EXTENSIONS OF THE TRUESKILL™ RATING SYSTEM

SERGEY I. NIKOLENKO AND ALEXANDER V. SIROTKIN

Abstract. The TrueSkill™ Bayesian rating system, developed a few years ago in Microsoft Research, provides an accurate probabilistic model for estimating relative skills of participants in the most general situation of participants re-organizing into different teams for each game. However, in cases when data on each participant is scarce, the teams may be of different size and their strength does not grow proportional to the size the TrueSkill™ system does not cope so well. We present several extensions and ramifications of the TrueSkill™ system and compare their predictive power on a testbed that exhibits all the problems described above.

Keywords. Probabilistic rating models, Bayesian inference.

1. Introduction. Probabilistic rating models

A Bayesian rating system is a probabilistic model that aims to infer a linear ordering (usually by providing a set of real numbers called *ratings*) on a certain set of entities (players) based on a set of noisy comparisons of small subsets of these entities (games). Naturally, any such model adopts certain assumptions on the base events (the comparisons) and provides a probabilistic description of the process, for which a maximal likelihood hypothesis on the ratings of the entities is finally inferred. While it is easier to think of probabilistic rating models in terms of sports, players, and games, they find other applications in areas where results of pairwise comparisons (or comparisons of a small number of entities) are available, and with this data one has to infer a general ordering [1–6].

The simplest example of a Bayesian rating system is the Elo rating system developed by Arpad Elo for comparing the skills of chessplayers [7]. The system makes very restrictive assumptions (for example, the Elo rating fixes the variance as a global constant and does not attempt to infer it from the data) and makes full use of the restrictive features of chess (e.g., strictly two player games). Nevertheless, it has been widely accepted, and the Elo rating and its close variations are widely used in chess and other sports with the same properties. The TrueSkill™ model is in fact a generalization of the Elo model, so we will not describe the latter in detail.

Another family of probabilistic rating models are the so-called Bradley–Terry models [8–10]. In their simplest form, Bradley–Terry models assume that each player i has a real rating γ_i, and the win probability of a player in a game is proportional to his rating γ_i ($\frac{\gamma_1}{\gamma_1+\gamma_2}$ and $\frac{\gamma_2}{\gamma_1+\gamma_2}$ in the simplest case of two players competing with no ties). There are natural generalizations of this model that incorporate ties, home advantage, and multiplayer contests, although the latter have limited support since a Bradley–Terry model grows exponentially

Research of the first author was partially supported by the Russian Presidential Grant Programme for Young Ph.D.'s, grant no. MK-4089.2010.1, for Leading Scientific Schools, grant no. NSh-5282.2010.1, Federal Target Programme "Scientific and scientific-pedagogical personnel of the innovative Russia" contracts no. 02.740.11.5192 and P265, and RFBR grants no. 09-01-12137-ofi_m, 09-01-00784-a, and 08-01-00640-a.

in the number of players (it enumerates transpositions). After a model is constructed, the maximal likelihood estimate of the ratings is found with likelihood maximization algorithms, e.g., minorization–maximization algorithms [11, 12] or neural networks [13].

The TrueSkill™ rating system [14] has been developed as a general probabilistic model that supports all possible situations in a multiplayer contest. It supports teams of individual players that compete in varying rosters; the system supports ties and teams of different size, and the model grows polynomially in the number of players. The inference algorithm is derived from standard Bayesian message passing algorithms [15, 16], and the authors of TrueSkill™ present an iterative approximate algorithm to make all involved distributions normal.

The TrueSkill™ model was designed to be used in online gaming, on the Xbox360 Live servers; later studies showed that TrueSkill™ has better predictive power than the classical Elo rating for chess games [17], and most recent results apply the ideas of TrueSkill™ to online learning problems [5, 6]. The present paper's ideas originate from an attempt to implement the TrueSkill™ system in a slightly different environment. It turned out that our dataset exhibits properties that make the basic TrueSkill™ system hard to use. We present modifications and extensions that can increase TrueSkill™'s predictive power in applications with the same characteristic features as ours.

The paper is organized as follows. Section 2 recounts the structure of the TrueSkill™ model. Section 3 explains the features of our dataset and why they make TrueSkill™ undesirable. In Section 4, we present our most important modification to the TrueSkill™ model, learning the places ratings, and list other modifications. Section 5 presents experimental results.

2. The TrueSkill™ rating model

The TrueSkill™ rating system fits the probabilistic model for skills of players who unite in teams of different size and participate in matches (tournaments) with several participants. The mathematical problem is to recompute posterior ratings (skill estimates) after each tournament.

