Journal of Machine Learning Research 12 (2011) 267-300 Submitted 9/09; Revised 7/10; Published 1/11

A Bayesian Approximation Method for Online Ranking

Ruby C. Weng CHWENG@NCCU.EDU.TW
Department of Statistics
National Chengchi University
Taipei 116, Taiwan

Chih-Jen Lin CJLIN@CSIE.NTU.EDU.TW
Department of Computer Science
National Taiwan University
Taipei 106, Taiwan

Editor: Thore Graepel

Abstract

This paper describes a Bayesian approximation method to obtain online ranking algorithms for games with multiple teams and multiple players. Recently for Internet games large online ranking systems are much needed. We consider game models in which a k-team game is treated as several two-team games. By approximating the expectation of teams' (or players') performances, we derive simple analytic update rules. These update rules, without numerical integrations, are very easy to interpret and implement. Experiments on game data show that the accuracy of our approach is competitive with state of the art systems such as TrueSkill, but the running time as well as the code is much shorter.

Keywords: Bayesian inference, rating system, Bradley-Terry model, Thurstone-Mosteller model, Plackett-Luce model

1. Introduction

Many have proposed online updating algorithms for paired comparison experiments. These online algorithms are especially useful when the number of teams to be ranked and the number of games are very large. For the ranking of many sports, possibly the most prominent ranking system in use today is Elo (1986). The Elo ranking system has been used successfully by leagues organized around two-player games, such as world football league, the US Chess Federation (USCF) or the World Chess Federation (FIDE), and a variety of others. Glickman (1999) proposed the Glicko updating system, which improves over Elo by incorporating the variability in parameter estimates. To the best of our knowledge, Glicko is the first Bayesian ranking system. To begin, prior to a rating period, a player's skill (θ) is assumed to follow a Gaussian distribution which can be characterized by two numbers: the average skill of the player (μ) and the degree of uncertainty in the player's skill (σ). Then, Glicko models the game outcomes by the Bradley-Terry model (Bradley and Terry, 1952) and updates players' skills after a rating period. Glickman (1999) also reported that the Glicko system performs best when the number of games per player is around 5-10 in a rating period. Though the Elo and Glicko ranking systems have been successful, they are designed for two-player games. In video games a game often involves more than two players or teams.

To address this problem, recently Microsoft Research developed TrueSkill (Herbrich et al., 2007), a ranking system for Xbox Live. TrueSkill is also a Bayesian ranking system using a Gaussian belief over a player's skill, but it differs from Glicko in several ways. First, it is designed for multi-team/multi-player games, and it updates skills after each game rather than a rating period. Secondly, Glicko assumes that the performance difference follows the logistic distribution (the model is termed the Bradley-Terry model), while TrueSkill uses the Gaussian distribution (termed the Thurstone-Mosteller model). Moreover, TrueSkill models the draws and offers a way to measure the quality of a game between any set of teams. The way TrueSkill estimates skills is by constructing a graphical model and using approximate message passing. In the easiest case, a two-team game, the TrueSkill update rules are fairly simple. However, for games with multiple teams and multiple players, the update rules are not possible to write down as they require an iterative procedure.

The present paper concerns the ranking of players from outcomes of multiple players or games. We consider a k-team game as a single match and discuss the possibility of obtaining efficient update algorithms. We introduce a Bayesian approximation method to derive simple analytic rules for updating team strength in multi-team games. These update rules avoid a numerical integration and are easy to interpret and implement. Strength of players in a team are then updated by assuming that a team's skill is the sum of skills of ts members. Our framework can be applied by considering various ranking models. In this paper, we demonstrate the use of the Bradley-Terry model, the Thurstone-Mosteller model, and the Plackett-Luce model. Experiments on game data show that the accuracy of our approach is competitive with the TrueSkill ranking system, but the running time as well as the code are shorter. Our method is faster because we employ analytic update rules rather than iterative procedures in TrueSkill.

The organization of this paper is as follows. In Section 2, we briefly review the modeling of ranked data. Section 3 presents our approximation method and gives update equations of using the Bradley-Terry model. Update rules of using other ranking models are given in Section 4. As Glicko is also based on the Bradley-Terry model, for a comparison purpose we describe its approximation procedures in Section 5. Experimental studies are provided in Section 6. Section 7 concludes the paper. Some notation is given in Table 1.

2. Review of Models and Techniques

This section reviews existing methods for modeling ranked data and discusses approximation techniques for Bayesian inference.

2.1 Modeling Ranked Data

Given the game outcome of k teams, we define $r(i)$ as the rank of team i. If teams $i_1, \ldots, i_d$ are tied together, we have

$$r(i_1) = \cdots = r(i_d),$$

and let the team q ranked next have

$$r(q) = r(i_1) + d.$$

Notation	Explanation
k	number of teams participating in a game
n_i	number of players in team i
θ_{ij}	strength of the jth player in team i
$N(\mu_{ij}, \sigma_{ij}^2)$	prior distribution of θ_{ij}
Z_{ij}	standardized quantity of θ_{ij}; see (45)
θ_i	strength of team i; $\theta_i = \sum_{j=1}^{n_i} \theta_{ij}$
β_i^2	uncertainty about the performance of team i
X_i	performance of team i ($X_i \sim N(\theta_i, \beta_i^2)$ for Thurstone-Mosteller model)
$N(\mu_i, \sigma_i^2)$	prior distribution of θ_i
μ_i	$\sum_{j=1}^{n_i} \mu_{ij}$
σ_i^2	$\sum_{j=1}^{n_i} \sigma_{ij}^2$
Z_i	standardized quantity of θ_i; see (24)
$r(i)$	rank of team i in a game; smaller is better; see Section 2.1
$\bar{r}(i)$:	index of the ith ranked team; "inverse" of r; see Section 2.1
ϵ	draw margin (Thurstone-Mosteller model)
ϕ	probability density function of a standard normal distribution; see (66)
Φ	cumulative distribution function of a standard normal distribution
ϕ_k	probability density function of a k-variate standard normal distribution
Φ_k	cumulative distribution function of a k-variate standard normal distribution
κ	a small positive value to avoid σ_i^2 becoming negative; see (28) and (44)
D	the game outcome
$E(\cdot)$	expectation with respect to a random variable

Table 1: Notation

For example, if four teams participate in a game, their ranks may be

$$r(1) = 2, r(2) = 2, r(3) = 4, r(4) = 1, \tag{1}$$

where teams 1 and 2 are both ranked the second. Then team 3, which ranked the next, has $r(3) = 4$. We also need the "inverse" of r, so that $\bar{r}(i)$ indicates the index of the ith ranked team. However, the function r is not one-to-one if ties occur, so the inverse is not directly available. We choose $\bar{r}$ to be any one-to-one mapping from $\{1, \ldots, k\}$ to $\{1, \ldots, k\}$ satisfying

$$r(\bar{r}(i)) \leq r(\bar{r}(i+1)), \forall i. \tag{2}$$

For example, if r is as in Equation (1), then $\bar{r}$ could be

$$\bar{r}(1) = 4, \bar{r}(2) = 1, \bar{r}(3) = 2, \bar{r}(4) = 3.$$

We may have $\bar{r}(2) = 2$ and $\bar{r}(3) = 1$ instead, though in this paper choosing any $\bar{r}$ satisfying (2) is enough.

A detailed account of modeling ranked data is by Marden (1995). For simplicity, in this section we assume that ties do not occur though ties are handled in later sections. Two most commonly used models for ranked data are the Thurstone-Mosteller model (Thurstone, 1927) and the Bradley-Terry model. Suppose that each team is associated with a continuous but unobserved random variable X_i, representing the actual performance. The observed ordering that team $\bar{r}(1)$ comes in first, team $\bar{r}(2)$ comes in second and so on is then determined by the X_i's:

$$X_{\bar{r}(1)} > X_{\bar{r}(2)} > \cdots > X_{\bar{r}(k)}. \tag{3}$$

Thurstone (1927) invented (3) and proposed using the normal distribution. The resulting likelihood associated with (3) is

$$P(X_{\bar{r}(1)} - X_{\bar{r}(2)} > 0, \ldots, X_{\bar{r}(k-1)} - X_{\bar{r}(k)} > 0), \tag{4}$$

where $X_{\bar{r}(i)} - X_{\bar{r}(i+1)}$ follows a normal distribution. In particular, if $k = 2$ and X_i follows $N(\theta_i, \beta_i^2)$, where θ_i is the strength of team i and β_i^2 is the uncertainty of the actual performance X_i, then

$$P(X_i > X_q) = \Phi\left(\frac{\theta_i - \theta_q}{\sqrt{\beta_i^2 + \beta_q^2}}\right), \tag{5}$$

where Φ denotes the cumulative distribution function of a standard normal density.

Numerous papers have addressed the ranking problem using models like (5). However, most of them consider an off-line setting. That is, they obtain the likelihood using all available data and maximize the likelihood. Such an approach is suitable if data are not large. Recent attempts to extend this off-line approach to multiple players and multiple teams include Huang et al. (2006). However, for large systems which constantly have results being added/dropped, an online approach is more appropriate.

The Elo system is an online rating scheme which models the probability of game output as (5) with $\beta_i = \beta_q$ and, after each game, updates the strength θ_i by

$$\theta_i \leftarrow \theta_i + K(s - P(i \text{ wins})), \tag{6}$$

where K is some constant, and $s = 1$ if i wins and 0 otherwise. This formula is a very intuitive way to update strength after a game. More discussions of (6) can be seen in, for example, Glickman (1999). The Elo system with the logistic variant corresponds to the Bradley-Terry model (Bradley and Terry, 1952). The Bradley-Terry model for paired comparisons has the form

$$P(X_i > X_q) = \frac{v_i}{v_i + v_q}, \tag{7}$$

where $v_i > 0$ is the strength of team i. The model (7) dates back to Zermelo (1929) and can be derived in several ways. For instance, it can be obtained from (3) by letting X_i follow a Gumbel distribution with the cumulative distribution function

$$P(X_i \le x) = \exp(-\exp(-(x - \theta_i))), \text{ where } \theta_i = \log v_i.$$

Then $X_i - X_q$ follows a logistic distribution with the cumulative distribution function

$$P(X_i - X_q \le x) = \frac{e^{\theta_q}}{e^{\theta_i - x} + e^{\theta_q}}. \tag{8}$$

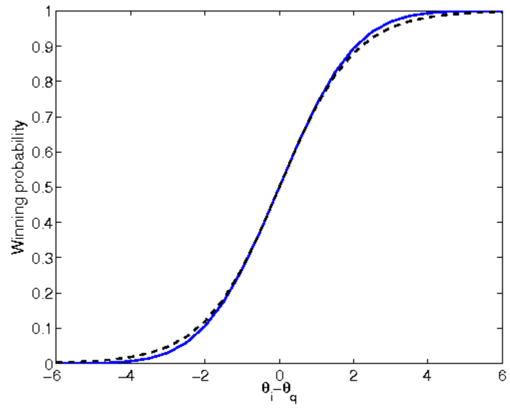

Figure 1: Winning probability $P(X_i > X_q)$. Solid (blue): Gaussian distribution (5), Dashed (black): logistic distribution (8).

Using $x = 0$ and $P(X_i > X_q) = 1 - P(X_i \leq X_q)$, we obtain (7). In fact, most currently used Elo variants for chess data use a logistic distribution rather than Gaussian because it is argued that weaker players have significantly greater winning chances than the Gaussian model predicts.[1] Figure 1 shows i's winning probability $P(X_i > X_q)$ against the skill difference $\theta_i - \theta_q$ for the two models (5) and (8). The $(\beta_i^2 + \beta_q^2)^{1/2}$ in (5) are set as $4/\sqrt{2\pi} \approx 1.6$ so that the two winning probability curves have the same slope at $\theta_i = \theta_q$. Clearly, given that the two models closely match when two teams have about the same skill levels, the logistic model gives a weak team i a higher winning chance than the Gaussian model does.

In addition to Elo and Glicko, other online systems have been proposed. For example, Menke and Martinez (2008) propose using Artificial Neural Networks. Though this approach can handle multiple players per team, it aims to handle only two teams per game.

For comparisons involving $k \geq 3$ teams per game, the Bradley-Terry model has been generalized in various ways. The Plackett-Luce model (Marden, 1995) is one of such models. This model, motivated by a k-horse race, has the form

$$P(\bar{r}(1), \ldots, \bar{r}(k)) = \frac{e^{\theta_{\bar{r}_1}}}{e^{\theta_{\bar{r}_1}} + \cdots + e^{\theta_{\bar{r}_k}}} \times \frac{e^{\theta_{\bar{r}_2}}}{e^{\theta_{\bar{r}_2}} + \cdots + e^{\theta_{\bar{r}_k}}} \times \cdots \times \frac{e^{\theta_{\bar{r}_k}}}{e^{\theta_{\bar{r}_k}}}. \tag{9}$$

An intuitive explanation of this model is a multistage ranking in which one first chooses the most favorite, then chooses the second favorite out of the remaining, etc.

When $k \geq 3$, as the $X_{\bar{r}(i)} - X_{\bar{r}(i+1)}$'s in (4) are dependent, the calculation of the joint probability (4) involves a $(k-1)$-dimensional integration, which may be difficult to calculate. Therefore, TrueSkill uses a factor graph and the approximate message passing (Kschischang et al., 2001) to infer the marginal belief distribution over the skill of each team. In fact, some messages in the factor graph are non Gaussian and these messages are approximated via moment matching, using the Expectation Propagation algorithm (Minka, 2001).

1. According to `http://en.wikipedia.org/wiki/Elo_rating_system`, USCF and FIDE use formulas based on the logistic distribution.

2.2 Approximation Techniques for Bayesian Inference

From a Bayesian perspective, both the observed data and the model parameters are considered random quantities. Let D denote the observed data, and θ the unknown quantities of interest. The joint distribution of D and θ is determined by the prior distribution $P(\theta)$ and the likelihood $P(D|\theta)$:

$$P(D, \theta) = P(D|\theta)P(\theta).$$

After observing D, Bayes theorem gives the distribution of θ conditional on D:

$$P(\theta|D) = \frac{P(\theta, D)}{P(D)} = \frac{P(\theta, D)}{\int P(\theta, D)d\theta}.$$

This is the *posterior distribution* of θ, which is useful for estimation. Quantities about the posterior distribution such as moments, untiles, etc can be expressed in terms of posterior expectations of some functions $g(\theta)$; that is,

$$E[g(\theta)|D] = \frac{\int g(\theta)P(\theta, D)d\theta}{\int P(\theta, D)d\theta}. \tag{10}$$

The probability $P(D)$, called *evidence* or *marginal likelihood* of the data, is useful for model selection. Both $P(\theta|D)$ and $P(D)$ are major objects of Bayesian inference.

