Chess Game Result Prediction System

Zheyuan Fan, Yuming Kuang, Xiaolin Lin, *Stanford University*
CS 229 Machine Learning Project Report, Autumn 2013

Abstrct

In this project we train World Chess Federation (FIDE) rating systems using a training dataset of a recent eleven-year period with games from 2000 chess players. We will then use our system to predict the outcome of chess games played by the same players in the following half year. Accuracy between predicted results and actual game results is the primary indicator of whether our approach is a practical chess rating system.

Background

World Chess Federation (FIDE) adopted Elo rating system since 1970 and this rating system has been the primary yardstick in the world to measure the strength of chess players. The Elo rating system also has many other applications in sports rankings. Despite the popularity of the Elo system, it has never been demonstrated that it is technically superior to other approaches. In this project we are going to investigate approaches that might do better than the Elo system. Such an investigation could have major implications on the theory and practice of ratings methodology.

Dataset

All game results and rating data are extracted from FIDE internal database and Chessbase database. We have collected the following data in a 135-month period of professional chess games from 2000 different chess players from year 2000 to year 2011:

1. Primary training dataset from the first 127 months to train our prediction system.
2. Secondary training dataset, which is used along with the primary data set to validate and tune parameters.
3. Initial rating list that provides an initial list of the involved players FIDE ratings and K-factor (player's game activity factor).
4. Test games dataset that identifies the chess games that we are predicting.

Method

Model - Hidden Markov Process

We model the game results as a Hidden Markov Process. We assume that each chess player i has a rating, or relative strength, in month t, denoted as $X_{i,t}$. We can't directly observe $X_{i,t}$, so it's the hidden state of the Hidden Markov Process. We can observe the chess game results Y_{t,j_1,j_2}, which denotes the result of the game between white player j_1 and black player j_2 in month t. Y_{t,j_1,j_2} can take 3 values, 1 for 'white player win', 0.5 for 'draw', 0 for 'white player lose'. The dynamics of $X_{i,t}$ and Y_{t,j_1,j_2} is summarized as

$$X_{i,t} = w_0\bar{X}_t + \sum_{l=1}^{k} w_l X_{i,t-l} + \epsilon_{i,t} \quad \epsilon_{i,t} \sim N(0, \sigma_{i,t}^2) \tag{1}$$

$$Y_{t,j_1,j_2} \propto \phi_{t,j_1,j_2}^{Y_{t,j_1,j_2}} (1 - \phi_{t,j_1,j_2})^{(1-Y_{t,j_1,j_2})} \tag{2}$$

$$\log \frac{\phi_{t,j_1,j_2}}{1-\phi_{t,j_1,j_2}} = X_{j_1,t} - X_{j_2,t} + FA \qquad (3)$$

In (1), the dynamic of ratings $X_{i,t}$ follows a time series model $AR(k)$. In other words, $X_{i,t}$ is a weighted average of $\bar{X}_{t-1}$ (average rating of all the players at month $t-1$) and $X_{i,t-l}(l = 1, 2, \ldots, k)$ (the player's ratings of previous k months), plus a Gaussian noise. $\bar{X}_{t-1}$ is included in the weighted average, because if a player doesn't play this month, we would like his rating to move to the average rating of all players, instead of just the average rating of the player.

In (2), Y_{t,j_1,j_2} follows a Bernoulli-like distribution, with a parameter ϕ_{t,j_1,j_2} which reflects the 'winning probability' determined by $X_{j_1,t}$ and $X_{j_2,t}$. Here, a multinomial distribution might be an alternative choice, but it also introduces a new tuning parameter which is hard to determine.

In (3), we model the odds of 'winning probability' to be linear in $X_{j_1,t}$ and $X_{j_2,t}$, because of the simplicity of its log-likelihood function. FA is a constant representing the advantage of white player , taking value $\log(\frac{56\%}{44\%})$ since in the rating range of these players, white players' winning probability is about 56% according to FIDE report.