The model does not assume to know actual prior skills, but rather a certain prior distribution (assumed to be normal) $f(s_i) = \mathcal{N}(s; \mu_i, \sigma_i)$. Here μ_i is the actual skill of player i, and σ_i is the variance that characterizes how accurate the estimate is. After each match, the variance decreases (if the model does not artificially increase it).

Each "true" skill is a mean value around which the actual performance shown by a player in a given match is distributed, $f(p_i) = \mathcal{N}(p_i; s_i, \beta^2)$. The TrueSkill™ system assumes that β^2 is a universal constant common for all players[1].

It is easy to express p_i via initial parameters by integrating over all possible s_i:

$$f(p_i \mid \mu_i, \sigma_i) = \int_{-\infty}^{\infty} \mathcal{N}(p_i; s_i, \beta^2)\mathcal{N}(s_i; \mu_i, \sigma_i)ds_i.$$

Then player performances are combined to yield team performances. The TrueSkill™ system uses a simple assumption: team performance is the sum of performances of its players: $t_i = \sum_i p_i$, or, in a functional form, $f(t_i) = \mathbb{I}(t_i - \sum_i p_i)$.

[1]Compare to the "skill level" constant of 200 points in the Elo rating.

After that, team performances in a tournament must be compared; their comparison should generate the order actually given by tournament results. Some teams can finish in a tie; in this case, the TrueSkill™ system introduces a new global constant, ϵ, and assumes that a draw between teams with performances t_1 and t_2 means that $|t_1 - t_2| < \epsilon$.

The problem is to compute the posterior ratings; the data is a permutation of the teams π that reflects match results (in which neighboring teams may finish in a draw). In other words, we are to compute

$$p(\boldsymbol{s} \mid \pi) = \frac{p(\pi \mid \boldsymbol{s})p(\boldsymbol{s})}{\int p(\pi \mid \boldsymbol{s})p(\boldsymbol{s})d\boldsymbol{s}}.$$

Apart from s_i and π, variables p_i, t_i, and d_i are introduced, and the joint distribution density of the entire system is presented as a product of distributions

$$p(\pi, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{p}, \boldsymbol{s}) = p(\pi \mid \boldsymbol{d})p(\boldsymbol{d} \mid \boldsymbol{t})p(\boldsymbol{t} \mid \boldsymbol{p})p(\boldsymbol{p} \mid \boldsymbol{s})p(\boldsymbol{s}).$$

The problem is to compute

$$p(\pi \mid \boldsymbol{s}) = \int\int\int p(\pi, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{p}, \boldsymbol{s})d\boldsymbol{d}d\boldsymbol{t}d\boldsymbol{p}.$$

Consequently, we are facing a marginalization problem which is one of the basic subject of Bayesian analysis theory [15, 16]. To solve the problem, the TrueSkill™ system employs a well-known marginalization message passing algorithm. See Fig. 1 for sample factor graphs, including simple but representative cases of a match between two players, a match between two teams of two players, and a match of four players with a draw.

The only trick is the approximate message passing in the bottom part of the graph. All distributions are normal except for the distributions generated by the bottom nodes (e.g., the $\mathbb{I}(d_1 > \epsilon)$ node on Fig. 1c). Therefore, in TrueSkill™ this distribution is approximated with a normal distribution (by computing the first two moments), and the message passing algorithm goes on along the bottom part of the graph until convergence [14].

3. Dataset and problems encountered

In this section, we list the problems that can make the predictive power of classical TrueSkill™ suffer. The dataset we attempted to apply TrueSkill™ to are tournament histories for a Russian intellectual sport "What? Where? When?". The game is played by a (possibly large) number of teams who compete in answering specifically composed questions. Unfortunately, the number of correctly answered questions, on which we could base a different probabilistic model, is unavailable, and tournament results are presented as lists of places.

The game community shares all problems that TrueSkill™ is designed to overcome: it is a friendly non-professional activity so players often switch teams even in official tournaments, the teams cap at six players but are often understaffed, and so on. However, certain features of this dataset have presented problems for classical TrueSkill™. In this section, we list these problems and present our approaches to solving them.