The integrations involved in Bayesian inference are usually intractable. The approximation techniques can be divided into deterministic and nondeterministic methods. The nondeterministic method refers to the Monte Carlo integration such as Markov Chain Monte Carlo (MCMC) methods, which draw samples approximately from the desired distribution and forms sample averages to estimate the expectation. However, when it comes to sequential updating with new data, the MCMC methods may not be computationally feasible, the reason being that it does not make use of the analysis from the previous data; see, for example, Section 2.8 in Glickman (1993).

The popular deterministic approaches include Laplace method, variational Bayes, expectation propagation, among others. The Laplace method is a technique for approximating integrals:

$$\int e^{nf(\boldsymbol{x})}d\boldsymbol{x} \approx \left(\frac{2\pi}{n}\right)^{\frac{k}{2}} |-\nabla^2 f(\boldsymbol{x}_0)|^{-\frac{1}{2}} e^{nf(\boldsymbol{x}_0)},$$

where $\boldsymbol{x}$ is k-dimensional, n is a large number, $f : R^k \to R$ is twice differentiable with a unique global maximum at $\boldsymbol{x}_0$, and $|\cdot|$ is the determinant of a matrix. By writing $P(\theta, D) = \exp(\log P(\theta, D))$, one can approximate the integral $\int P(\theta, D)d\theta$. This method has been applied in Bayesian statistics; for example, see Tierney and Kadane (1986) and Kass and Raftery (1995).

The variational Bayes methods are a family of techniques for approximating these intractable integrals. They construct a lower bound on the marginal likelihood and then try to optimize this bound. They also provide an approximation to the posterior distribution which is useful for estimation.

The Expectation Propagation algorithm (Minka, 2001) is an iterative approach to approximate posterior distributions. It tries to minimize Kullback-Leibler divergence between the true posterior and the approximated distribution. It can be viewed as an extension of assumed-density filtering to batch situation. The TrueSkill system (Herbrich et al., 2007) is based on this algorithm.

Now we review an identity for integrals in Lemma 1 below, which forms the basis of our approximation method. Some definitions are needed. A function $f : R^k \to R$ is called almost differentiable if there exists a function $\nabla f : R^k \to R^k$ such that

$$f(\boldsymbol{z}+\boldsymbol{y}) - f(\boldsymbol{z}) = \int_0^1 \boldsymbol{y}^T \nabla f(\boldsymbol{z}+t\boldsymbol{y})dt \tag{11}$$

for $\boldsymbol{z}, \boldsymbol{y} \in R^k$. Of course, a continuously differentiable function f is almost differentiable with ∇f equal to the gradient, and (11) is the indefinite integral in multi-dimensional case.

Given $h : R^k \to R$, let $h_0 = \int h(\boldsymbol{z})d\Phi_k(\boldsymbol{z})$ be a constant, $h_k(\boldsymbol{z}) = h(\boldsymbol{z})$,

$$h_j(z_1,\ldots,z_j) = \int_{R^{k-j}} h(z_1,\ldots,z_j,\boldsymbol{w})d\Phi_{k-j}(\boldsymbol{w}), \text{ and} \tag{12}$$

$$g_j(z_1,\ldots,z_k) = e^{z_j^2/2}\int_{z_j}^{\infty} [h_j(z_1,\ldots,z_{j-1},w) - h_{j-1}(z_1,\ldots,z_{j-1})]e^{-w^2/2}dw, \tag{13}$$

for $-\infty < z_1,\ldots,z_k < \infty$ and $j = 1,\ldots,k$. Then let

$$Uh = [g_1,\ldots,g_k]^T \qquad \text{and} \qquad Vh = \frac{U^2h + (U^2h)^T}{2}, \tag{14}$$

where U^2h is the $k \times k$ matrix whose jth column is Ug_j and g_j is as in (13).

Let Γ be a measure of the form:

$$d\Gamma(\boldsymbol{z}) = f(\boldsymbol{z})\phi_k(\boldsymbol{z})d\boldsymbol{z}, \tag{15}$$

where f is a real-valued function (not necessarily non-negative) defined on R^k.

Lemma 1 *(W-Stein's Identity) Suppose that $d\Gamma$ is defined as in (15), where f is almost differentiable. Let h be a real-valued function defined on R^k. Then,*

$$\int h(\boldsymbol{z})d\Gamma(\boldsymbol{z}) = \int f(\boldsymbol{z})d\Phi_k(\boldsymbol{z}) \cdot \int h(\boldsymbol{z})d\Phi_k(\boldsymbol{z}) + \int (Uh(\boldsymbol{z}))^T \nabla f(\boldsymbol{z})d\Phi_k(\boldsymbol{z}), \tag{16}$$

provided all the integrals are finite.

Lemma 1 was given by Woodroofe (1989). The idea of this identity originated from Stein's lemma (Stein, 1981), but the latter considers the expectation with respect to a normal distribution (i.e., the integral $\int h(\boldsymbol{z})d\Phi_k(\boldsymbol{z})$), while the former studies the integration with respect to a "nearly normal distribution" Γ in the sense of (15). Stein's lemma is famous and of interest because of its applications to James-Stein estimator (James and Stein, 1961) and empirical Bayes methods.

The proof of this lemma is in Proposition 1 of Woodroofe (1989). For self-completeness, we sketch it for the 1-dimensional case in Appendix A. Essentially the proof is based on exchanging the order of integration (Fibini theorem), and it is the very idea for proving Stein's lemma. Due to this reason, Woodroofe termed (16) a version of Stein's identity. However, to distinguish it from Stein's lemma, here we refer to it as W-Stein's identity.

Now we assume that $\partial f/\partial z_j,\ j=1,\dots,k$ are almost differentiable. Then, by writing

$$(Uh(\boldsymbol{z}))^T\nabla f(\boldsymbol{z})=\sum_{i=1}^k g_i(\boldsymbol{z})\frac{\partial f(\boldsymbol{z})}{\partial z_i}$$

and applying (16) with h and f replacing by g_i and $\partial f/\partial z_i$, we obtain

$$\int g_i\frac{\partial f}{\partial z_i}d\Phi_k(\boldsymbol{z}) \;=\; \Phi_k(g_i)\int\frac{\partial f}{\partial z_i}d\Phi_k(\boldsymbol{z})+\int(U(g_i))^T\nabla\left(\frac{\partial f}{\partial z_i}\right)d\Phi_k(\boldsymbol{z}), \tag{17}$$

provided all the integrals are finite. Note that $\Phi_k(g_i)$ in the above equation is a constant defined as

$$\Phi_k(g_i)=\int g_i(\boldsymbol{z})\phi_k(\boldsymbol{z})d\boldsymbol{z}.$$

By summing up both sides of (17) over $i=1,\dots,k$, we can rewrite (16) as

$$\begin{aligned}\int h(\boldsymbol{z})f(\boldsymbol{z})d\Phi_k(\boldsymbol{z}) \;&=\; \int f(\boldsymbol{z})d\Phi_k(\boldsymbol{z})\cdot\int h(\boldsymbol{z})d\Phi_k(\boldsymbol{z})+(\Phi_kUh)^T\int\nabla f(\boldsymbol{z})d\Phi_k(\boldsymbol{z})\\ &\quad+\int\operatorname{tr}\left[(Vh(\boldsymbol{z}))\nabla^2 f(\boldsymbol{z})\right]d\Phi_k(\boldsymbol{z}); \end{aligned}\tag{18}$$

see Proposition 2 of Woodroofe and Coad (1997) and Lemma 1 of Weng and Woodroofe (2000). Here $\Phi_kUh=(\Phi_k(g_1),...,\Phi_k(g_k))^T$, “tr” denotes the trace of a matrix, and $\nabla^2 f$ the Hessian matrix of f. An extension of this lemma is in Weng (2010).

Let $\boldsymbol{Z}=[Z_1,\dots,Z_k]^T$ be a k-dimensional random vector with the probability density

$$C\phi_k(\boldsymbol{z})f(\boldsymbol{z}), \tag{19}$$

where

$$C=\left(\int\phi_k(\boldsymbol{z})f(\boldsymbol{z})d\boldsymbol{z}\right)^{-1}$$

is the normalizing constant. Lemma 1 can be applied to obtain expectations of functions of $\boldsymbol{Z}$ in the following corollary.

Corollary 2 *Suppose that $\boldsymbol{Z}$ has probability density (19). Then,*

$$\int fd\Phi_k=C^{-1}\ \textit{and}\ Eh(\boldsymbol{Z})=\int h(\boldsymbol{z})d\Phi_k(\boldsymbol{z})+E\left[(Uh(\boldsymbol{Z}))^T\frac{\nabla f(\boldsymbol{Z})}{f(\boldsymbol{Z})}\right]. \tag{20}$$

Further, (18) *and* (20) *imply*

$$Eh(\boldsymbol{Z})=\int h(\boldsymbol{z})d\Phi_k(\boldsymbol{z})+(\Phi_kUh)^TE\left[\frac{\nabla f(\boldsymbol{Z})}{f(\boldsymbol{Z})}\right]+E\left[\operatorname{tr}\left(Vh(\boldsymbol{Z})\frac{\nabla^2 f(\boldsymbol{Z})}{f(\boldsymbol{Z})}\right)\right]. \tag{21}$$

In particular, if $h(\boldsymbol{z})=z_i$, then by (14) it follows $Uh(\boldsymbol{z})=\boldsymbol{e}_i$ (a function from R^k to R^k); and if $h(\boldsymbol{z})=z_iz_j$ and $i<j$, then $Uh(\boldsymbol{z})=z_i\boldsymbol{e}_j$, where $\{\boldsymbol{e}_1,\cdots,\boldsymbol{e}_k\}$ denote the standard basis for R^k. For example, if $k=3$ and $h(\boldsymbol{z})=z_1z_2$, then $Uh(\boldsymbol{z})=[0,z_1,0]^T$ and $U^2h(\boldsymbol{z})$ is

the matrix whose $(1,2)$ entry is 1 and the rest entries are zeros; see Appendix B for details. With these special h functions, (20) and (21) become

$$E[\boldsymbol{Z}] = E\left[\frac{\nabla f(\boldsymbol{Z})}{f(\boldsymbol{Z})}\right], \tag{22}$$

$$E[Z_i Z_q] = \delta_{iq} + E\left[\frac{\nabla^2 f(\boldsymbol{Z})}{f(\boldsymbol{Z})}\right]_{iq}, \quad i, q = 1, \ldots, k, \tag{23}$$

where $\delta_{iq} = 1$ if $i = q$ and 0 otherwise, and $[\cdot]_{iq}$ indicates the (i, q) component of a matrix.

In the current context of online ranking, since the skill θ is assumed to follow a Gaussian distribution, the update procedure is mainly for the mean and the variance. Therefore, (22) and (23) will be useful. The detailed approximation procedure is in the next section.

3. Method

In this section, we first present our proposed method for updating team and individual skills. Then, we give the detailed derivation for the Bradley-Terry model.

3.1 Approximating the Expectations

Let θ_i be the strength parameter of team i whose ability is to be estimated. Bayesian online rating systems such as Glicko and TrueSkill start by assuming that θ_i has a prior distribution $N(\mu_i, \sigma_i^2)$ with μ_i and σ_i^2 known, next model the game outcome by some probability models, and then update the skill (by either analytic or numerical approximations of the posterior mean and variance of θ_i) at the end of the game. These revised mean and variance are considered as prior information for the next game, and the updating procedure is repeated.

Equations (22) and (23) can be applied to online skill updates. To start, suppose that team i has a strength parameter θ_i and assume that the prior distribution of θ_i is $N(\mu_i, \sigma_i^2)$. Upon the completion of a game, their skills are characterized by the posterior mean and variance of $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_k]^T$. Let D denote the result of a game and $\boldsymbol{Z} = [Z_1, \ldots, Z_k]^T$ with

$$Z_i = \frac{\theta_i - \mu_i}{\sigma_i}, i = 1, \ldots, k, \tag{24}$$

where k is the number of teams. The posterior density of $\boldsymbol{Z}$ given the game outcome D is

$$P(\boldsymbol{z}|D) = C\phi_k(\boldsymbol{z})f(\boldsymbol{z}),$$

where $f(\boldsymbol{z})$ is the probability of game outcome $P(D|\boldsymbol{z})$. Thus, $P(\boldsymbol{z}|D)$ is of the form (19). Subsequently we omit D in all derivations.

Next, we shall update the skill as the posterior mean and variance of $\boldsymbol{\theta}$. Equations (22), (23) and the relation between Z_i and θ_i in (24) give that

$$\begin{aligned} \mu_i^{\text{new}} &= E[\theta_i] = \mu_i + \sigma_i E[Z_i] \\ &= \mu_i + \sigma_i E\left[\frac{\partial f(\boldsymbol{Z})/\partial Z_i}{f(\boldsymbol{Z})}\right] \end{aligned} \tag{25}$$

and

$$
\begin{aligned}
(\sigma_i^{\text{new}})^2 =& \text{Var}[\theta_i] = \sigma_i^2 \text{Var}[Z_i] \\
=& \sigma_i^2 \left(E[Z_i^2] - E[Z_i]^2 \right) \\
=& \sigma_i^2 \left(1 + E\left[\frac{\nabla^2 f(\boldsymbol{Z})}{f(\boldsymbol{Z})} \right]_{ii} - E\left[\frac{\partial f(\boldsymbol{Z})/\partial Z_i}{f(\boldsymbol{Z})} \right]^2 \right).
\end{aligned} \tag{26}
$$

The relation between the current and the new skills are explained below. By chain rule and the definition of Z_i in (24), the second term on the right side of (25) can be written as

$$
\sigma_i E\left[\frac{\partial f(\boldsymbol{Z})/\partial Z_i}{f(\boldsymbol{Z})} \right] = E\left[\frac{\partial f(\boldsymbol{Z})/\partial \theta_i}{f(\boldsymbol{Z})} \right] = E\left[\frac{\partial \log f(\boldsymbol{Z})}{\partial \theta_i} \right],
$$

which is the average of the relative rate of change of f (the probability of game outcome) with respect to strength θ_i. For instance, suppose that team 1 beats team 2. Then, the larger θ_1 is, the more likely we have such an outcome. Hence, f is increasing in θ_1, and the adjustment to team 1's skill is the average of the relative rate of change of team 1's winning probability with respect to its strength θ_1. On the other hand, a larger θ_2 is less likely to result in this outcome; hence, f is decreasing in θ_2 and the adjustment to team 2's skill will be negative. Similarly, we can write the last two terms on the right side of (26) as

$$
\sigma_i^2 \left(E\left[\frac{\nabla^2 f(\boldsymbol{Z})}{f(\boldsymbol{Z})} \right]_{ii} - E\left[\frac{\partial f(\boldsymbol{Z})/\partial Z_i}{f(\boldsymbol{Z})} \right]^2 \right) = E\left[\frac{\partial^2 \log f(\boldsymbol{Z})}{\partial \theta_i^2} \right],
$$

which is the average of the rate of change of $\partial(\log f)/\partial\theta_i$ with respect to θ_i.

