision of the Glicko updating algorithm that addresses the advantage for white is described in Section 2.1. The method to boost a player's pre-month RD based on an exceptional performance is presented in Section 2.2. Finally, the method for increasing players' rating deviations due to the passage of time is described in Section 2.3. In Section 2.4, the details of the entire rating algorithm are described with reference to the three preceding component modules.

2.1 Glicko updating with a white advantage parameter

Assume that a player with rating r and rating deviation RD competes in J games in a particular month, with the opponents in each game having ratings $r_1, \ldots, r_J$ and rating deviations $\mathrm{RD}_1, \ldots, \mathrm{RD}_J$. It is possible that the player has competed against the same opponent multiple times, in which case the value of r_j and RD_j is the same for those games. Also, let $s_1, \ldots, s_J$ be the scores (0, 0.5, 1) for these games. For $j = 1, \ldots, J$, let

$$w_j = \begin{cases} 1 & \text{if the player had white in game } j \\ -1 & \text{if the player had black in game } j \end{cases}$$

Further, let η be the rating advantage for playing white (a system parameter that needs to be estimated). Then to update the player's rating and rating deviation to r' and RD', respectively, based on the game outcomes, compute

$$\begin{aligned} \mathrm{RD}' &= \sqrt{\left(\frac{1}{\mathrm{RD}^2} + \frac{1}{d^2}\right)^{-1}} \\ r' &= r + (\mathrm{RD}')^2 q \sum_{j=1}^{J} g(\mathrm{RD}_j)(s_j - \mathrm{E}(\eta, w_j, r, r_j, \mathrm{RD}_j)) \end{aligned}$$

where

$$\begin{aligned} q &= \frac{\ln 10}{400} = 0.0057565 \\ g(\mathrm{RD}_j) &= 1/\sqrt{1 + 3q^2(\mathrm{RD}_j^2)/\pi^2} \\ \mathrm{E}(\eta, w_j, r, r_j, \mathrm{RD}_j) &= \frac{1}{1 + 10^{-g(\mathrm{RD}_j)(r + w_j\eta - r_j)/400}} \\ d^2 &= \left(q^2 \sum_{j=1}^{J} (g(\mathrm{RD}_j))^2 \mathrm{E}(\eta, w_j, r, r_j, \mathrm{RD}_j)(1 - \mathrm{E}(\eta, w_j, r, r_j, \mathrm{RD}_j))\right)^{-1}. \end{aligned}$$

The only difference between these formulas and the original Glicko updating formulas is the replacement of the rating difference $(r - r_j)$ in the computation for the function E() with a term that accounts for the advantage to white, $(r + w_j\eta - r_j)$.

2.2 RD boost based on exceptional performances

The following computation results in the inflation of a player's RD if the player's performance in a given month was exceptional relative to what was expected. Assuming the notation in Section 2.1, let

$$Z = \frac{\sum_{j=1}^{J} g(\mathrm{RD}_j)(s_j - \mathrm{E}(\eta, w_j, r, r_j, \mathrm{RD}_j))}{\sqrt{\sum_{j=1}^{J} g(\mathrm{RD}_J)^2 \mathrm{E}(\eta, w_j, r, r_j, \mathrm{RD}_j)(1 - \mathrm{E}(\eta, w_j, r, r_j, \mathrm{RD}_j))}}$$

which is a standardized measure (a so-called "z-score") of the degree to which the player outperformed the expected score. The z-score is *approximately* normally distributed with mean 0 and standard deviation 1. Based on this computation, RD', the revised value of RD (the original pre-month rating deviation), is computed as

$$\mathrm{RD}' = \begin{cases} \mathrm{RD} & \text{if } Z \leq k \\ (1 + (Z - k)B_1)\mathrm{RD} + B_2 & \text{if } Z > k \end{cases}$$

where the threshold value k was chosen to be 1.96, the value at which 2.5% of the time Z would be larger just by chance. The system parameters B_1 and B_2 are constrained to be positive, and are to be estimated as system parameters. Note that if B_1 and B_2 are estimated to be close to 0, the value of RD changes by a negligible amount if a player's performance is exceptional, and larger values of these parameters results in larger boosts in RD.