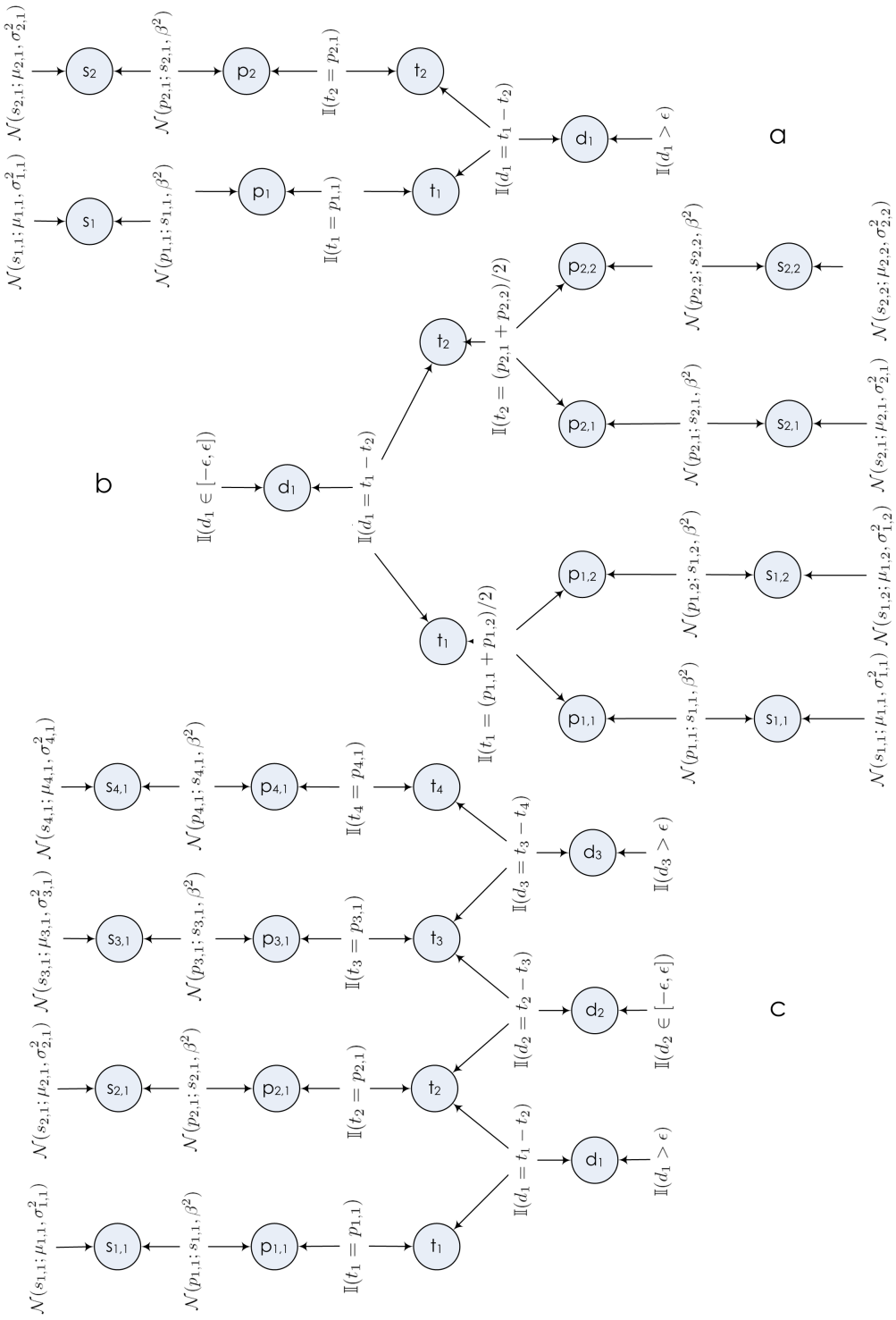

FIGURE 1. Sample TrueSkill™ factor graphs. *a* – a match of two players, the first has won; *b* – a match of two teams of two players each that has ended in a draw; *c* – a match of four players, in which the first won, second and third finished in a tie, and the fourth lost.

3.1. **Common multiway draws.** In many tournaments of our dataset, many teams often draw; there are large tournaments with limited number of different places (say, 30–40) and a large number of teams (up to several thousand).[2] The TrueSkill™ system behaves badly in this situation due to the way it handles ties. If several teams line up in a multiway draw, semantics of the $\mathbb{I}(|d_{i+1} - d_i| \le \epsilon)$ node cause incorrect behaviour, as we show on the following example. Suppose there are four teams in a tournament, 1 through 4, with performances $p_1, \dots, p_4$. Team 1 has won, while teams 2–4, listed in this order, drew behind the first. Then the structure of the factor graph imposes the following restrictions:

$$\begin{aligned} p_2 &< p_1 - \epsilon, \\ |p_2 - p_3| &\le \epsilon, \\ |p_3 - p_4| &\le \epsilon. \end{aligned}$$

Note that team 3's performance may actually nearly equal p_1, and p_4 may exceed p_1!