We propose approximating expectations in (25) and (26) to obtain the update rules:

$$
\mu_i \leftarrow \mu_i + \Omega_i, \tag{27}
$$

$$
\sigma_i^2 \leftarrow \sigma_i^2 \max(1 - \Delta_i, \kappa), \tag{28}
$$

where

$$
\Omega_i = \sigma_i \left. \frac{\partial f(\boldsymbol{z})/\partial z_i}{f(\boldsymbol{z})} \right|_{\boldsymbol{z}=\mathbf{0}} \tag{29}
$$

and

$$
\begin{aligned}
\Delta_i =& - \left. \frac{\partial^2 f(\boldsymbol{z})/\partial^2 z_i}{f(\boldsymbol{z})} \right|_{\boldsymbol{z}=\mathbf{0}} + \left(\left. \frac{\partial f(\boldsymbol{z})/\partial z_i}{f(\boldsymbol{z})} \right|_{\boldsymbol{z}=\mathbf{0}} \right)^2 \\
=& - \left. \frac{\partial}{\partial z_i} \left(\frac{\partial f(\boldsymbol{z})/\partial z_i}{f(\boldsymbol{z})} \right) \right|_{\boldsymbol{z}=\mathbf{0}}.
\end{aligned} \tag{30}
$$

We set $\boldsymbol{z} = \mathbf{0}$ so that $\boldsymbol{\theta}$ is replaced by $\boldsymbol{\mu}$. Such a substitution is reasonable as we expect that the posterior density of $\boldsymbol{\theta}$ to be concentrated on $\boldsymbol{\mu}$. Then the right-hand sides of (27)-(28) are functions of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, so we can use the current values to obtain new estimates. Due to the approximation (30), $1 - \Delta_i$ may be negative. Hence in (28) we set a small positive lower bound κ to avoid a negative σ_i^2. Further, we find that the prediction results may be affected by how fast the variance σ_i^2 is reduced in (28). More discussion on this issue is in Section 3.5.

3.2 Error Analysis of the Approximation

This section discusses the error induced by evaluating the expectations in (25) and (26) at a single $\boldsymbol{z} = 0$, and then suggests a correction by including the prior uncertainty of skill in the variance of the actual performance. For simplicity, below we only consider a two-team game using the Thurstone-Mosteller model. Another reason of using the Thurstone-Mosteller model is that we can exactly calculate the posterior probability. To begin, suppose that the variance of ith team's actual performance is β_i^2. Then, for the Thurstone-Mosteller model, the joint posterior density of (θ_1, θ_2) is proportional to

$$\phi\left(\frac{\theta_1-\mu_1}{\sigma_1}\right)\phi\left(\frac{\theta_2-\mu_2}{\sigma_2}\right)\Phi\left(\frac{\theta_1-\theta_2}{\sqrt{\beta_1^2+\beta_2^2}}\right),$$

and the marginal posterior density of θ_1 is proportional to

$$\begin{aligned}
&\int_{-\infty}^{\infty}\phi\left(\frac{\theta_1-\mu_1}{\sigma_1}\right)\phi\left(\frac{\theta_2-\mu_2}{\sigma_2}\right)\Phi\left(\frac{\theta_1-\theta_2}{\sqrt{\beta_1^2+\beta_2^2}}\right)d\theta_2 \\
= \quad &\phi\left(\frac{\theta_1-\mu_1}{\sigma_1}\right)\int_{-\infty}^{\infty}\phi\left(\frac{\theta_2-\mu_2}{\sigma_2}\right)\int_{-\infty}^{\theta_1}\frac{1}{\sqrt{2\pi}(\sqrt{\beta_1^2+\beta_2^2})}e^{-\frac{(y-\theta_2)^2}{2(\beta_1^2+\beta_2^2)}}dyd\theta_2 \\
= \quad &\sigma_2\phi\left(\frac{\theta_1-\mu_1}{\sigma_1}\right)\Phi\left(\frac{\theta_1-\mu_2}{\sqrt{\beta_1^2+\beta_2^2+\sigma_2^2}}\right), \qquad (31)
\end{aligned}$$

where the last two equalities are obtained by writing the function $\Phi(\cdot)$ as an integral of ϕ (see (66)) and then interchanging the orders of the double integral. From (31), the posterior mean of θ_1 given D is

$$E(\theta_1) \quad = \quad \frac{\int_{-\infty}^{\infty}\theta_1\phi(\frac{\theta_1-\mu_1}{\sigma_1})\Phi(\frac{\theta_1-\mu_2}{\sqrt{\beta_1^2+\beta_2^2+\sigma_2^2}})d\theta_1}{\int_{-\infty}^{\infty}\phi(\frac{\theta_1-\mu_1}{\sigma_1})\Phi(\frac{\theta_1-\mu_2}{\sqrt{\beta_1^2+\beta_2^2+\sigma_2^2}})d\theta_1}. \qquad (32)$$

Again, by writing the function $\Phi(\cdot)$ as an integral and interchanging the orders of the integrals, we obtain that the numerator and the denominator of the right side of (32) are respectively

$$\Phi\left(\frac{\mu_1-\mu_2}{\sqrt{\sum_{i=1}^2(\beta_i^2+\sigma_i^2)}}\right)\left(\mu_1+\frac{\sigma_1^2}{\sqrt{\sum_{i=1}^2(\beta_i^2+\sigma_i^2)}}\frac{\phi(\frac{\mu_1-\mu_2}{\sqrt{\sum_{i=1}^2(\beta_i^2+\sigma_i^2)}})}{\Phi(\frac{\mu_1-\mu_2}{\sqrt{\sum_{i=1}^2(\beta_i^2+\sigma_i^2)}})}\right)$$

and

$$\Phi\left(\frac{\mu_1-\mu_2}{\sqrt{\sum_{i=1}^2(\beta_i^2+\sigma_i^2)}}\right).$$

Therefore, the exact posterior mean of θ_1 is

$$E(\theta_1) = \mu_1 + \frac{\sigma_1^2}{\sqrt{\sum_{i=1}^2(\beta_i^2+\sigma_i^2)}}\frac{\phi\left(\frac{\mu_1-\mu_2}{\sqrt{\sum_{i=1}^2(\beta_i^2+\sigma_i^2)}}\right)}{\Phi\left(\frac{\mu_1-\mu_2}{\sqrt{\sum_{i=1}^2(\beta_i^2+\sigma_i^2)}}\right)}. \qquad (33)$$

Now we check our estimation. According to (25), (27), and (29),

$$E(\theta) = \mu_1 + \sigma_1 E\left[\frac{\partial f(\boldsymbol{Z})/\partial Z_1}{f(\boldsymbol{Z})}\right] \tag{34}$$

$$\approx \mu_1 + \sigma_1 \left.\frac{\partial f(\boldsymbol{z})/\partial z_1}{f(\boldsymbol{z})}\right|_{\boldsymbol{z}=\boldsymbol{0}}, \tag{35}$$

where

$$f(\boldsymbol{z}) = \Phi\left(\frac{\theta_1-\theta_2}{\sqrt{\beta_1^2+\beta_2^2}}\right) \text{ and } z_i = \frac{\theta_i-\mu_i}{\sigma_i}, i=1,2.$$

The derivation later in (93) shows that (35) leads to the following estimation for $E(\theta_1)$:

$$\mu_1 + \frac{\sigma_1^2}{\sqrt{\beta_1^2+\beta_2^2}} \frac{\phi\left(\frac{\mu_1-\mu_2}{\sqrt{\beta_1^2+\beta_2^2}}\right)}{\Phi\left(\frac{\mu_1-\mu_2}{\sqrt{\beta_1^2+\beta_2^2}}\right)}. \tag{36}$$

The only difference between (33) and (36) is that the former uses $\beta_1^2+\beta_2^2+\sigma_1^2+\sigma_2^2$, while the latter has $\beta_1^2+\beta_2^2$. Therefore, the approximation from (34) to (35) causes certain bias. We can correct the error by substituting β_i^2 with $\beta_i^2+\sigma_i^2$ when using our approximation method. In practice, we use $\beta_i^2 = \beta^2+\sigma_i^2$, where β^2 is a constant.

The above arguments also apply to the Bradley-Terry model. We leave the details in Appendix C.

3.3 Modeling Game Outcomes by Factorization

To derive update rules using (27)-(30), we must define $f(\boldsymbol{z})$ and then calculate Ω_i, Δ_i. Suppose that there are k teams in a game. We shall consider models for which the $f(\boldsymbol{z})$ in (19) can be factorized as

$$f(\boldsymbol{z}) = \prod_{q=1}^{m} f_q(\boldsymbol{z}) \tag{37}$$

for some $m > 0$. If $f_q(\boldsymbol{z})$ involves only several elements of $\boldsymbol{z}$, the above factorization may lead to an easier gradient and Hessian calculation in (22) and (23). The expectation on the right side of (22) involves the following calculation:

$$\begin{aligned}\frac{\partial f/\partial z_i}{f} &= \frac{\partial \log \prod_{q=1}^{m} f_q(\boldsymbol{z})}{\partial z_i} = \sum_{q=1}^{m} \frac{\partial \log f_q(\boldsymbol{z})}{\partial z_i} \\ &= \sum_{q=1}^{m} \frac{\partial f_q/\partial z_i}{f_q}.\end{aligned} \tag{38}$$

Then all we need is to ensure that calculating $\frac{\partial f_q/\partial z_i}{f_q}$ is feasible.

Clearly the Plackett-Luce model (9) has the form of (37). However, the Thurstone's model (3) with the Gaussian distribution can hardly be factorized into the form (37). The main reason is that the probability (4) of a game outcome involves a $(k-1)$-dimensional integration, which is intractable. One may address this problem by modeling a k-team game outcome as $(k-1)$ two-team games (between all teams on neighboring ranks); that is,

$$f(\boldsymbol{z}) = \prod_{i=1}^{k-1} P(\text{outcome between teams ranked } i\text{th and } (i+1)\text{st}). \tag{39}$$

Alternatively, we may consider the game result of k teams as $k(k-1)/2$ two-team games. Then

$$f(\boldsymbol{z}) = \prod_{i=1}^{k} \prod_{q=i+1}^{k} P(\text{outcome between team } i \text{ and team } q). \tag{40}$$

Both (39) and (40) are of the form (37). In Section 3.5, we shall demonstrate the calculation to obtain update rules. Subsequently we refer to (39) as the *partial-pair* approach, while (40) as the *full-pair* approach.

3.4 Individual Skill Update

Now, we consider the case where there are multiple players in each team. Suppose that the ith team has n_i players, the jth player in the ith team has strength θ_{ij}, and the prior distribution of θ_{ij} is $N(\mu_{ij}, \sigma_{ij}^2)$. Let θ_i denote the strength of the ith team. As in Huang et al. (2006) and Herbrich et al. (2007), we assume that a team's skill is the sum of its members' skills. Thus,

$$\theta_i = \sum_{j=1}^{n_i} \theta_{ij} \text{ for } i = 1, \ldots, k, \tag{41}$$

and the prior distribution of θ_i is

$$\theta_i \sim N(\mu_i, \sigma_i^2), \text{ where } \mu_i = \sum_{j=1}^{n_i} \mu_{ij} \text{ and } \sigma_i^2 = \sum_{j=1}^{n_i} \sigma_{ij}^2. \tag{42}$$

Similar to (27)-(28), we propose updating the skill of the jth player in team i by

$$\mu_{ij} \leftarrow \mu_{ij} + \frac{\sigma_{ij}^2}{\sigma_i^2} \Omega_i, \tag{43}$$

$$\sigma_{ij}^2 \leftarrow \sigma_{ij}^2 \max\left(1 - \frac{\sigma_{ij}^2}{\sigma_i^2} \Delta_i, \kappa\right), \tag{44}$$

where Ω_i and Δ_i are defined in (29) and (30), respectively and κ is a small positive value to ensure a positive σ_{ij}^2. Equations (43) and (44) say that Ω_i, the mean skill change of team i, is partitioned to n_i parts with the magnitude proportional to σ_{ij}^2. These rules can be obtained from the following derivation. Let Z_{ij} be the normalized quantity of the random variable θ_{ij}; that is,

$$Z_{ij} = (\theta_{ij} - \mu_{ij})/\sigma_{ij}. \tag{45}$$

As in (27), we could update μ_{ij} by

$$\mu_{ij} \leftarrow \mu_{ij} + \sigma_{ij} \left. \frac{\partial \bar{f}(\bar{z})/\partial z_{ij}}{\bar{f}} \right|_{\bar{z}=\mathbf{0}}, \tag{46}$$

where $\bar{f}(\bar{\boldsymbol{z}})$ is the probability of game outcomes and

$$\bar{\boldsymbol{z}} = [z_{11}, \ldots, z_{1n_1}, \ldots, z_{k1}, \ldots, z_{kn_k}]^T.$$

Since we assume a team's strength is the sum of its members', from (24), (41), (42), and (45) we have

$$Z_i = \frac{\theta_i - \mu_i}{\sigma_i} = \frac{\sum_j \sigma_{ij} Z_{ij}}{\sigma_i}; \tag{47}$$

hence, it is easily seen that $\bar{f}(\bar{\boldsymbol{z}})$ is simply a reparametrization of $f(\boldsymbol{z})$ (defined in Section 3.1):

$$f(\boldsymbol{z}) = f\left(\sum_{j=1}^{n_1} \frac{\sigma_{1j} z_{1j}}{\sigma_1}, \ldots, \sum_{j=1}^{n_k} \frac{\sigma_{kj} z_{kj}}{\sigma_k}\right) = \bar{f}(\bar{\boldsymbol{z}})$$

With (47),

$$\frac{\partial \bar{f}(\bar{\boldsymbol{z}})}{\partial z_{ij}} = \frac{\partial f(\boldsymbol{z})}{\partial z_i} \cdot \frac{\partial z_i}{\partial z_{ij}} = \frac{\sigma_{ij}}{\sigma_i} \frac{\partial f(\boldsymbol{z})}{\partial z_i}$$

and (46) becomes

$$\mu_{ij} \leftarrow \mu_{ij} + \frac{\sigma_{ij}^2}{\sigma_i^2} \cdot \sigma_i \left. \frac{\partial f(\boldsymbol{z})/\partial z_i}{f} \right|_{\boldsymbol{z}=\mathbf{0}}.$$

Following the definition of Ω_i in (29) we obtain the update rule (43), which says that within team i the adjustment to μ_{ij} is proportional to σ_{ij}^2. The update rule (44) for the individual variance can be derived similarly.