2.3 RD increase over time

To account for the increased uncertainty in ability due to the passage of time, the original Glicko system updates a player's RD (after updating from game results that month) by the formula

$$\text{RD}_{new} = \sqrt{\text{RD}^2 + c^2},$$

clipping the value at 350 if the above computation produced a larger value. The current system, instead, uses

$$\text{RD}_{new} = \sqrt{\text{RD}^2 + \exp(\alpha_0 + \alpha_1\text{RD} + \alpha_2\text{RD}(r/1000) + \alpha_3(r/1000) + \alpha_4(r/1000)^2)}.$$

where r is the player's current rating, and $\alpha_0, \alpha_1, \ldots, \alpha_4$ are system parameters that are to be estimated. Clipping of this rating deviation is relative to a system parameter (as described below) that is estimated in the optimization process. The above formulation allows the increase in RD to depend on the player's rating and on the RD itself.

2.4 Algorithm implementation

Below are the steps of the Glicko-boost system in more complete detail. Assume at the start of month t we have ratings r_i and rating deviations RD_i for a subset of players. For players without a rating or RD, use values r_{unr} and RD_{unr}, which are system values that are to be estimated. For players with a FIDE rating but without an RD (which occurs in the contest at $t = 1$), use RD_{30} for players that have competed in at least 30 FIDE games, and use RD_{29} for those having played 29 or fewer games. Again, these values, which have been constrained in the system by the following inequality

$$\text{RD}_{30} \leq \text{RD}_{29} \leq \text{RD}_{unr},$$

are to be estimated through the optimization procedure. Once all players have actual or imputed ratings and RDs, carry out the following steps. Assume players $1, 2, \ldots, J$ are involved in competition during month t.

1. Letting $r_1, \ldots, r_J$ be the ratings at the start of month t, and $\text{RD}_1, \ldots, \text{RD}_J$ be the RDs at the start of month t, determine the updated ratings and RDs based on Glicko updating (see Section 2.1). Denote these values $r'_1, \ldots, r'_J$ for the ratings, and $\text{RD}'_1, \ldots, \text{RD}'_J$ for the RDs.

2. For each player i, perform Glicko updating using the player's pre-month t rating and RD (r_i and RD_i), but using the opponents' ratings and RDs resulting from Step 1 (that is, the r'_j and RD'_j). Denote the resulting updates $r^*_1, \ldots, r^*_J$ and $\mathrm{RD}^*_1, \ldots, \mathrm{RD}^*_J$.

3. With ratings and RDs $r^*_1, \ldots, r^*_J$ and $\mathrm{RD}^*_1, \ldots, \mathrm{RD}^*_J$ as the inputs, perform the RD boost computation of Section 2.2 for each player to determine revised RDs. Denote the recomputed rating deviations $\mathrm{RD}^\dagger_1, \ldots, \mathrm{RD}^\dagger_J$. If any of these values is larger than RD_{unr}, set them to RD_{unr}.

4. Re-perform Glicko updating using ratings $r_1, \ldots, r_J$ (pre-month ratings) and rating deviations $\mathrm{RD}^\dagger_1, \ldots, \mathrm{RD}^\dagger_J$ (RDs from Step 3). Denote the resulting updated ratings $r^+_1, \ldots, r^+_J$ and rating deviations $\mathrm{RD}^+_1, \ldots, \mathrm{RD}^+_J$.

5. As in Step 2, re-perform Glicko updating for each player i using rating r_i and rating deviation $\mathrm{RD}^\dagger_i$, but using the r^+_j and RD^+_j for the opponents' ratings. Call the resulting ratings $r^{fin}_1, \ldots, r^{fin}_J$ and rating deviations $\mathrm{RD}^{fin}_1, \ldots, \mathrm{RD}^{fin}_J$, which are the final updated ratings due to the game outcomes.