This problem is magnified when there are many teams, and the boundary cases are actually often realized as maximal likelihood hypotheses (say, in a situation when an *a priori* leader lost in a multiway draw). Thus, in a setting with common multiway draws a modification of the TrueSkill™ model is required; we describe the corresponding modification in Section 4.

3.2. **Draw constant ϵ.** Another problem related to the lack of attention to draws in basic TrueSkill™. Our dataset contains two distinct classes of tournaments. In one, multiway draws are common (see item 1). In the other, additional parameters are used, and draws are virtually impossible. Thus, using the same value of ϵ for both kinds of tournaments would either group the teams too close together in the first kind or spread them impossibly far apart in the second kind.

To alleviate this problem, we learn the value of ϵ automatically from tournament results. Several approaches to computing ϵ are possible, all based on the number of different places m in the tournament results. The simplest approach is a linear spread from the prior estimate of the best team's performance p_{best} to the prior estimate of the worst team's performance p_{worst}:

$$\epsilon = \frac{p_{\text{best}} - p_{\text{worst}}}{m}.$$

In our particular case, the dataset had a large gap between the two strictly different cases outlined above. Therefore, we introduced two values of ϵ, recognized which case we are dealing with, and applied the corresponding value. In other applications, more care may be needed in working with the ϵ constant.

3.3. **Variable team size.** The game is played in teams of six players, but teams are often incomplete. The expected performance of a five- or four-player team is not all that much worse than for a six-player team; in fact, if a relatively weak player leaves the team, it will lose hardly anything. Basic TrueSkill™ uses sums to represent team performance, which is unacceptable here: teams with fewer players will get an almost automatic rating boost.

This problem can be alleviated by using another function for team performance. The first idea is to use *average* performance instead of the sum of performances (all linear functions are

[2]In the problem setting, this results from the fact that competitions actually consist of solving several dozen problems, and the teams are ranked according to the number of correctly solved problems. Thus, if there are no additional parameters then large multiway draws are inevitable. This feature also applies to any other competition of the same discrete nature.

fully supported by basic TrueSkill™: a weighted sum of normal distributions is still normal) but discount it linearly for incomplete teams (the best value of the discount may vary from dataset to dataset). In Section 5, we denote an implementation of this idea by "incomplete teams discount".

The second idea is to use a team performance function weighted towards the team leaders. However, this may lead to a "rich get richer" situation, when leaders of their corresponding teams get larger rating bonuses, and the gap between players of the same team who often play together grows. Different leader bonus values are also compared in Section 5. We plan further experiments with other, nonlinear team performance functions. However, we believe that no single performance function will suit all problems, so we encourage other TrueSkill™ users to experiment on their own data and find which team performance function works best in their situation.

3.4. **Variances.** In the TrueSkill™ system, rating variances are estimated together with the means. This actually implies that variance always decreases over time. This also leads to a "rich get richer" problem, and it becomes very hard for a player with meek beginnings to achieve greatness. To alleviate this problem, authors of the TrueSkill™ system suggest to artificially increase the variance before (or after) each tournament. However, for tournaments with a wide range of sizes (from dozens to thousands of teams) it becomes hard to pick a unified constant that will suit all situations. Therefore, after experimenting with different strategies for increasing the variance, we have come to a conclusion that the best way to process variance would be to set it constant for all players (much like the Elo rating does).

4. Fitting place performances

In this section, we describe a modification to the TrueSkill™ system that we have implemented to cope with common multiway draws (see Section 3.1).

We introduce an additional entity to the base TrueSkill™ model: the layer of *place performances* l_i. Each place performance provides an estimate for the team performance it took to get to a given place in the final rankings of a tournament. The TrueSkill™ bottom level remains the same, but it is now connected to the place performances level, and each team is connected to its corresponding place via a node that requires it to score a performance in the ϵ-neighborhood of this node. Fig. 2 shown a sample factor graph for the new probabilistic model (let us call it TS2 for the moment) corresponding to the factor graph on Fig. 1c. Note how an additional layer of place performances corrects the errors in handling multiway draws shown in Section 3.1. After that, the first version of TS2 performs the usual Bayesian inference on the modified factor graph.

In the second version of the modified model, TS3, the factor graph stays the same but the inference algorithm changes. Basically, we break the inference up in two stages: the first stage computes maximal likelihood estimates for place performances, and the second stage takes them as point estimates and computes posterior player ratings. A (more) formal description of the algorithm follows. The algorithm receives as input prior ratings of all players of all participating teams and a table of places of all participating teams. Suppose that there are m different places, and team i placed $j(i)^{\text{th}}$ in the tournament.