3.5 Example: Bradley-Terry Model (Full-pair)

In this section, we consider the Bradley-Terry model and derive the update rules using the full-pair setting in (40). Following the discussion in Equations. (7)-(8), the difference $X_i - X_q$ between two teams follows a logistic distribution. However, by comparing the Thurstone-Mosteller model (5) and the Bradley-Terry model (7), clearly the Bradley-Terry model lacks variance parameters β_i^2 and β_q^2, which account for the performance uncertainty. We thus extend the Bradley-Terry model to include variance parameters; see Appendix C. The resulting model is

$$P(\text{team } i \text{ beats } q) \equiv f_{iq}(\boldsymbol{z}) = \frac{e^{\theta_i/c_{iq}}}{e^{\theta_i/c_{iq}} + e^{\theta_q/c_{iq}}}, \tag{48}$$

where

$$c_{iq}^2 = \beta_i^2 + \beta_q^2 \text{ and } \theta_i = \mu_i + \sigma_i z_i.$$

The parameter β_i is the uncertainty about the actual performance X_i. However, in the model specification, the uncertainty of X_i is not related to σ_i. Following the error analysis of the approximation in Section 3.2 for the Thurstone-Mosteller model, we show in Appendix C that σ_i^2 can be incorporated to

$$\beta_i^2 = \sigma_i^2 + \beta^2,$$

where β^2 is some positive constant.

Algorithm 1 Update rules using the Bradley-Terry model with full-pair

1. Given a game result and the current $\mu_{ij}, \sigma^2_{ij}, \forall i, \forall j$. Given β^2 and $\kappa > 0$. Decide a way to set γ_q in (50)
2. For $i = 1, \ldots, k$, set
$$\mu_i = \sum_{j=1}^{n_i} \mu_{ij}, \quad \sigma^2_i = \sum_{j=1}^{n_i} \sigma^2_{ij}.$$
3. For $i = 1, \ldots, k$,
   - 3.1. Team skill update: obtain Ω_i and Δ_i in (27) and (28) by the following steps.
     - 3.1.1. For $q = 1, \ldots, k, q \neq i$,
$$c_{iq} = (\sigma^2_i + \sigma^2_q + 2\beta^2)^{1/2}, \qquad \hat{p}_{iq} = \frac{e^{\mu_i/c_{iq}}}{e^{\mu_i/c_{iq}} + e^{\mu_q/c_{iq}}}, \tag{49}$$
$$\delta_q = \frac{\sigma^2_i}{c_{iq}}(s - \hat{p}_{iq}), \quad \eta_q = \gamma_q \Big(\frac{\sigma_i}{c_{iq}}\Big)^2 \hat{p}_{iq}\hat{p}_{qi}, \text{ where } s = \begin{cases} 1 & \text{if } r(q) > r(i), \\ 1/2 & \text{if } r(q) = r(i), \\ 0 & \text{if } r(q) < r(i). \end{cases} \tag{50}$$
     - 3.1.2. Calculate
$$\Omega_i = \sum_{q:q\neq i} \delta_q, \qquad \Delta_i = \sum_{q:q\neq i} \eta_q.$$
   - 3.2. Individual skill update

     For $j = 1, \ldots, n_i$,
$$\mu_{ij} \leftarrow \mu_{ij} + \frac{\sigma^2_{ij}}{\sigma^2_i}\Omega_i, \qquad \sigma^2_{ij} \leftarrow \sigma^2_{ij} \max\left(1 - \frac{\sigma^2_{ij}}{\sigma^2_i}\Delta_i, \kappa\right).$$

There are several extensions to the Bradley-Terry model incorporating ties. In Glicko (Glickman, 1999), a tie is treated as a half way between a win and a loss when constructing the likelihood function. That is,

$$\begin{aligned} P(i \text{ draws with } q) &= (P(i \text{ beats } q)P(q \text{ beats } i))^{1/2} \\ &= \sqrt{f_{iq}(\boldsymbol{z}) f_{qi}(\boldsymbol{z})}. \end{aligned} \tag{51}$$

By considering all pairs, the resulting $f(\boldsymbol{z})$ is (40). To obtain update rules (27)-(28), we need to calculate $\partial f / \partial z_i$. We see that terms related to z_i in the product form of (40) are

$$P(\text{outcome of } i \text{ and } q), \forall q = 1, \ldots, k, q \neq i. \tag{52}$$

With (38) and (51),

$$\begin{aligned}&\frac{\partial f/\partial z_i}{f} \\ &= \sum_{q:r(q)<r(i)} \frac{\partial f_{qi}/\partial z_i}{f_{qi}} + \sum_{q:r(q)>r(i)} \frac{\partial f_{iq}/\partial z_i}{f_{iq}} + \frac{1}{2} \sum_{q:r(q)=r(i),q\neq i} \left(\frac{\partial f_{qi}/\partial z_i}{f_{qi}} + \frac{\partial f_{iq}/\partial z_i}{f_{iq}} \right).\end{aligned} \tag{53}$$

Using (24) and (48), it is easy to calculate that

$$\frac{\partial f_{qi}}{\partial z_i} = \frac{-e^{\theta_i/c_{iq}} e^{\theta_q/c_{iq}}}{c_{iq}(e^{\theta_i/c_{iq}} + e^{\theta_q/c_{iq}})^2} \cdot \frac{\partial \theta_i}{\partial z_i} = \frac{-\sigma_i}{c_{iq}} f_{iq} f_{qi} \tag{54}$$

and

$$\frac{\partial f_{iq}}{\partial z_i} = \frac{(e^{\theta_i/c_{iq}} + e^{\theta_q/c_{iq}}) e^{\theta_i/c_{iq}} - e^{\theta_i/c_{iq}} e^{\theta_i/c_{iq}}}{c_{iq}(e^{\theta_i/c_{iq}} + e^{\theta_q/c_{iq}})^2} \cdot \sigma_i = \frac{\sigma_i}{c_{iq}} f_{iq} f_{qi}.$$

Therefore, an update rule following (27) and (29) is

$$\mu_i \leftarrow \mu_i + \Omega_i, \tag{55}$$

where

$$\Omega_i = \sigma_i^2 \left(\sum_{q:r(q)<r(i)} \frac{-\hat{p}_{iq}}{c_{iq}} + \sum_{q:r(q)>r(i)} \frac{\hat{p}_{qi}}{c_{iq}} + \frac{1}{2} \sum_{q:r(q)=r(i),q\neq i} \left(\frac{-\hat{p}_{iq}}{c_{iq}} + \frac{\hat{p}_{qi}}{c_{iq}} \right) \right) \tag{56}$$

and

$$\hat{p}_{iq} \equiv \frac{e^{\mu_i/c_{iq}}}{e^{\mu_i/c_{iq}} + e^{\mu_q/c_{iq}}} \tag{57}$$

is an estimate of $P(\text{team } i \text{ beats team } q)$. Since $\hat{p}_{iq} + \hat{p}_{qi} = 1$, (56) can be rewritten as

$$\Omega_i = \sum_{q:q\neq i} \frac{\sigma_i^2}{c_{iq}} (s - \hat{p}_{iq}), \qquad \text{where } s = \begin{cases} 1 & \text{if } r(q) > r(i), \\ \frac{1}{2} & \text{if } r(q) = r(i), \\ 0 & \text{if } r(q) < r(i). \end{cases} \tag{58}$$

To apply (26) and (30) for updating σ_i, we use (53) to obtain

$$\begin{aligned}\frac{\partial}{\partial z_i}\left(\frac{\partial f/\partial z_i}{f}\right) &= \sum_{q:r(q)<r(i)} \frac{\partial}{\partial z_i}\left(\frac{\partial f_{qi}/\partial z_i}{f_{qi}}\right) + \sum_{q:r(q)>r(i)} \frac{\partial}{\partial z_i}\left(\frac{\partial f_{iq}/\partial z_i}{f_{iq}}\right) \\ &+ \frac{1}{2} \sum_{q:r(q)=r(i),q\neq i} \left(\frac{\partial}{\partial z_i}\left(\frac{\partial f_{qi}/\partial z_i}{f_{qi}}\right) + \frac{\partial}{\partial z_i}\left(\frac{\partial f_{iq}/\partial z_i}{f_{iq}}\right) \right).\end{aligned} \tag{59}$$

From (54),

$$\frac{\partial}{\partial z_i}\left(\frac{\partial f_{qi}/\partial z_i}{f_{qi}}\right) = \frac{\partial(-f_{iq}/c_{iq})}{\partial z_i} = -\frac{\sigma_i^2}{c_{iq}^2} f_{iq} f_{qi}$$

and similarly

$$\frac{\partial}{\partial z_i}\left(\frac{\partial f_{iq}/\partial z_i}{f_{iq}}\right) = -\frac{\sigma_i^2}{c_{iq}^2} f_{iq} f_{qi}. \tag{60}$$

From (30), by setting $\boldsymbol{z} = \mathbf{0}$, Δ_i should be the sum of (60) over all $q \neq i$. However, we mentioned in the end of Section 3.1 that controlling the reduction of σ_i^2 is sometimes important. In particular, σ_i^2 should not be reduced too fast. Hence we introduce an additional parameter γ_q so that the update rule is

$$\sigma_i^2 \leftarrow \sigma_i^2 \max\left(1 - \sum_{q:q\neq i} \gamma_q \xi_q, \kappa\right),$$

where

$$\xi_q = \frac{\sigma_i^2}{c_{iq}^2} \hat{p}_{iq} \hat{p}_{qi}$$

is from (60) and $\gamma_q \leq 1$ is decided by users; further discussions on the choice of γ_q are in Section 6. Algorithm 1 summarizes the procedure.

The formulas (55) and (58) resemble the Elo system. The Elo treats θ_i as nonrandom and its update rule is in (6):

$$\theta_i \leftarrow \theta_i + K(s - p_{iq}^*),$$

where K is a constant (e.g., $K = 32$ in the USCF system for amateur players) and

$$p_{iq}^* = \frac{10^{\theta_i/400}}{10^{\theta_i/400} + 10^{\theta_q/400}}$$

is the approximate probability that i beats q; see Equations. (11) and (12) in Glickman (1999). Observe that p_{iq}^* is simply a variance free and reparameterized version of $\hat{p}_{iq}$ in (57). As for Glicko, it is a Bayesian system but designed for paired comparisons over a rating period. Detailed comparisons with Glicko are in Section 5.

4. Update Rules Using Other Ranking Models

If we assume different distributions of the team performance X_i or model the game results by other ways than the Bradley-Terry model, the same framework in Sections 3.1-3.3 can still be applied. In this section, we present several variants of our proposed method.

4.1 Bradley-Terry Model (Partial-pair)

We now consider the partial-pair approach in (39). With the definition of $\bar{r}$ in (2), the function $f(\boldsymbol{z})$ can be written as

$$f(\boldsymbol{z}) = \prod_{a=1}^{k-1} \bar{f}_{\bar{r}(a)\bar{r}(a+1)}(\boldsymbol{z}), \tag{63}$$

Algorithm 2 Update rules using the Bradley-Terry model with partial-pair

---

The procedure is the same as Algorithm 1 except Step 3:

3. Let $\bar{r}(a), a = 1, \ldots, k$ be indices of teams ranked from the first to the last

   For $a = 1, \ldots, k$,

   3.1. Team skill update: let $i \equiv \bar{r}(a)$ and obtain Ω_i and Δ_i in (27) and (28) by the following steps.

   3.1.1. Define a set Q as

$$Q \equiv \begin{cases} \{\bar{r}(a+1)\} & \text{if } a = 1, \\ \{\bar{r}(a-1)\} & \text{if } a = k, \\ \{\bar{r}(a-1), \bar{r}(a+1)\} & \text{otherwise.} \end{cases} \quad (61)$$

   For $q \in Q$

   Calculate δ_q, η_q by the same way as (49)-(50) of Algorithm 1.

   3.1.2. Calculate

$$\Omega_i = \sum_{q \in Q} \delta_q \quad \text{and} \quad \Delta_i = \sum_{q \in Q} \eta_q. \quad (62)$$

   3.2 Individual skill update: same as Algorithm 1.

---

where we define $\bar{f}_{\bar{r}(a)\bar{r}(a+1)}(\boldsymbol{z})$ as follows:

$$i \equiv \bar{r}(a), \quad q \equiv \bar{r}(a+1),$$
$$\bar{f}_{iq} = \begin{cases} f_{iq} & \text{if } r(i) < r(q), \\ \sqrt{f_{iq} f_{qi}} & \text{if } r(i) = r(q). \end{cases} \quad (64)$$

Note that f_{iq} and f_{qi} are defined in (48) of Section 3.5. Since the definition of $\bar{r}$ in (2) ensures $r(i) \leq r(q)$, in (64) we do not need to handle the case of $r(i) > r(q)$. By a derivation similar to that in Section 3.5, we obtain update rules in Algorithm 2. Clearly, Algorithm 2 differs from Algorithm 1 in only Step 3. The reason is that $\partial f(\boldsymbol{z})/\partial z_i$ is only related to game outcomes between $\bar{r}(a)$ and teams of adjacent ranks, $\bar{r}(a-1)$ and $\bar{r}(a+1)$. In (61), we let Q be the set of these teams. Thus, Q contains at most two elements, and Ω_i and Δ_i in (62) are calculated using δ_q and η_q with $q \in Q$. Details of the derivation are in Appendix D.