6. To update the RDs due to the passage of time between month t and month $t+1$, apply the RD increase algorithm of Section 2.3 to all players (not just those who competed in month t). For those who competed in month t, apply the RD increase formula using ratings $r^{fin}_1, \ldots, r^{fin}_J$ and rating deviations $\mathrm{RD}^{fin}_1, \ldots, \mathrm{RD}^{fin}_J$ as the inputs. For those who did not compete in month t, use the ratings and RDs from the start of month t as the inputs. This will result in a new set of RDs for use at the start of the following month, which can be relabeled $\mathrm{RD}_1, \ldots, \mathrm{RD}_J$. If any of the computed RDs is larger than RD_{unr}, then set these RDs to RD_{unr}.

   The ratings at the start of month $t+1$ are either $r^{fin}_1, \ldots, r^{fin}_J$ for the players who competed in month t, or the ratings at the start of month t for those who did not compete.

3 Optimization

The Glicko-boost system described in Section 2.4 involves 12 system parameters that needed to be estimated; a white advantage parameter (η), two RD boost parameters (B_1 and B_2), five RD increase parameters ($\alpha_0, \ldots, \alpha_4$), a default rating for unrated players (r_{unr}), and three default RD parameters (RD_{unr}, RD_{30}, and RD_{29}). The following briefly explains the optimization approach I used to estimate the Glicko-boost system parameters.

3.1 Use of the primary, secondary and tertiary game data

All three supplied data sets were used for rating players as well as estimating all system parameters. Rather than combining all three data sets into one large data set, the rating update and optimization procedure recognized two features of the secondary and tertiary data sets to incorporate them distinctly from the primary data set.

- Color information was lacking for the secondary and tertiary data sets.
- The secondary and tertiary data sets were of possibly worse quality or lesser relevance to making predictions to the test data than the primary data set.

The lack of color information in secondary and tertiary data sets was addressed in a straightforward manner. In the rating algorithm, when computing the E() function in the Glicko component algorithm of the rating system, the color indicator variable w_j was forced to be 0 for games in the secondary and tertiary data sets. This ensured that the E() function always involved only the difference in players' ratings, $r - r_j$, rather than an explicit white advantage in the form of $r + w_j\eta - r_j$.

To address the lesser relevance of games in the non-primary data sets, the games within each data set received weights depending on their data set membership. The games in the primary data set received full weight (i.e., a weight of 1). Games in the secondary data set were weighted by a factor Wt_2 in the computations, and the games in the tertiary data set were weighted by a factor Wt_3, both being values that were to be estimated through the optimization procedure. More specifically, the expression $g(\mathrm{RD}_j)$ in the Glicko updating algorithm, which itself can be understood as a weight for game j based on the opponent's rating uncertainty, was replaced by $(\mathrm{Wt}_2 g(\mathrm{RD}_j))$ whenever the game came from the secondary data set, and by $(\mathrm{Wt}_3 g(\mathrm{RD}_j))$ whenever the game came from the tertiary data set. The values Wt_2 and Wt_3 were constrained to be between 0 and 1.

The incorporation of the secondary and tertiary data sets therefore added two more parameters (Wt_2 and Wt_3) to estimate in the optimization process, though these parameters were only used to help optimize the system parameters and were not of interest themselves as part of the Glicko-boost system.

3.2 Optimization criterion and implementation

While several different optimization criteria were considered, the one used in the submitted entry involved the following procedure.

1. For a set of selected system parameters and weight parameters, run the Glicko-boost rating system (on all three data sets, appropriately weighted) for months 1 through 129. Obtain the ratings and RDs for all players relevant for predicting games in month 130.
2. Only for the primary data set, compute the expected scores of the games (see below) in month 130. Retain all ratings, and inflate all the RDs by the RD inflation algorithm in Section 2.3 to obtain the appropriate ratings and RDs for predictions of games in month 131 of the primary data set, and then compute expected scores of games in month 131. Recursively, do the same procedure to obtain expected scores for games in month 132 in the primary data set. Note that the predictions for month 131 do not involve information on game outcomes from month 131, and similarly predictions in month 132 do not use game outcome information from months

System Parameter	Optimum Value	Description
η	30.0	Advantage to white
B_1	0.20139	RD boost multiplicative factor
B_2	17.5	RD boost additive factor
α_0	5.83733	RD increase: intercept
α_1	−1.75374e-04	RD increase: RD factor
α_2	−7.080124e-05	RD increase: RD× rating factor
α_3	0.001733792	RD increase: rating factor
α_4	0.00026706	RD increase: rating-squared factor
r_{unr}	1946.25	Default rating for unrated players
RD_{unr}	250.0	Default RD for unrated players
RD_{30}	250.0	RD for rated players with 30+ games
RD_{29}	250.0	RD for rated players with fewer than 30 games
Wt_2	1.0	Weight for secondary data set
Wt_3	0.5005	Weight for tertiary data set

Table 1: Optimized system parameter values using the Nelder-Mead algorithm.