The parameters in the model, the weights w_l's and the noise variance $\sigma_{i,t}^2$, are chosen to have the form

$$w_l = \frac{\exp^{-\lambda l}}{\sum_{s=1}^{k+1} \exp^{-\lambda l}} \quad l = 1, 2, \ldots, k \qquad (4)$$

$$w_0 = 1 - \sum_{l=1}^{k} w_l \qquad (5)$$

$$\sigma_{i,t}^2 = Var(X_{i,t-1}, X_{i,t-2}, \ldots, X_{i,t-k}) + \sigma_0^2 \qquad (6)$$

In(4), the weight is decreasing exponentially, which corresponds to the idea that historical ratings' influence on future games is decreasing exponentially.

In (6), the noise variance includes a term that represents the stability of a player's rating. Note that the tuning parameters of this Hidden Markov Process are k, λ and σ_0^2.

Fitting - Newton Raphson's Method

The idea of fitting the data is to update the player ratings by maximizing the log-likelihood function. At month t, the likelihood function of player ratings $X_{i,t}$ and game results Y_{t,j_1,j_2} is

$$\begin{aligned}
& P(X_{i,t}, Y_{t,j_1,j_2} | X_{i,s}, s < t) \\
\propto \ & \prod_{i=1}^{p} \frac{1}{\sqrt{2\pi}\sigma_{i,t}} e^{-\frac{1}{2\sigma_{i,t}^2}(X_{i,t}-(w_0\bar{X}_{t-1}+\sum_{l=1}^{k} w_l X_{i,t-l}))^2} \\
& \cdot \prod_{(j_1,j_2)} \phi_{t,j_1,j_2}^{Y_{t,j_1,j_2}} (1-\phi_{t,j_1,j_2})^{(1-Y_{t,j_1,j_2})} \\
\propto \ & \prod_{i=1}^{p} \frac{1}{\sqrt{2\pi}\sigma_{i,t}} e^{-\frac{1}{2\sigma_{i,t}^2}(X_{i,t}-(w_0\bar{X}_{t-1}+\sum_{l=1}^{k} w_l X_{i,t-l}))^2} \\
& \cdot \prod_{(j_1,j_2)} e^{Y_{t,j_1,j_2}(X_{j_1,t}-X_{j_2,t}+FA)} \\
& \cdot \frac{1}{1+e^{X_{j_1,t}-X_{j_2,t}+FA}}
\end{aligned} \qquad (7)$$

Take *log*, we get the log-likelihood function

$$\begin{aligned}
l_t &= log(c_t) - \sum_i \frac{1}{2\sigma_{i,t}^2}(X_{i,t} - \\
& \quad (w_0\bar{X}_{t-1} + \sum_{l=1}^{k} w_l X_{i,t-l}))^2 \\
& \quad + \sum_{(j_1,j_2)} Y_{t,j_1,j_2}(X_{j_1,t} - X_{j_2,t} + FA) \\
& \quad - \sum_{(j_1,j_2)} log(1 + e^{X_{j_1,t}-X_{j_2,t}+FA})
\end{aligned} \qquad (8)$$

It's easy to see the log-likelihood function is concave. To maximize it, we compute the gradient and Hessian

$$\begin{aligned}
grad_i &= \frac{\partial l_t}{\partial X_{i,t}} \\
&= -\frac{1}{\sigma_{i,t}^2}(X_{i,t} - (w_0\bar{X}_{t-1} + \sum_{l=1}^{k} w_l X_{i,t-l})) \\
& \quad + \sum_{(i,j_2)} Y_{t,i,j_2} - \sum_{(j_1,i)} Y_{t,j_1,i} \\
& \quad - \sum_{(i,j_2)} \frac{e^{X_{i,t}-X_{j_2,t}+FA}}{1+e^{X_{i,t}-X_{j_2,t}+FA}} \\
& \quad + \sum_{(j_1,i)} \frac{e^{X_{j_1,t}-X_{i,t}+FA}}{1+e^{X_{j_1,t}-X_{i,t}+FA}}
\end{aligned} \qquad (9)$$

$$H_{i,j} = \frac{\partial^2 l_t}{\partial X_{i,t} \partial X_{j,t}}$$
$$= \begin{cases} -\frac{1}{\sigma_{i,t}^2} - \sum_{(i,j_2)} \frac{e^{X_{i,t}-X_{j_2,t}+FA}}{(1+e^{X_{i,t}-X_{j_2,t}+FA})^2} \\ -\sum_{(j_1,i)} \frac{e^{X_{j_1,t}-X_{i,t}+FA}}{(1+e^{X_{j_1,t}-X_{i,t}+FA})^2} & i = j \\ \sum_{(i,j)} \frac{e^{X_{i,t}-X_{j,t}+FA}}{(1+e^{X_{i,t}-X_{j,t}+FA})^2} \\ +\sum_{(j,i)} \frac{e^{X_{j,t}-X_{i,t}+FA}}{(1+e^{X_{j,t}-X_{i,t}+FA})^2} & i \neq j \end{cases} \quad (10)$$