4.2 Thurstone-Mosteller Model (Full-pair and Partial-pair)

In this section, we consider the Thurstone-Mosteller model by assuming that the actual performance of team i is

$$X_i \sim N(\theta_i, \beta_i^2),$$

where $\beta_i^2 = \sigma_i^2 + \beta^2$ as in Section 3.5. The performance difference $X_i - X_q$ follows a normal distribution $N(\theta_i - \theta_q, c_{iq}^2)$ with $c_{iq}^2 = \sigma_i^2 + \sigma_q^2 + 2\beta^2$. If one considers partial pairs

$$P(\text{team } i \text{ beats team } q) = P(X_i > X_q) = \Phi\left(\frac{\theta_i - \theta_q}{c_{iq}}\right)$$

and uses (51) to obtain $P(i \text{ draws with } q)$, then a derivation similar to that for the Bradley-Terry model leads to certain update rules. Instead, here we follow Herbrich et al. (2007) to let ϵ be the draw margin that depends on the game mode and assume that the probabilities that i beats q and a draw occurs are respectively

$$P(\text{team } i \text{ beats team } q) = P(X_i > X_q + \epsilon) = \Phi\left(\frac{\theta_i - \theta_q - \epsilon}{c_{iq}}\right)$$

and

$$\begin{aligned} &P(\text{team } i \text{ draws with } q) = P(|X_i - X_q| < \epsilon) \\ =&\Phi\left(\frac{\epsilon - (\theta_i - \theta_q)}{c_{iq}}\right) - \Phi\left(\frac{-\epsilon - (\theta_i - \theta_q)}{c_{iq}}\right). \end{aligned} \tag{65}$$

We can then obtain $f(\boldsymbol{z})$ using the full-pair setting (40). The way to derive update rules is similar to that for the Bradley-Terry model though some details are different. We summarize the procedure in Algorithm 3. Detailed derivations are in Appendix E.

Interestingly, if $k = 2$ (i.e., two teams), then the update rules (if i beats q) in Algorithm 3 are reduced to

$$\begin{aligned} \mu_i \leftarrow&\mu_i + \frac{\sigma_i^2}{c_{iq}} V\left(\frac{\mu_i - \mu_q}{c_{iq}}, \frac{\epsilon}{c_{iq}}\right), \\ \mu_q \leftarrow&\mu_q - \frac{\sigma_q^2}{c_{iq}} V\left(\frac{\mu_i - \mu_q}{c_{iq}}, \frac{\epsilon}{c_{iq}}\right), \end{aligned}$$

where the function V is defined in (67). These update rules are the same as the case of $k = 2$ in the TrueSkill system (see `http://research.microsoft.com/en-us/projects/trueskill/details.aspx`).

As a comparison, we note that TrueSkill considers partial-pair and obtains players' skills by a factor graph and the approximate message passing. In fact, some messages in the factor graph are non Gaussian and these messages are approximated via moment matching, using the *Expectation Propagation* algorithm (Minka, 2001). Their algorithm is effective, but simple update rules are not available for the cases of multiple teams/players.

4.3 Plackett-Luce Model

We now discuss the situation of using the Plackett-Luce model. If ties are not allowed, an extension of the Plackett-Luce model (9) incorporating variance parameters is

$$f(\boldsymbol{z}) = \prod_{q=1}^{k} f_q(\boldsymbol{z}) = \prod_{q=1}^{k}\left(\frac{e^{\theta_q/c}}{\sum_{s\in C_q} e^{\theta_s/c}}\right), \tag{70}$$

Algorithm 3 Update rules using Thurstone-Mosteller model with full-pair

The procedure is the same as Algorithm 1 except Step 3.1.1:

3.1.1 For $q = 1, \dots, k; q \neq i$,

$$\delta_q = \frac{\sigma_i^2}{c_{iq}} \times \begin{cases} V(\frac{\mu_i-\mu_q}{c_{iq}}, \frac{\epsilon}{c_{iq}}) & \text{if } r(q) > r(i), \\ \tilde{V}(\frac{\mu_i-\mu_q}{c_{iq}}, \frac{\epsilon}{c_{iq}}) & \text{if } r(q) = r(i), \\ -V(\frac{\mu_q-\mu_i}{c_{iq}}, \frac{\epsilon}{c_{iq}}) & \text{if } r(q) < r(i), \end{cases}$$

$$\eta_q = (\frac{\sigma_i}{c_{iq}})^2 \times \begin{cases} W(\frac{\mu_i-\mu_q}{c_{iq}}, \frac{\epsilon}{c_{iq}}) & \text{if } r(q) > r(i), \\ \tilde{W}(\frac{\mu_i-\mu_q}{c_{iq}}, \frac{\epsilon}{c_{iq}}) & \text{if } r(q) = r(i), \\ W(\frac{\mu_q-\mu_i}{c_{iq}}, \frac{\epsilon}{c_{iq}}) & \text{if } r(q) < r(i), \end{cases}$$

where

$$c_{iq} = (\sigma_i^2 + \sigma_q^2 + 2\beta^2)^{1/2},$$

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \quad \Phi(x) = \int_{-\infty}^{x} \phi(u) du, \tag{66}$$

$$V(x,t) = \phi(x-t)/\Phi(x-t), \quad W(x,t) = V(x,t)(V(x,t) + (x-t)), \tag{67}$$

$$\tilde{V}(x,t) = -\frac{\phi(t-x) - \phi(-t-x)}{\Phi(t-x) - \Phi(-t-x)}, \tag{68}$$

$$\tilde{W}(x,t) = \frac{(t-x)\phi(t-x) - (-(t+x))\phi(-(t+x))}{\Phi(t-x) - \Phi(-t-x)} + \tilde{V}(x,t)^2. \tag{69}$$

where

$$z_i = \frac{\theta_i - \mu_i}{\sigma_i}, c = \left(\sum_{i=1}^{k} (\sigma_i^2 + \beta^2)\right)^{1/2} \text{ and } C_q = \{i : r(i) \geq r(q)\}.$$

Instead of the same c in $e^{\theta_q/c}$, similar to the Bradley-Terry model, we can define c_q to sum up $\sigma_i^2, i \in C_q$. However, here we take the simpler setting of using the same c. Note that $f_q(\boldsymbol{z})$ corresponds to the probability that team q is the winner among teams in C_q. In (9), $f(\boldsymbol{z})$ is represented using $\bar{r}(1), \dots, \bar{r}(k)$, but (70) is a reformulation using $r(1), \dots, r(k)$.

We extend this model to allow ties. If teams $i_1, \dots, i_d$ are tied together, then $r(i_1) = \cdots = r(i_d)$. A generalization of the tie probability (51) gives the likelihood based on these d stages as:

$$\left(\frac{e^{\theta_{i_1}/c}}{\sum_{s:r(s) \geq r(i_1)} e^{\theta_s/c}} \times \cdots \times \frac{e^{\theta_{i_d}/c}}{\sum_{s:r(s) \geq r(i_d)} e^{\theta_s/c}} \right)^{1/d}. \tag{71}$$

Algorithm 4 Update rules using the Plackett-Luce model

The procedure is the same as Algorithm 1 except Step 3:

3. Find and store

$$c = \left(\sum_{i=1}^{k}(\sigma_i^2+\beta^2)\right)^{1/2},$$
$$A_q = |\{s : r(s) = r(q)\}|, \quad q = 1,\dots,k$$
$$\sum_{s\in C_q} e^{\theta_s/c}, q = 1,\dots,k, \quad \text{where } C_q = \{i : r(i) \ge r(q)\}.$$

For $i = 1,\dots,k$,

3.1. Team skill update: obtain Ω_i and Δ_i in (27) and (28) by the following steps.

3.1.1. For $q = 1,\dots,k$,

$$\delta_q = \frac{\sigma_i^2}{cA_q} \times \begin{cases} 1-\hat{p}_{i,C_q} & \text{if } q = i, \\ -\hat{p}_{i,C_q} & \text{if } r(q) \le r(i), q \ne i, \\ 0 & \text{if } r(q) > r(i), \end{cases}$$

$$\eta_q = \frac{\gamma_q \sigma_i^2}{c^2 A_q} \times \begin{cases} \hat{p}_{i,C_q}(1-\hat{p}_{i,C_q}) & \text{if } r(q) \le r(i), \\ 0 & \text{if } r(q) > r(i), \end{cases}$$

where

$$\hat{p}_{i,C_q} = \frac{e^{\theta_i/c}}{\sum_{s\in C_q} e^{\theta_s/c}}.$$

3.1.2 Same as Algorithm 1.

3.2 Same as Algorithm 1.

We can explain (71) as follows. Now d factors in (71) all correspond to the likelihood of the same rank, so we multiply them and take the dth root. The new $f(\boldsymbol{z})$ becomes

$$f(\boldsymbol{z}) = \prod_{q=1}^{k} f_q(\boldsymbol{z}) = \prod_{q=1}^{k}\left(\frac{e^{\theta_q/c}}{\sum_{s\in C_q} e^{\theta_s/c}}\right)^{1/A_q}, \tag{72}$$

where

$$A_q = |\{s : r(s) = r(q)\}| \text{ and } f_q(\boldsymbol{z}) = \left(\frac{e^{\theta_q/c}}{\sum_{s\in C_q} e^{\theta_s/c}}\right)^{1/A_q}, \quad q = 1,\dots,k.$$

If ties do not occur, $A_q = 1$, so (72) goes back to (70). By calculations shown in Appendix F, the update rules are in Algorithm 4.

Algorithm 5 Update rules of Glicko with a single game

1. Given a game result and the current $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2$. Set
$$q = \frac{\log 10}{400}. \tag{73}$$

2. For $i = 1, 2$
$$g(\sigma_i^2) = \frac{1}{\sqrt{1 + \frac{3q^2\sigma_i^2}{\pi^2}}}. \tag{74}$$

3. For $i = 1, 2$, set $j \neq i$ and
$$p_j^* = \frac{1}{1 + 10^{-g(\sigma_j^2)(\mu_i - \mu_j)/400}}, \qquad (\delta_i^2)^* = \left[q^2 (g(\sigma_j^2))^2 p_j^* (1 - p_j^*)\right]^{-1}.$$

4. Update rule: For $i = 1, 2$, set $j \neq i$
$$\begin{aligned} \mu_i &\leftarrow \mu_i + \frac{q}{\frac{1}{\sigma_i^2} + \frac{1}{(\delta_i^2)^*}} g(\sigma_j^2)(s_{ij} - p_j^*), \text{ where } s_{ij} = \begin{cases} 1 & \text{if } i \text{ wins,} \\ 1/2 & \text{if draw,} \\ 0 & \text{if } i \text{ loses,} \end{cases} \\ \sigma_i^2 &\leftarrow \left(\frac{1}{\sigma_i^2} + \frac{1}{(\delta_i^2)^*}\right)^{-1}. \end{aligned}$$

5. Description of Glicko

Since our Algorithm 1 and the Glicko system are both based on the Bradley-Terry model, it is of interest to compare these two algorithms. We describe the derivation of Glicko in this section. Note that notation in this section may be slightly different from other sections of this paper.

Consider a rating period of paired comparisons. Assume that prior to a rating period the distribution of a player's strength θ is $N(\mu, \sigma^2)$, with μ and σ^2 known. Assume that, during the rating period, the player plays n_j games against opponent j, where $j = 1, \ldots, m$, and that the jth opponent's strength θ_j follows $N(\mu_j, \sigma_j^2)$, with μ_j and σ_j^2 known. Let s_{jk} be the outcome of the kth game against opponent j, with $s_{jk} = 1$ if the player wins, $s_{jk} = 0.5$ if the game results in a tie, and $s_{jk} = 0$ if the player loses. Let D be the collection of game results during this period. The interest lies in the marginal posterior distribution of θ given D:

$$P(\theta|D) = \int \cdots \int P(\theta_1, \ldots, \theta_m|D) P(\theta|\theta_1, \ldots, \theta_m, D) d\theta_1 \cdots d\theta_m, \tag{75}$$

where $P(\theta|\theta_1, \ldots, \theta_m, D)$ is the posterior distribution of θ conditional on opponents' strengths,

$$P(\theta|\theta_1, \ldots, \theta_m, D) \propto \phi(\theta|\mu, \sigma^2) P(D|\theta, \theta_1, \ldots, \theta_m). \tag{76}$$

Here $P(D|\theta, \theta_1, \ldots, \theta_m)$ is the likelihood for all parameters. The approximation procedure is described in steps (I)-(V) below, where step (I) is from Section 3.3 of Glickman (1999) and steps (II)-(IV) are summarized from his Appendix A.

(I) Glickman (1999) stated that "The key idea is that the marginal posterior distribution of a player's strength is determined by integrating out the opponents' strengths over their *prior* distribution rather than over their posterior distribution." That is, the posterior distribution of opponents' strengths $P(\theta_1, \ldots, \theta_m|D)$ is approximated by the prior distribution

$$\phi(\theta_1|\mu_1, \sigma_1^2) \cdots \phi(\theta_m|\mu_m, \sigma_m^2).$$

Then, together with (75) and (76) it follows that, approximately

$$\begin{aligned} P(\theta|D) &\propto \phi(\theta|\mu, \sigma^2) \int \cdots \int \phi(\theta_1|\mu_1, \sigma_1^2) \cdots \phi(\theta_m|\mu_m, \sigma_m^2) P(D|\theta, \theta_1, \ldots, \theta_m) d\theta_1 \cdots d\theta_m \\ &\propto \phi(\theta|\mu, \sigma^2) \underbrace{\prod_{j=1}^{m} \left\{ \int \left[\prod_{k=1}^{n_j} \left(\frac{(10^{(\theta-\theta_j)/400})^{s_{jk}}}{1 + 10^{(\theta-\theta_j)/400}} \right) \phi(\theta_j|\mu_j, \sigma_j^2) \right] d\theta_j \right\}}_{P(D|\theta)}, \end{aligned} \tag{77}$$

where the last line follows by treating terms in the likelihood that do not depend on θ (which correspond to games played between other players) as constant. We denote a term in (77) as $P(D|\theta)$ for subsequent analysis.