130 and 131. This sequence of computing expected scores mimicked the sequence necessary in the test data set.

3. Identify the players (real and fictitious) involved in games in months 133 to 135 in the test data set, and compute the binomial deviance criterion on the subset of games from months 130 to 132 restricted to this subset of players. This (averaged) binomial deviance was the optimization criterion used for estimating the system and weighting parameters.

The expected score for a game between players i and j with ratings r_i and r_j and rating deviations RD_i and RD_j was approximated by

$$\text{Expected Score} = \frac{1}{1 + 10^{-g(\sqrt{\text{RD}_i^2 + \text{RD}_j^2})(r_i + w_{ij}\eta - r_j)/400}}.$$

This approximation to the true expected score has been demonstrated to be reasonably accurate in Monte Carlo simulations.

The optimization was carried out in R using the `optim` function, using the Nelder-Mead algorithm to optimize over the system and weighting parameters. The Nelder-Mead algorithm is a reasonably robust iterative procedure that does not require continuous second derivatives of the function being optimized, as is the case with more conventional gradient-based methods. The core functions to run the rating system for the 132 months of data on all three data sets were implemented in R, as well as Fortran functions called from within R. The results of the optimization procedure are summarized in Table 1.

Several aspects of the results are worthy of comment. The advantage to white is estimated to be about 30 points, similar to that of the contest Glicko benchmark. The RD boost parameters of $B_1 =$

Player, Rating (RD)	A	B	C	D	E	F	G	H	Total
Player A, 2300 (140)	×	W 0	B 0		W 0	B 1	W 1	B 0	2.0
Player B, 2295 (80)	B 1	×	W $\frac{1}{2}$	W 1	B $\frac{1}{2}$	W 1	B 1		5.0
Player C, 2280 (150)	W 1	B $\frac{1}{2}$	×	B 1		B 1	W 1	W $\frac{1}{2}$	5.0
Player D, 2265 (70)		B 0	W 0	×	W 0	B $\frac{1}{2}$	W 0	B 0	0.5
Player E, 2260 (90)	B 1	W $\frac{1}{2}$		B 1	×	W 1	B $\frac{1}{2}$	W 0	4.0
Player F, 2255 (200)	W 0	B 0	W 0	W $\frac{1}{2}$	B 0	×		B 0	0.5
Player G, 2250 (50)	B 0	W 0	B 0	B 1	W $\frac{1}{2}$		×	W 0	1.5
Player H, 2075 (120)	W 1		B $\frac{1}{2}$	W 1	B 1	W 1	B 1	×	5.5

Table 2: Game results among eight fictitious players labeled A through H, followed by each player's rating and RD. Each entry in the table indicates the color and the game result against the player in the corresponding column.

0.20139 and $B_2 = 17.5$ indicated a moderate increase to players' RDs who perform exceptionally well in a month; for example, a player who performed 3 standard deviations better than expected (based on the second iteration of Glicko updating) would experience a $(1+(3.0-1.96)) = 1.21$ or a 21% increase in RD (plus the additional $B_2 = 17.5$) before updating the ratings again. The initial rating for all unrated players was estimated at 1946 with a corresponding RD of 250. The initial RD did not seem to depend on whether the player was unrated, or whether the player had a FIDE rating. It is also of interest to note that the secondary data set was estimated to have full weight and that the tertiary data set was estimated to have approximately half weight.

4 Example application

To illustrate the Glicko-boost rating system, we demonstrate its application on a fictitious tournament. Assume eight players (labeled A through H) compete in six games, three each as white and as black. For this illustration, we assume that these are the only games each competitor plays in a given month. The players' initial ratings and RDs, and the game results, are displayed in Table 2.