Then we can apply Newton Raphson's Method to get the updated player ratings.

$$X_{i,t}^{new} := X_{i,t}^{old} - H^{-1}\vec{grad} \quad (11)$$

At the start of the updating process, we initialize the ratings to be

$$X_{i,t} \sim N(0, \sigma_0^2) \quad t = -(k-1), -(k-2), \ldots, 0 \quad (12)$$

Prediction

Given the current ratings X_i and X_j of player i and j, we predict the game result $Y_{i,j}$ by computing

$$P(Y_{i,j} = 1 \text{ (white win)}) \propto \phi_{i,j} \quad (13)$$
$$P(Y_{i,j} = 0.5 \text{ (draw)}) \propto C\sqrt{\phi_{i,j}(1-\phi_{i,j})} \quad (14)$$
$$P(Y_{i,j} = 0 \text{ (white lose)}) \propto 1 - \phi_{i,j} \quad (15)$$

where $\phi_{i,j}$ is computed by (3) and constant C is set to be $\pi/8$. Constant C is set such that the draw game probability agrees with the probability that a draw game appears in the training data set, which is roughlt 50%. We then pick the result with largest probability as our prediction.

Picking tuning parameter

In our model, we have 3 tuning parameters, k, λ and σ_0^2 that we have to specify before fitting the data. To pick them, we separate the data into training set and testing set, try different combinations of k, λ and σ_0^2, and pick the combination with the largest log-likelihood for testing data. Figure 1 shows how testing log-likelihood changes with tuning parameters.

With this analysis, we pick $k = 3$, $\lambda = 0.2$, $\sigma_0^2 = 1$.

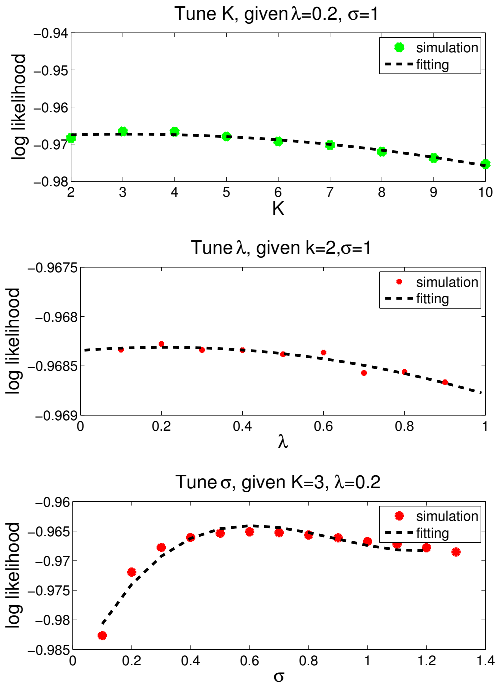

Figure 1: *Testing log-likelihood v.s Tuning parameters*

Results

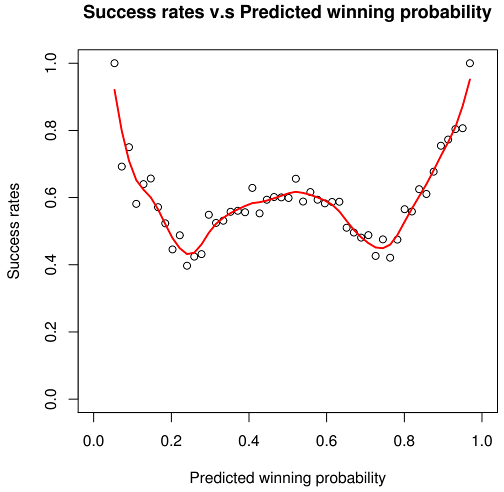

Figure 2: *Success rates v.s 'predict winning probability' ϕ*

We applied our method to a data set of game results of 2000 chess players in 11 years. The data set was separated into 2 parts, game results of month 1 to 127 as the training set and results of month 128 to 132 as testing set. For the training part, we fitted the training set to get the player ratings. Then for testing part, each month we used the most recently updated ratings to predict the game results, then the ratings were updated using the observed game results of this month.