(II) $P(D|\theta)$ in (77) is the likelihood integrated over the opponents' prior strength distribution. Then, (77) becomes

$$P(\theta|D) \quad \propto \quad \phi(\theta|\mu, \sigma^2) P(D|\theta). \tag{78}$$

In this step, $P(D|\theta)$ is approximated by a product of logistic cumulative distribution functions:

$$P(D|\theta) \quad \approx \quad \prod_{j=1}^{m} \prod_{k=1}^{n_j} \int \frac{(10^{(\theta-\theta_j)/400})^{s_{jk}}}{1 + 10^{(\theta-\theta_j)/400}} \phi(\theta_j|\mu_j, \sigma_j^2) d\theta_j. \tag{79}$$

(III) In this step, $P(D|\theta)$ is further approximated by a normal distribution. First, one approximates each logistic cdf in the integrand of (79) by a normal cdf with the same mean and variance so that the integral can be evaluated in a closed form to a normal cdf. This yields the approximation

$$\int \frac{(10^{(\theta-\theta_j)/400})^{s_{jk}}}{1 + 10^{(\theta-\theta_j)/400}} \phi(\theta_j|\mu_j, \sigma_j^2) d\theta_j \quad \approx \quad \frac{\left(10^{g(\sigma_j^2)(\theta-\mu_j)/400}\right)^{s_{jk}}}{1 + 10^{g(\sigma_j^2)(\theta-\mu_j)/400}},$$

where $g(\sigma_j^2)$ is defined in (74). Therefore, the (approximate) marginal likelihood in (79) is

$$P(D|\theta) \quad \approx \quad \prod_{j=1}^{m} \prod_{k=1}^{n_j} \frac{\left(10^{g(\sigma_j^2)(\theta-\mu_j)/400}\right)^{s_{jk}}}{1 + 10^{g(\sigma_j^2)(\theta-\mu_j)/400}}. \tag{80}$$

Second, by central limit theorem we approximate this marginal likelihood (80) by a normal density $\phi(\theta|\hat{\theta},\delta^2)$, where $\hat{\theta}$ is the mode of this marginal likelihood and δ^2 is minus of inverse of Hessian of the log marginal likelihood evaluated at $\hat{\theta}$. Then, together with (78) we obtain an approximation:

$$
\begin{aligned}
P(\theta|D) &\propto \phi(\theta|\mu,\sigma^2)\phi(\theta|\hat{\theta},\delta^2) \\
&\propto \phi\left(\theta \mid \frac{\frac{\mu}{\sigma^2}+\frac{\hat{\theta}}{\delta^2}}{\frac{1}{\sigma^2}+\frac{1}{\delta^2}}, \left(\frac{1}{\sigma^2}+\frac{1}{\delta^2}\right)^{-1}\right).
\end{aligned}
$$

Therefore, the update of μ and σ^2 (i.e., posterior mean and variance) is:

$$
\sigma^2 \leftarrow \left(\frac{1}{\sigma^2}+\frac{1}{\delta^2}\right)^{-1} \quad \text{and} \quad \mu \leftarrow \frac{\frac{\mu}{\sigma^2}+\frac{\hat{\theta}}{\delta^2}}{\frac{1}{\sigma^2}+\frac{1}{\delta^2}} = \mu + \frac{\frac{1}{\delta^2}}{\frac{1}{\sigma^2}+\frac{1}{\delta^2}}(\hat{\theta}-\mu). \tag{81}
$$

Note that we obtain $\hat{\theta}$ by equating the derivative of $\log P(D|\theta)$ to zero, and approximating δ^2 by substituting μ for $\hat{\theta}$. The expression of approximation for δ^2 is

$$
\delta^2 \approx \big(q^2 \sum_{j=1}^{m} n_j (g(\sigma_j^2))^2 p_j(\mu)(1-p_j(\mu))\big)^{-1}, \tag{82}
$$

where q is defined in (73), $g(\sigma_j^2)$ is defined in (74) and

$$
p_j(\mu) = \frac{1}{1+10^{-g(\sigma_j^2)(\mu-\mu_j)/400}}, \tag{83}
$$

which is an approximate probability that the player beats opponent j.

(IV) Finally, $\hat{\theta}-\mu$ in (81) is approximated as follows. From (80) it follows that

$$
\frac{d}{d\theta}\log P(D|\theta) \approx \sum_{j=1}^{m}\sum_{k=1}^{n_j} \frac{\log 10}{400}\left\{g(\sigma_j^2)\left(s_{jk} - \frac{1}{1+10^{-g(\sigma_j^2)(\theta-\mu_j)/400}}\right)\right\}. \tag{84}
$$

If we define

$$
h(\theta) = \sum_{j=1}^{m}\sum_{k=1}^{n_j} \frac{g(\sigma_j^2)}{1+10^{-g(\sigma_j^2)(\theta-\mu_j)/400}}, \tag{85}
$$

then setting the right-hand side of (84) to zero gives

$$
h(\hat{\theta}) = \sum_{j=1}^{m}\sum_{k=1}^{n_j} g(\sigma_j^2)s_{jk}. \tag{86}
$$

Then, a Taylor series expansion of $h(\theta)$ around μ gives

$$
h(\hat{\theta}) \approx h(\mu) + (\hat{\theta}-\mu)h'(\mu), \tag{87}
$$

where

$$
h'(\mu) = q\sum_{j=1}^{m}\sum_{k=1}^{n_j}(g(\sigma_j^2))^2 p_j(\mu))(1-p_j(\mu)) = q\sum_{j=1}^{m} n_j (g(\sigma_j^2))^2 p_j(\mu))(1-p_j(\mu)) \tag{88}
$$

Game type	# games	# players	BT-full	BT-partial	PL	TM-full	TrueSkill
Free for All	5,943	60,022	**30.59**%	32.40%	31.74%	44.65%	30.82%
Small Teams	27,539	4,992	**33.97**%	**33.97**%	**33.97**%	36.46%	35.23%
Head to Head	6,227	1,672	32.53%	32.53%	32.53%	**32.41**%	32.44%
Large Teams	1,199	2,576	**37.30**%	**37.30**%	**37.30**%	39.37%	38.15%

Table 2: Data description and prediction errors by various methods. The method with the smallest error is bold-faced. The column "TrueSkill" is copied from a table in Herbrich et al. (2007). Note that we use the same way as TrueSkill to calculate prediction errors.

Game type	BT-full	PL	TM-full
Free for All	31.24%	31.73%	33.13%
Small Teams	33.84%	33.84%	36.50%
Head to Head	32.55%	32.55%	32.74%
Large Teams	37.30%	37.30%	39.13%

Table 3: Prediction errors using $\gamma_q = 1/k$ in (50), where k is the number of teams in a game.

with $p_j(\mu)$ defined in (83). Using (86), $h(\mu)$ by (85), and (88), we can apply (87) to obtain an estimate of $\hat{\theta} - \mu$. Then with (82), (81) becomes

$$\mu \quad \leftarrow \quad \mu + \frac{q}{\frac{1}{\sigma^2} + \frac{1}{\delta^2}} \sum_{j=1}^{m} \sum_{k=1}^{n_j} g(\sigma_j^2)(s_{jk} - p_j(\mu)).$$

However, when there is only one game, $P(D|\theta)$ in (80) would have just one term (because $m = 1$ and $n_1 = 1$), and it is a monotone function. Therefore, the mode $\hat{\theta}$ of $P(D|\theta)$ would be either ∞ or $-\infty$ and the central limit theorem can not be applied. Although this problem seems to disappear when the approximation in step (IV) is employed, the justification of the whole procedure may be weak. In fact, the Glicko system treats a collection of games within a "rating period" to have simultaneous occurrences, and it works best when the number of games in a rating period is moderate, say an average of 5-10 games per player in a rating period.[2] The Glicko algorithm for a single game is in Algorithm 5.

6. Experiments

We conduct experiments to assess the performance of our algorithms and TrueSkill on the game data set used by Herbrich et al. (2007). The data are generated by Bungie Studios during the beta testing of the Xbox title Halo 2.[3] The set contains data from four different game types:

- Free for All: up to 8 players in a game. Each team has a single player.
- Small Teams: up to 12 players in 2 teams.[4]

2. According to `http://math.bu.edu/people/mg/glicko/glicko.doc/glicko.html`.
3. Credits for the use of the Halo 2 Beta Data set are given to Microsoft Research Ltd. and Bungie.
4. Herbrich et al. (2007) indicate that for "Small Teams," each team has no more than 4 players, and for "Large Teams," each has no more than 8. However, we find a few exceptions.

- Head to Head: 2 players in a game. Each player is considered as a team.
- Large Teams: up to 16 players in 2 teams.

The numbers of games and players are given in Table 2. In the following, let BT, TM, and PL denote Bradley-Terry, Thurstone-Mosteller, and Plackett-Luce models, respectively; BT-full and BT-partial denote BT with full-pair and partial-pair, and similarly for TM-full and TM-partial. The TrueSkill code is obtained at `http://blogs.technet.com/apg/archive/2008/06/16/trueskill-in-f.aspx`.

6.1 Implementation and Evaluation

Below we discuss initial values and parameters. Generally we follow the setting in Herbrich et al. (2007).

- Initial $\mu_i = 25$ and $\sigma_i^2 = (25/3)^2$, $\forall i$.
- The additional variance of performance $\beta^2 = (25/6)^2$.
- $\epsilon = 0.1$ is the draw margin in (65) for the Thurstone-Mosteller model.
- $\kappa = 0.0001$ is the positive lower bound in (28) to avoid negative σ_i^2. The result is insensitive to this parameter as in general $1 - \Delta_i$ is larger than κ.
- γ_q in (50) is set as σ_i/c_{iq} for BT-full. The same γ_q is applied to BT-partial and TM-full. For PL, we use $\gamma_q = \sigma_i/c$. The use of γ_q is further discussed later in this section.

The update rules for the Thurstone-Mosteller model need to calculate the cumulative distribution function $\Phi(x)$, which is not available in most programming languages. We adopt the same way as in TrueSkill to implement the function $\Phi(x)$. Moreover, if the Thurstone-Mosteller model is used, some numerical difficulties may occur. When $x - t$ in (67) is small,

$$\phi(x-t) \approx 0 \text{ and } \Phi(x-t) \approx 0, \tag{89}$$

so the calculation of $V(x,t)$ via $\phi(x-t)/\Phi(x-t)$ is inaccurate. We employ the same safeguard as in TrueSkill:

$$\text{If } \Phi(x-t) \leq 2.222758749 \times 10^{-162}, \text{ then } V(x,t) \text{ is assigned as } -x+t.$$

Note that $-x+t$ is the limit of $V(x,t)$ when $x - t \to -\infty$. We also need some safeguards in calculating $\tilde{V}$ and $\tilde{W}$.

We implement our methods in both C and F#. The F# code is used for the running time comparison with TrueSkill, which is also written in F#. On the same computer, TrueSkill takes 13 seconds to run the "Free for All" data, but BT-full needs only 1.2 seconds. Our method is more efficient because it uses analytic update rules. In contrast, TrueSkill requires an iterative procedure. Moreover, it is simpler to implement our update rules. Using F#, our code takes less than 100 lines, but TrueSkill needs more than 500 lines. Sources used for experiments in this paper are available at

`http://www.csie.ntu.edu.tw/~cjlin/papers/online_ranking`

For the evaluation of prediction results, following Herbrich et al. (2007), we consider the error of using the current $\boldsymbol{\mu}$ to predict the outcome of the next game. We check only team pairs whose ranks are different. For example, if there are three teams A, B, and C and the rank of one game is $(1,1,2)$, then only the two pairs (A,C) and (B,C) count. Further, if

Game type	BT-full	BT-partial	PL	TM-full	TrueSkill
Free for All	**35.44%**	36.70%	36.31%	46.11%	35.58%

Table 4: Prediction errors (difficult cases). Team pairs with rank differences no more than two are considered. We consider only "Free for All" because the TrueSkill code provided by authors does not handle multi-player teams and we have not conducted suitable modifications. Moreover, under our selection rule, all games in "Head to Head" will be selected and results are the same as Table 2. Hence this set is not included either.

Avg. Occurances	Num. Pairs	BT-full	TrueSkill
≤5	23,567	38.74%	39.15%
≤10	69,145	36.22%	36.41%
≤20	148,654	34.54%	34.52%
≤40	276,203	32.64%	32.64%
No restriction	595,500	30.59%	30.74%

(a) Free for All

Avg. Occurances	Num. Pairs	BT-full	TrueSkill
≤5	2,367	38.70%	38.36%
≤10	3,748	35.17%	34.61%
≤20	4,852	33.29%	33.02%
≤40	5,501	32.61%	32.61%
No restriction	5,715	32.53%	32.49%

(b) Head to Head

Table 5: Prediction errors for competitions where players have only played few games. Games with the average number of players' past appearances no more than the value in the first column are considered. The last row includes all games. The second column indicates the number of total team pairs used for the evaluation. The 30.74% and 32.49% rates by TrueSkill are slightly different from 30.82% and 32.44% in Table 2, respectively, because the former is from running the F# code provided by TrueSkill authors, but the latter is copied from Herbrich et al. (2007).

before the game we have $\mu_A = \mu_C$ and the game output shows $\text{rank}(A) < \text{rank}(C)$, it is considered a wrong prediction. This situation seldom happens as $\boldsymbol{\mu}$ is a real-valued vector, but it does occur in early games because all players' $\boldsymbol{\mu}$ were set equally in the beginning. We have confirmed with TrueSkill authors that these detailed settings are the same as what they used in Herbrich et al. (2007). The prediction error rate is the fraction of total team pairs (from the second to the last game) that are wrongly predicted.

6.2 Comparison on Prediction Errors

We report the prediction error in Table 2 and make the following observations. First, BT-full, BT-partial, and PL have the same error rate except "Free for All." This result is reasonable as when every game involves only two teams, using full pairs, partial pairs or the Plackett-Luce model does not make any difference. Second, when the number of teams is more than two (i.e., Free for All), BT-full is better than BT-partial. The same observation holds when comparing TM-full and TM-partial (numbers not shown). A possible explanation is that the full-pair approach uses more information. Third, using the Bradley-Terry model yields superior results to the Thurstone-Mosteller model. The error of using TM-full on "Free for All" is very high. Besides, numerical problems discussed in (89) do not occur for the Bradley-Terry model. Fourth, TM-full, which uses the same likelihood model as TrueSkill, is consistently worse than TrueSkill, indicating that the much faster, single-pass approximation may come at the expense of less accurate prediction. Finally, our proposed method for BT-full and PL is competitive with TrueSkill.

The reason why TM-full performs poorly for "Free for All" in Table 2 might be that σ_i quickly goes to zero and μ_i becomes a huge positive/negative value. The parameter γ_q in (50) can help to control how fast the variance σ_i^2 is reduced. In Table 2, γ_q is set as σ_i/c_{iq}. Table 3 gives results of using $\gamma_q = 1/k$, where k is the number of teams in a game. For "Free for All," k is around 8, so γ_q is quite small. Clearly, a slower reduction of σ_i^2 significantly improves the performance of TM-full, while the results of BT-full and PL do not change much.