The players are listed in rating order (as opposed to the order based on the final total). The example was constructed to highlight the rating movement across the individual steps of the algorithm. The first seven players have ratings within 50 points of each other, and the eighth player has a much

Player	Initial Rating (RD)	Step 1 Rating (RD)	Step 2 Rating (RD)	Reset RD	Step 3 Rating (RD)	Final Rating (RD)	RD time increase Rating (RD)
A	2300 (140)	2209 (104.3)	2223 (103.3)	140	2210 (104.5)	2230 (103.4)	2230 (105.0)
B	2295 (80)	2344 (70.9)	2338 (71.0)	80	2344 (70.9)	2338 (71.0)	2338 (73.3)
C	2280 (150)	2386 (107.5)	2379 (107.0)	150	2387 (107.8)	2385 (107.2)	2385 (108.8)
D	2265 (70)	2205 (63.8)	2209 (63.6)	70	2205 (63.8)	2211 (63.7)	2211 (66.3)
E	2260 (90)	2287 (77.7)	2283 (77.4)	90	2288 (77.8)	2287 (77.5)	2287 (79.7)
F	2255 (200)	2051 (121.7)	2075 (120.1)	200	2053 (122.0)	2082 (120.7)	2082 (122.0)
G	2250 (50)	2232 (47.5)	2235 (47.4)	50	2232 (47.5)	2236 (47.5)	2236 (50.9)
H	2075 (120)	2280 (98.6)	2265 (97.0)	**153.1**	2353 (114.7)	2330 (112.2)	2330 (113.7)

Table 3: Step-by-step results of Glicko-boost applied to game results in Table 2.

lower rating. This last player, who performed extremely well relative to his rating, demonstrates the effect of the RD boost portion of the algorithm. The remaining players highlight other aspects of the rating system and algorithm, including the effect of different RDs on the pre-post tournament rating changes, and the effect of the second pass of the re-rating (that is, Steps 2 and 4). The RDs in the example are generally large, corresponding to players who most likely do not compete frequently, so that rating changes would be expected to be sizable. The results of Glicko-boost, displayed step-by-step, are shown in Table 3.

The column in Table 3 labeled "Step 1" shows the result of applying Glicko updating (with the 30-point white advantage). The larger rating changes tend to correspond, as expected, to players with large RDs. Players A, D, and F have much poorer performances than their initial rating would predict, so the Step 1 rating drops accordingly. Meanwhile players B, C and H have much better performances than predicted, and therefore have large rating increases in Step 1. In the case of player H, would had an extraordinary performance, the rating increase is over 200 points (the magnitude partially explained by the large initial RD of 120).

In Step 2, the players' initial ratings are updated relative to the opponents' Step 1 ratings and RDs. It is interesting to note how the results differ from the ratings derived from Step 1. In every case, the rating changes from the initial ratings to Step 2 are less extreme than the change to Step 1. This is because the Step 2 ratings factor in the opponents' results. So, for example, while player F's rating dropped from 2255 to 2051 in Step 1 based on very poor results, the 2075 Step 2 rating recognizes that his opponents may have been stronger than their initial ratings indicate.

The column labeled "RD Reset" is either the initial RD, or a boosted RD if the player's performance was exceptional. Only player H met the criterion to warrant a boost in RD, which increased his initial RD from 120 to 153.1.

Steps 3 and 4 then reapply the computations of Steps 1 and 2 using the initial ratings and the reset RDs (which are all the same as the initial RDs except for player H). The Step 4 ratings are labeled in the table "Final Rating (RD)." It is worth pointing out that if none of the RDs increased through the RD boost, then Steps 3 and 4 would be unnecessary to carry out and the results of Step 2 would be the final ratings and RDs. The results of Steps 3 and 4 are very similar, as expected, as those in

Steps 1 and 2, except that the greater initial uncertainty (larger RD) in player H's strength changes the computation slightly. The differences in ratings between Steps 1 and 3, and between Steps 2 and 4, are relatively small, except in the case of player H whose rating increased more dramatically due to the RD boost.