The success rate of prediction is 55.64%, which is much better than a random guess (33%, since result has 'white win', 'draw', 'white lose' 3 cases). Also if we only look at the games with the case that both prediction and observed results are not 'draw', the success rate of prediction is 85.73%, which implies that the prediction is reliable when ratings show one player dominates the other.

To further illustrate the result, we plot the success rates v.s 'predict winning probability' ϕ as Figure 2. We can see that when ϕ is below 0.1 or above 0.9, the success rate is very high, which implies the confidence that when one player dominates the other, our rating system is reliable. When ϕ is between 0.4 and 0.6, the success rate is around 60%, which makes sense because for games between players with similar ratings, most results would be 'draw' which is consistent with our prediction but it's also likely that 'win' or 'lost' happens. For ϕ between 0.2 and 0.4 (0.6 and 0.8), the prediction is poor, because the result is on the edge of 'lose' and 'draw' (or 'win' and 'draw'), which is difficult to predict in nature.

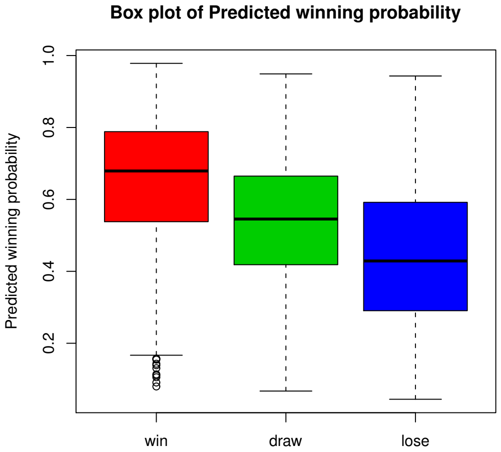

Figure 3: *Box plot of 'predict winning probability'* ϕ

We also plot the box plot of 'predict winning probability' ϕ when true result is 'win', 'draw' and 'lose' as Figure 3. The figure clearly shows that ϕ predicts the trend of true results.

Summary and Future Work

As a summary, the overall performance of the Hidden Markov Process Model in predicting chess game results is satisfactory. Chess itself is a rather unpredictable game, especially if two players are close in rating performance and there are tons of games where lower rated players upset higher rated players. Therefore, a success rate of 55.64% given the three possible game outcomes is not a bad result.

From Figure 2 we can see that when the predicted winning probability ϕ is near the threshold between a draw and win/lose game, the prediction result is poor. In future work, a new model to better predict the game results when player ratings' difference is near the threshold may be possible. In general, it is hard to predict game results in this edge case but a study of a large dataset of this particular type of games may help to establish a new model. Several factors, including the two players' historical game results and trends, game time control, category of the tournament, chess opening preference and playing style, may be incorporated to better evaluate their individual winning probability.

Another further work that can be done is to compare our system with the prevailing Elo chess rating system adopted by FIDE. Accurate comparison needs careful calibration from the Elo system into our system but from an empirical points of view, our result should be no worse than Elo rating system. Much of the appeal of the Elo system comes from its simplicity and familiarity, and it was ideally suited to a time when the computation of ratings was a significant practical challenge even for an annual list of a few hundred players. Elo's formula was derived theoretically, in an era without large amounts of historical data or anything approaching today's computing power. With the benefit of powerful computers and large game databases, we are able to investigate approaches that might do better than Elo at predicting chess results. Such an investigation could have major implications on the theory and practice of ratings methodology, both for chess and also for the world beyond chess.

References

[The USCF Rating System] Glickman, Mark E., and Thomas Doan. (2008).

[Rating the chess rating system] Glickman, Mark E., and Albyn C. Jones. (1999). CHANCE-BERLIN THEN NEW YORK-12 (1999): 21-28.

[Hidden markov processes] Ephraim, Yariv, and Neri Merhav. (2002). Information Theory, IEEE Transactions on 48.6 (2002): 1518-1569.

[FIDE Chess Rating Challenge] http://www.kaggle.com/c/ChessRatings2

[Elo rating system] http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Elo_rating_system.html