We conduct a further comparison using only team pairs which are more difficult for prediction. For "Free for All," the team pairs whose ranks in a game are closer can be viewed as difficult cases for prediction. We take all pairs with rank differences no more than two and compare the prediction errors by our methods and TrueSkill. The results, shown in Table 4, are consistent with those in Table 2.

After a team (or player) has played many games, the obtained ability becomes more accurate. To check the performance when teams have only played few games, we select games where the average number of players' past appearances is small. We present results in Table 5. Clearly if players in a game have only played few games, the prediction is more difficult.

We also implement the single game version of Glicko (Algorithm 5) for "Head to Head" and find the prediction error to be 33.88%, a bit worse than those in Table 2. Such a result is expected as Glicko is not designed to update skills after each single game.

Finally, we discuss how to apply our proposed technique in practice. Following the experimental results and the numerical concerns, TM is not recommended. Further as BT-full is slightly better than BT-partial, it seems that to factorize a multi-team game to several two-team games, we should use as much information as possible. Therefore, in applying our approximation, BT-full and PL may be the first choice. As TM-full uses the same likelihood as TrueSkill and performs worse, our approximation, while very simple, may be more sensitive to the likelihood used.

7. Discussion and Conclusions

Huang and Frey (2008) propose a graphical model, cumulative distribution network (CDF), which can be used for online ranking. They experiment with the same data used by Herbrich et al. (2007) and report superior results. However, they use a full covariance matrix over all skills of all players. This setting provides more information for accurate predictions, but may not be practical for large-scale systems.

Guiver and Snelson (2009) apply Power EP (expectation propagation) to perform Bayesian inference for parameters of the Plackett-Luce model. They conduct experiments in an offline setting on NASCAR 2002 car racing results and the MovieLens data set. It is worth studying the performance in online setting. We leave it for future work.

In summary, this paper approximates the expectation of teams' performances to derive simple update rules for online ranking. The proposed method is efficient and can be easily applied to large-scale systems with multiple teams and multiple players. While the approximation of the expectation is only a kind of heuristics, experiments show that its application to BT-full and PL models is competitive with state of the art approaches such as TrueSkill. Further, the implementation is simpler and the running time is shorter.

Acknowledgments

The authors thank Ralf Herbrich and Thore Graepel of Microsoft Research Cambridge for providing their F# codes (Free for All and Head to Head) and for answering many questions.

Appendix A. A Sketch of the Proof for Lemma 1

We borrow a few lines from Woodroofe (1989) to sketch the proof for (16) in the 1-dimensional case. Let $'$ denote the differentiation and Φh denote $\int_R h(z)d\Phi(z)$. By assumptions in Lemma 1, we have $f(z) = \int_{-\infty}^{z} f'(y)dy$ and

$$\begin{aligned} & \int_R h(z)d\Gamma(z) - \int_R f(z)d\Phi(z) \cdot \int_R h(z)d\Phi(z) \\ = \ & \int_R f(z)\phi(z)[h(z) - \Phi h]dz \ = \ \int_R \Big\{ \int_{-\infty}^{z} f'(y)dy \Big\} \phi(z)[h(z) - \Phi h]dz \\ = \ & \int_R \Big\{ \int_{y}^{\infty} \phi(z)[h(z) - \Phi h]dz \Big\} f'(y)dy \ = \ \int_R Uh(y)f'(y)\phi(y)dy, \end{aligned}$$

where the interchange of orders of integration is justified by assumed integrability conditions.

Appendix B. An Example on Calculating Uh and Vh in (14)

We take $k = 3$ and $h(\boldsymbol{z}) = z_1 z_2$ to illustrate the calculation of Uh and Vh. First by (12) we obtain

$$\begin{aligned} & h_0 = \int z_1 z_2 d\Phi_3(\boldsymbol{z}) = 0, \\ & h_1(z_1) = \int h(z_1, w_1, w_2)d\Phi_2(w_1, w_2) = \int z_1 w_1 d\Phi_2(w_1, w_2) = 0, \\ & h_2(z_1, z_2) = \int h(z_1, z_2, w)d\Phi_1(w) = \int z_1 z_2 d\Phi(w) = z_1 z_2, \\ & h_3(z_1, z_2, z_3) = h(z_1, z_2, z_3) = z_1 z_2. \end{aligned}$$

Next from (13) it follows that

$$\begin{aligned} g_1(\boldsymbol{z}) &= e^{z_1^2/2} \int_{z_1}^{\infty} [h_1(w) - h_0] e^{-w^2/2} dw = 0, \\ g_2(\boldsymbol{z}) &= e^{z_2^2/2} \int_{z_2}^{\infty} [h_2(z_1, w) - h_1(z_1)] e^{-w^2/2} dw = e^{z_2^2/2} \int_{z_2}^{\infty} z_1 w e^{-w^2/2} dw = z_1, \\ g_3(\boldsymbol{z}) &= e^{z_3^2/2} \int_{z_3}^{\infty} [h_3(z_1, z_2, w) - h_2(z_1, z_2)] e^{-w^2/2} dw = 0; \end{aligned}$$

hence, by (14) we have $Uh(\boldsymbol{z}) = (g_1, g_2, g_3)^T = (0, z_1, 0)^T$. Applying the same steps to g_i gives $Ug_1 = Ug_3 = [0,0,0]^T$ and $Ug_2 = [1,0,0]^T$. Therefore, by (14) we obtain

$$Vh = \frac{1}{2}\left(U^2h + (U^2h)^T\right) = \frac{1}{2}\left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Appendix C. A Bradley-Terry Model with Variance Parameters

Our approach is motivated by the relation between the normal model (5) and the Bradley-Terry model (7). To begin, we reparametrize v_i in (7) as $e^{\theta_i/c}$ and similarly for v_q so that (7) can be written as

$$P(X_i > X_q) = \frac{e^{(\theta_i - \theta_q)/c}}{1 + e^{(\theta_i - \theta_q)/c}}. \tag{90}$$

Next, observe that the cumulative distribution function of a logistic distribution with mean 0 and variance $(c\pi/\sqrt{3})^2$ is

$$F(x) = \frac{e^{x/c}}{(1 + e^{x/c})},$$

which can be approximated by the cumulative distribution function of a normal distribution with the same mean and variance. Therefore,

$$\begin{aligned} \frac{e^{(\theta_i - \theta_q)/c}}{1 + e^{(\theta_i - \theta_q)/c}} &\approx \int_{-\infty}^{\theta_i - \theta_q} \frac{1}{\sqrt{2\pi}(c\pi)/\sqrt{3}} e^{-u^2/(2(c\pi/\sqrt{3})^2)} du \\ &= \Phi\left(\frac{\theta_i - \theta_q}{c\pi/\sqrt{3}}\right). \end{aligned} \tag{91}$$

The idea of approximating the logistic distribution in an integral by a Gaussian one has appeared in Aitchison and Begg (1976), Glickman (1993), and references therein. By comparing (91) with (5), it suggests to take $c^2 \propto (\beta_i^2 + \beta_q^2)$ and then replace v_i and v_q in (7) with $e^{\theta_i/c}$ and $e^{\theta_q/c}$. In summary, we have shown that (90) can be obtained by assuming that each team has a performance uncertainty parameter β_i^2, and that when teams i and q compete, their actual performance follow Gumbel distributions with cumulative distribution function

$$P(X_i \le x) = \exp(-\exp(-(x - \frac{\theta_i}{c}))),$$

where $c^2 = \beta_i^2 + \beta_q^2$. Note that this model presumes that a team's actual performance depends on the team it competes with.

Regarding the error induced by evaluating the expectations in (25) and (26), we can apply the same analysis in Section 3.2 to the Bradley-Terry model. Here we give details. By (48), the joint posterior density of (θ_1, θ_2) is proportional to

$$\phi\left(\frac{\theta_1 - \mu_1}{\sigma_1}\right)\phi\left(\frac{\theta_2 - \mu_2}{\sigma_2}\right)\frac{e^{\theta_1/c_{12}}}{e^{\theta_1/c_{12}} + e^{\theta_2/c_{12}}}.$$

Next, by an approximation like (91), the marginal posterior density of θ_1 is approximately proportional to

$$\begin{aligned}
&\phi\left(\frac{\theta_1-\mu_1}{\sigma_1}\right)\int\phi\left(\frac{\theta_2-\mu_2}{\sigma_1}\right)\int_{-\infty}^{\theta_1}\frac{1}{\sqrt{2\pi}(\alpha c_{12})}e^{-\frac{y-\theta_2}{2(\alpha c_{12})^2}}dyd\theta_2\\
\approx\;&\phi\left(\frac{\theta_1-\mu_1}{\sigma_1}\right)\Phi\left(\frac{\theta_1-\mu_2}{\sqrt{(\alpha c_{12})^2+\sigma_2^2}}\right)\\
\approx\;&\phi\left(\frac{\theta_1-\mu_1}{\sigma_1}\right)\frac{e^{\theta_1/c'_{12}}}{e^{\theta_1/c'_{12}}+e^{\theta_2/c'_{12}}},
\end{aligned}$$

where $\alpha=\pi/\sqrt{3}$ as in (91) and $(c'_{12})^2=\alpha^2c_{12}^2+\sigma_2^2$. As in the previous paragraph, we can calculate the posterior mean of θ_1, and again the result suggests that the bias induced by our approximation method can be reduced by substituting β_i^2 with $\beta_i^2+\sigma_i^2$.

Appendix D. Derivations of Update Rules for the Bradley-Terry Model (Partial-pair)

To calculate $\partial f/\partial z_i$, if $i=\bar{r}(a)$, then in (63) there are only two terms related to i:

$$\bar{f}_{\bar{r}(a-1)\bar{r}(a)}(\boldsymbol{z})\text{ and }\bar{f}_{\bar{r}(a)\bar{r}(a+1)}(\boldsymbol{z}).$$

Define Q as in (61). Then,

$$\begin{aligned}
\frac{\partial f/\partial z_i}{f}=&\sum_{q:q\in Q,r(q)<r(i)}\frac{\partial f_{qi}/\partial z_i}{f_{qi}}+\\
&\sum_{q:q\in Q,r(q)>r(i)}\frac{\partial f_{iq}/\partial z_i}{f_{iq}}+\frac{1}{2}\sum_{q:q\in Q,r(q)=r(i),q\neq i}\left(\frac{\partial f_{qi}/\partial z_i}{f_{qi}}+\frac{\partial f_{iq}/\partial z_i}{f_{iq}}\right).
\end{aligned}$$

Next,

$$\begin{aligned}
\frac{\partial}{\partial z_i}\left(\frac{\partial f/\partial z_i}{f}\right)=&\sum_{q:q\in Q,r(q)<r(i)}\frac{\partial}{\partial z_i}\left(\frac{\partial f_{qi}/\partial z_i}{f_{qi}}\right)+\sum_{q:q\in Q,r(q)>r(i)}\frac{\partial}{\partial z_i}\left(\frac{\partial f_{iq}/\partial z_i}{f_{iq}}\right)\\
&+\frac{1}{2}\sum_{q:q\in Q,r(q)=r(i),q\neq i}\left(\frac{\partial}{\partial z_i}\left(\frac{\partial f_{qi}/\partial z_i}{f_{qi}}\right)+\frac{\partial}{\partial z_i}\left(\frac{\partial f_{iq}/\partial z_i}{f_{iq}}\right)\right).
\end{aligned}$$

These two results are almost the same as (53) and (59) used for the full-pair case. Hence δ_q and η_q are calculated by the same way as in Algorithm 1, but for Ω_i and Δ_i, instead of taking the sum over all $q=1,\ldots,k;q\neq i$, in (62) we sum up only elements in the set Q.

Appendix E. Derivations of Update Rules for the Thurstone-Mosteller Model

Define

$$f_{iq}(\boldsymbol{z})\equiv P(\text{team } i \text{ beats team } q)=\Phi\left(\frac{\theta_i-\theta_q-\epsilon}{c_{iq}}\right)$$

and

$$
\begin{aligned}
\bar{f}_{iq}(\boldsymbol{z}) \equiv & P(\text{team } i \text{ draws with team } q) \\
= & \Phi\left(\frac{\epsilon-(\theta_i-\theta_q)}{c_{iq}}\right) - \Phi\left(\frac{-\epsilon-(\theta_i-\theta_q)}{c_{iq}}\right),
\end{aligned}
$$

where $\theta_i = \sigma_i z_i + \mu_i$. Then

$$
P(\text{outcome of team } i \text{ and } q) = \begin{cases} f_{iq}(\boldsymbol{z}) & \text{if } r(i) > r(q), \\ f_{qi}(\boldsymbol{z}) & \text{if } r(i) < r(q), \\ \bar{f}_{iq}(\boldsymbol{z}) & \text{if } r(i) = r(q). \end{cases}
$$

Similar to the derivation for the Bradley-Terry model in (52) and (53),

$$
\frac{\partial f/\partial z_i}{f} = \sum_{q:r(q)<r(i)} \frac{\partial f_{qi}/\partial z_i}{f_{qi}} + \sum_{q:r(q)>r(i)} \frac{\partial f_{iq}/\partial z_i}{f_{iq}} + \sum_{q:r(q)=r(i)} \frac{\partial \bar{f}_{iq}/\partial z_i}{\bar{f}_{iq}}.
$$

Using the relation between ϕ and Φ in (66),

$$
\frac{\partial}{\partial \theta_i}\Phi\left(\frac{\theta_i-\theta_q-\epsilon}{c_{iq}}\right) = \phi\left(\frac{\theta_i-\theta_q-\epsilon}{c_{iq}}\right)\frac{1}{c_{iq}}. \tag{92}
$$

Therefore,

$$
\frac{\partial f_{iq}/\partial z_i}{f_{iq}} = \frac{1}{c_{iq}} \frac{\phi(\frac{\theta_i-\theta_q-\epsilon}{c_{iq}})}{\Phi(\frac{\theta_i-\theta_q-\epsilon}{c_{iq}})} \cdot \frac{\partial \theta_i}{\partial z_i} = \frac{\sigma_i}{c_{iq}} V\left(\frac{\theta_i-\theta_q}{c_{iq}}, \frac{\epsilon}{c_{iq}}\right), \tag{93}
$$

where the function V is defined in (67). Similarly,

$$
\frac{\partial f_{qi}}{\partial z_i} = \frac{-\sigma_i}{c_{iq}} V\left(\frac{\theta_q-\theta_i}{c_{iq}}, \frac{\epsilon}{c_{iq}}\right).
$$