The final column in Table 3 reports the ratings and RDs to use as the initial player information for the following month. The column incorporates the RD increase due to the passage of time. As can be seen, the increase in uncertainty over one month is fairly small, increasing the RDs typically by 2-3 points.

5 Comments

This description concludes with comments on the modularity of the the Glicko-boost system, and some final thoughts.

5.1 Modularity of the Glicko-boost system

One of the appeals of the Glicko-boost system is its modularity. Several of the system's components were developed specifically to improve the predictability on the test set, though the increased complexity was made at the sacrifice of simplicity. However, some of the system's components can be simplified or even eliminated altogether to create a more accessible algorithm. Below are some ways in which the method could be simplified while still retaining the main overall features.

- Simplify the formula for RD increases over time. Based on the optimized values of $\alpha_1 \ldots, \alpha_4$, RD increases actually vary over a small range (the amount added to RD^2 in the formula in Section 2.3 tends to stay between 17.0 and 19.0) as a function of the current RD and rating. For implementing a practical system, using the original Glicko addition of c^2 per month would probably be sufficient.
- Eliminate repeated Glicko computations in the system. The Glicko-boost system improved predictability substantially by iterating the Glicko computations twice before the RD boost, and twice after the RD boost. Simplified versions of the Glicko-boost system could involve eliminating one iteration of either or both pairs of Glicko computations.
- Eliminate the RD boost formula. The RD boost formula is intended to detect players whose improvement is quicker than the ordinary Glicko formulas can track. While the RD boost is considered a fundamental feature of the described system, for a FIDE player population this may not be essential (though this would depend on the extent of FIDE's intended membership expansion), in which case the RD boost formulas could be eliminated from the proposed algorithm. Along with the RD boost elimination, the second pair of Glicko iterations could be eliminated as well.

It should be noted as well that the modularity of the system also allows for straightforward extensions, and not just simplifications. For example, initial ratings and RDs for unrated players could depend on other factors, such as age, or rating in a national federation converted appropriately to the FIDE scale. Also, for increased accuracy, further iterations of the Glicko updating could be considered.

5.2 Final thoughts

The formulation of the Glicko-boost system was the culmination of various attempts at constructing a reasonably simple extension of the Glicko system. In fact, the Glicko-boost system with the optimized parameters in Table 1 was not the system with the greatest predictability among the entries I submitted. I chose the method described in this document because it had the greatest face validity among the competitive entries I had developed. For example, one of the system components I had experimented with was the initial RD being a function of the number of games previously played and of the initial FIDE rating. I discovered that the optimal relationship involved RDs *increasing* as a function of the player's initial rating. This made little sense because higher-rated players tend to be more stable than lower-rated players, and if implemented would be counter to the system having face validity despite the system producing slightly better predictions. Similarly, I had considered an RD boost formula that increased a player's RD for exceptional performances, but also for extremely poor performances. Again, while this resulted in slightly better predictions, my sense was that politically it would be a mistake to accelerate the decline of a player's rating, so I removed this feature from the final system I entered into the contest.

Because the rating system was a complicated function of many system parameters that had interdependencies, the optimization function was sensitive to starting values. The optimization criterion function was likely multimodal, so that any application of the optimization routine could find a *local* optimum depending on the starting value, but not necessarily a *global* optimum. The optimized parameters in Table 1 came about by starting the optimization routine at values suggested through other optimized systems I tried, and that had intuitive appeal. By using different starting values, I noticed that parameters such as the relative weight of the tertiary data set varied between 0.3 and 0.6 without much affecting the optimization criterion. Furthermore, the initial RDs for unrated and rated players also varied quite a bit from different starting values, and across different rating systems, again without substantially affecting the optimum criterion values. Finally, different choices of the criterion function did not end up playing a large role. I tried using binomial predictive deviance criteria that were averaged over all games in the final two months of results, as well as weighted binomial predictive deviance criteria that weighted the deviances according to the frequency of games played by players in the training data, but again the optimal values did not appear to be very sensitive to these changes.

Bibliography

Glickman, Mark E. (1999), "Parameter estimation in large dynamic paired comparison experiments." Applied Statistics, 48, 377-394.