For $\bar{f}_{iq}(\theta)$,

$$
\frac{\partial \bar{f}_{iq}}{\partial z_i} = \frac{-\sigma_i}{c_{iq}}\left(\phi(\frac{\epsilon-(\theta_i-\theta_q)}{c_{iq}}) - \phi(\frac{-\epsilon-(\theta_i-\theta_q)}{c_{iq}})\right),
$$

so

$$
\frac{\partial \bar{f}_{iq}/\partial z_i}{\bar{f}_{iq}} = \frac{-\sigma_i}{c_{iq}} \frac{\phi(\frac{\epsilon-(\theta_i-\theta_q)}{c_{iq}}) - \phi(\frac{-\epsilon-(\theta_i-\theta_q)}{c_{iq}})}{\Phi(\frac{\epsilon-(\theta_i-\theta_q)}{c_{iq}}) - \Phi(\frac{-\epsilon-(\theta_i-\theta_q)}{c_{iq}})} = \frac{\sigma_i}{c_{iq}} \tilde{V}\left(\frac{\theta_i-\theta_q}{c_{iq}}, \frac{\epsilon}{c_{iq}}\right), \tag{94}
$$

where the function $\tilde{V}$ is defined in (68). Then the update rule is

$$
\begin{aligned}
\mu_i \leftarrow & \mu_i + \sigma_i \left.\frac{\partial f(\boldsymbol{z})/\partial z_i}{f}\right|_{\boldsymbol{z}=\boldsymbol{0}} \\
\leftarrow & \mu_i + \sigma_i^2 \Bigg(\sum_{q:r(q)<r(i)} \frac{-1}{c_{iq}} V\left(\frac{\mu_q-\mu_i}{c_{iq}}, \frac{\epsilon}{c_{iq}}\right) + \sum_{q:r(q)>r(i)} \frac{1}{c_{iq}} V\left(\frac{\mu_i-\mu_q}{c_{iq}}, \frac{\epsilon}{c_{iq}}\right) \\
& + \sum_{q:r(q)=r(i), q\neq i} \frac{1}{c_{iq}} \tilde{V}\left(\frac{\mu_i-\mu_q}{c_{iq}}, \frac{\epsilon}{c_{iq}}\right) \Bigg).
\end{aligned}
$$

To update σ, similar to (59), we have

$$\frac{\partial}{\partial z_i}\left(\frac{\partial f/\partial z_i}{f}\right) = \sum_{q:r(q)<r(i)} \frac{\partial}{\partial z_i}\left(\frac{\partial f_{qi}/\partial z_i}{f_{qi}}\right) + \sum_{q:r(q)>r(i)} \frac{\partial}{\partial z_i}\left(\frac{\partial f_{iq}/\partial z_i}{f_{iq}}\right)$$
$$+ \sum_{q:r(q)=r(i),q\neq i} \frac{\partial}{\partial z_i}\left(\frac{\partial \bar{f}_{iq}/\partial z_i}{\bar{f}_{iq}}\right).$$

Using (93) and the fact that $d\phi(x)/dx = -x\phi(x)$

$$\begin{aligned}
&\frac{\partial}{\partial z_i}\frac{\partial f_{iq}/\partial z_i}{f_{iq}} \\
=&\frac{\sigma_i}{c_{iq}}\frac{\partial(\phi/\Phi)}{\partial \theta_i}\cdot\frac{\partial \theta_i}{\partial z_i} = \frac{\sigma_i^2}{c_{iq}}\frac{\Phi\frac{d\phi}{d\theta_i} - \phi\frac{d\Phi}{d\theta_i}}{\Phi^2} \\
=&\frac{\sigma_i^2}{c_{iq}}\left(-\left(\frac{\theta_i-\theta_q-\epsilon}{c_{iq}}\right)\cdot V\left(\frac{\theta_i-\theta_q}{c_{iq}},\frac{\epsilon}{c_{iq}}\right)\frac{1}{c_{iq}} - \frac{1}{c_{iq}}V\left(\frac{\theta_i-\theta_q}{c_{iq}},\frac{\epsilon}{c_{iq}}\right)^2\right) \qquad (95) \\
=&-\frac{\sigma_i^2}{c_{iq}^2}W\left(\frac{\theta_i-\theta_q}{c_{iq}},\frac{\epsilon}{c_{iq}}\right),
\end{aligned}$$

where the function W is defined in (67). Similarly,

$$\frac{\partial}{\partial z_i}\frac{\partial f_{qi}/\partial z_i}{f_{qi}} = -\frac{\sigma_i^2}{c_{iq}}W\left(\frac{\theta_q-\theta_i}{c_{iq}},\frac{\epsilon}{c_{iq}}\right). \qquad (96)$$

If $r(i) = r(q)$, then we use (92) and (94) to calculate

$$\frac{\partial}{\partial z_i}\left(\frac{\partial \bar{f}_{iq}/\partial z_i}{\bar{f}_{iq}}\right) = \frac{-\sigma_i^2}{c_{iq}}\frac{A-B}{\left(\Phi(\frac{\epsilon-(\theta_i-\theta_q)}{c_{iq}}) - \Phi(\frac{-\epsilon-(\theta_i-\theta_q)}{c_{iq}})\right)^2},$$

where

$$\begin{aligned}
A =&\frac{1}{c_{iq}}\left(\Phi(\frac{\epsilon-(\theta_i-\theta_q)}{c_{iq}}) - \Phi(\frac{-\epsilon-(\theta_i-\theta_q)}{c_{iq}})\right)\times \\
&\left(\frac{\epsilon-(\theta_i-\theta_q)}{c_{iq}}\phi(\frac{\epsilon-(\theta_i-\theta_q)}{c_{iq}}) - \frac{-\epsilon-(\theta_i-\theta_q)}{c_{iq}}\phi(\frac{-\epsilon-(\theta_i-\theta_q)}{c_{iq}})\right)
\end{aligned}$$

and

$$B = \frac{-1}{c_{iq}}\left(\phi(\frac{\epsilon-(\theta_i-\theta_q)}{c_{iq}}) - \phi(\frac{-\epsilon-(\theta_i-\theta_q)}{c_{iq}})\right)^2.$$

Hence

$$\begin{aligned}
&\frac{\partial}{\partial z_i}\left(\frac{\partial \bar{f}_{iq}/\partial z_i}{\bar{f}_{iq}}\right) \\
=&\frac{-\sigma_i^2}{c_{iq}^2}\left(\frac{\frac{\epsilon-(\theta_i-\theta_q)}{c_{iq}}\phi(\frac{\epsilon-(\theta_i-\theta_q)}{c_{iq}}) - \frac{-\epsilon-(\theta_i-\theta_q)}{c_{iq}}\phi(\frac{-\epsilon-(\theta_i-\theta_q)}{c_{iq}})}{\Phi(\frac{\epsilon-(\theta_i-\theta_q)}{c_{iq}}) - \Phi(\frac{-\epsilon-(\theta_i-\theta_q)}{c_{iq}})} + \tilde{V}\left(\frac{\theta_i-\theta_q}{c_{iq}},\frac{\epsilon}{c_{iq}}\right)^2\right) \qquad (97) \\
=&\frac{-\sigma_i^2}{c_{iq}^2}\tilde{W}\left(\frac{\theta_i-\theta_q}{c_{iq}},\frac{\epsilon}{c_{iq}}\right),
\end{aligned}$$

where the function $\tilde{W}$ is defined in (69). Combining (95), (96), and (97), the update rule for σ_i^2 is

$$\sigma_i^2 \leftarrow \sigma_i^2 \Bigg(1 - \Bigg(\sum_{q:r(q)<r(i)} \frac{\sigma_i^2}{c_{iq}^2} W(\frac{\mu_q - \mu_i}{c_{iq}}, \frac{\epsilon}{c_{iq}}) + \sum_{q:r(q)>r(i)} \frac{\sigma_i^2}{c_{iq}^2} W(\frac{\mu_i - \mu_q}{c_{iq}}, \frac{\epsilon}{c_{iq}}) \\ \sum_{q:r(q)=r(i), q\neq i} \frac{\sigma_i^2}{c_{iq}^2} \tilde{W}(\frac{\mu_i - \mu_q}{c_{iq}}, \frac{\epsilon}{c_{iq}}) \Bigg)\Bigg).$$

Appendix F. Derivations of Update Rules for the Plackett-Luce Model

Using $f(\boldsymbol{z})$ and $f_q(\boldsymbol{z})$ defined in (72),

$$f_q(\boldsymbol{z}) = \left(\frac{e^{\theta_q/c}}{\sum_{s\in C_q} e^{\theta_s/c}}, \right)^{1/A_q},$$

so

$$\begin{aligned}\frac{\partial f_q/\partial z_i}{f_q} &= \frac{\partial \log f_q}{\partial z_i} = \frac{1}{A_q} \left(\frac{\partial(\theta_q/c)}{\partial\theta_i} - \frac{\partial \log(\sum_{s\in C_q} e^{\theta_s/c})}{\partial\theta_i} \right) \frac{\partial\theta_i}{\partial z_i} \\ &= \frac{\sigma_i}{cA_q} \begin{cases} 1 - \frac{e^{\theta_i/c}}{\sum_{s\in C_q} e^{\theta_s/c}} & \text{if } q = i, \\ -\frac{e^{\theta_i/c}}{\sum_{s\in C_q} e^{\theta_s/c}} & \text{if } r(q) \le r(i), q \neq i, \\ 0 & \text{if } r(q) > r(i). \end{cases}\end{aligned} \tag{98}$$

From (38), the update rule is

$$\mu_i \leftarrow \mu_i + \Omega_i,$$

where

$$\begin{aligned}\Omega_i &= \sigma_i \sum_{q=1}^{k} \frac{\partial f_q(\boldsymbol{z})/\partial z_i}{f_q(\boldsymbol{z})} \Bigg|_{\boldsymbol{z}=\boldsymbol{0}} \\ &= \frac{\sigma_i^2}{c} \left(\frac{1}{A_i} \left(1 - \frac{e^{\mu_i/c}}{\sum_{s\in C_i} e^{\mu_s/c}} \right) + \sum_{q:q\neq i, r(q)\le r(i)} -\frac{1}{A_q} \frac{e^{\mu_i/c}}{\sum_{s\in C_q} e^{\mu_s/c}} \right).\end{aligned}$$

To update σ, similar to (59), we must calculate

$$\frac{\partial}{\partial z_i} \left(\frac{\partial f_q/\partial z_i}{f_q} \right), \forall q. \tag{99}$$

From (98), if $i \in C_q$, then

$$\begin{aligned}(99) = &-\frac{\sigma_i}{cA_q} \left(\frac{\partial}{\partial\theta_i} \frac{e^{\theta_i/c}}{\sum_{s\in C_q} e^{\theta_s/c}} \right) \cdot \frac{\partial\theta_i}{\partial z_i} = \frac{\sigma_i^2}{c^2 A_q} \frac{(\sum_{s\in C_q} e^{\theta_s/c}) e^{\theta_i/c} - (e^{\theta_i/c})^2}{(\sum_{s\in C_q} e^{\theta_s/c})^2} \\ = &\frac{\sigma_i^2}{c^2 A_q} \frac{e^{\theta_i/c}}{\sum_{s\in C_q} e^{\theta_s/c}} \left(1 - \frac{e^{\theta_i/c}}{\sum_{s\in C_q} e^{\theta_s/c}} \right).\end{aligned}$$

The update rule for σ_i^2 is

$$\sigma_i^2 \leftarrow \sigma_i^2 \left(1 - \sum_{q: r(q) \le r(i)} \frac{1}{c^2 A_q} \frac{e^{\mu_i/c}}{\sum_{s \in C_q} e^{\mu_s/c}} \left(1 - \frac{e^{\mu_i/c}}{\sum_{s \in C_q} e^{\mu_s/c}} \right) \right).$$

References

John Aitchison and Colin B. Begg. Statistical diagnosis when basic cases are not classified with certainty. *Biometrika*, 63:1–12, 1976.

Ralph A. Bradley and Milton E. Terry. The rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324–345, 1952.

Arpad E. Elo. *The Rating of Chessplayers, Past and Present*. Arco Publishing, New York, 2nd edition, 1986.

Mark E. Glickman. *Paired comparison models with time-varying parameters*. PhD thesis, Department of Statistics, Harvard University, 1993.

Mark E. Glickman. Parameter estimation in large dynamic paired comparison experiments. *Applied Statistics*, 48:377–394, 1999.

John Guiver and Edward Snelson. Bayesian inference for Plackett-Luce ranking model. In *Proceedings of the Twenty Sixth International Conference on Machine Learning (ICML)*, 2009.

Ralf Herbrich, Tom Minka, and Thore Graepel. TrueSkill$^{\text{TM}}$: A Bayesian skill rating system. In *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

Jim C. Huang and Brendan J. Frey. Cumulative distribution networks and the derivative-sum-product algorithm. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2008.

Tzu-Kuo Huang, Ruby C. Weng, and Chih-Jen Lin. Generalized Bradley-Terry models and multi-class probability estimates. *Journal of Machine Learning Research*, 7:85–115, 2006. URL `http://www.csie.ntu.edu.tw/~cjlin/papers/generalBT.pdf`.

Willard D. James and Charles M. Stein. Estimation with quadratic loss. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 361–379, 1961.

Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

John I. Marden. *Analyzing and Modeling Rank Data*. Chapman & Hall, London, 1995.

Joshua E. Menke and Tony R. Martinez. A Bradley-Terry artificial neural network model for individual ratings in group competitions. *Neural Computing and Applications*, 17: 175–186, 2008.

Thomas Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, MIT, 2001.

Charles M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 9:1135–1151, 1981.

Louis L. Thurstone. A law of comparative judgement. *Psychological Reviews*, 34:273–286, 1927.

Luke Tierney and Joseph B. Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81:82–86, 1986.

Ruby C. Weng. A Bayesian Edgeworth expansion by Stein's Identity. *Bayesian Analysis*, 5 (4):741–764, 2010.

Ruby C. Weng and Michael Woodroofe. Integrable expansions for posterior distributions for multiparameter exponential families with applications to sequential confidence levels. *Statistica Sinica*, 10:693–713, 2000.

Michael Woodroofe. Very weak expansions for sequentially designed experiments: linear models. *The Annals of Statistics*, 17:1087–1102, 1989.

Michael Woodroofe and D. Stephen Coad. Corrected confidence sets for sequentially designed experiments. *Statistica Sinica*, 7:53–74, 1997.

Ernst Zermelo. Die Berechnung der Turnier-Ergebnisse als ein Maximumproblem der Wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29:436–460, 1929.