(1) Compute prior estimates of team performances t_i.

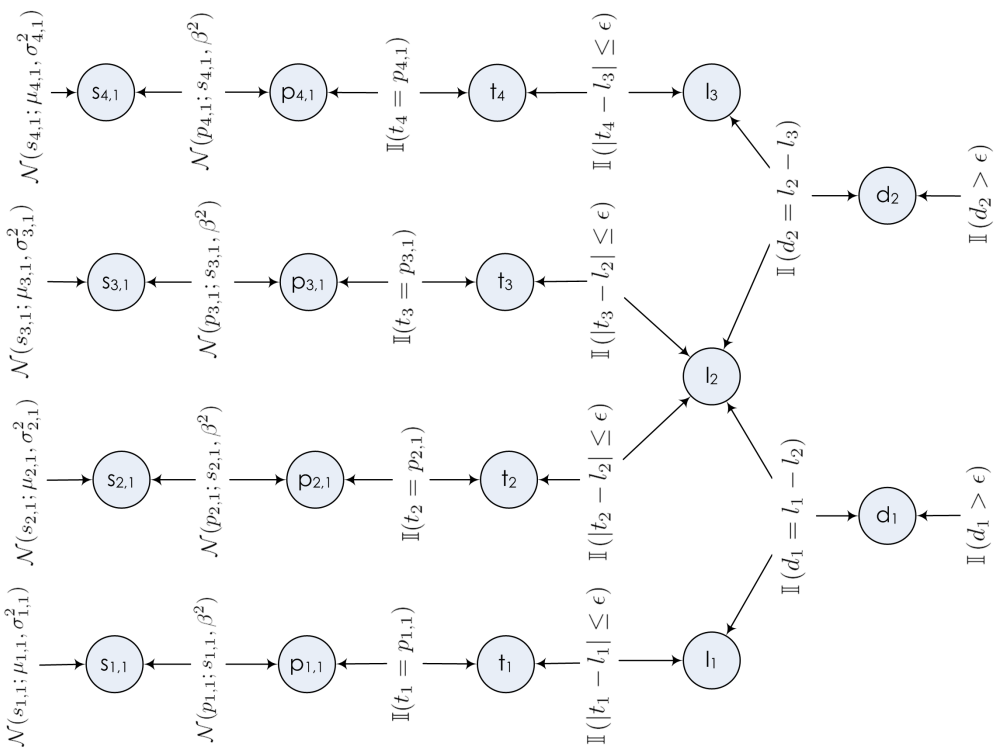

FIGURE 2. A sample TS2 factor graph corresponding to Fig. 1c. In this case, $j(1) = 1$, $j(2) = j(3) = 2$, $j(3) = 4$.

(2) Compute the joint likelihood of the fact that team i has shown the performance in an ϵ-neighborhood of an unknown performance value x_j, that is, $|t_i - x_{j(i)}| \leq \epsilon$. We get a large product of distributions, a function of all x_i's:

$$f(x_1, \ldots, x_m) = \prod_i p\left(|t_i - x_{j(i)}| \leq \epsilon\right).$$

(3) Maximize $f(x_1, \ldots, x_m)$ under the constraints

$$x_1 \geq x_2 + 2\epsilon \geq x_3 + 4\epsilon \geq \ldots \geq x_m + 2(m-1)\epsilon.$$

(4) Propagate the maximizing values of x_j to their corresponding teams, assuming that an event $|t_i - x_{j(i)}| \leq \epsilon$ has happened.
(5) Output the resulting posterior estimates.

This algorithm is incorrect in the sense that it is no longer guaranteed to produce the maximal likelihood hypothesis for the model shown on Fig. 2. However, our experiments have shown that TS3 actually outperforms TS2 in predictive power (see Section 5). We leave a probabilistic explanation of this model as subject for further study.

5. Experimental results

In this Section, we present some experimental result. In our experiments, various rating models try to learn the ratings of the players and predict the results of new . We consider six version of the prediction system's implementation.

(1) The basic TrueSkill™ system.

TABLE 1. Pairwise comparisons of the rating models: numbers of better predictions.

	1	2	3	4	5	6
1	0	40	40	43	49	45
2	110	0	70	75	72	63
3	109	75	0	74	76	63
4	107	69	53	0	72	63
5	100	76	72	76	0	45
6	104	85	83	83	96	0

(2) TS2 (with the additional layer for place performances).
(3) TS2 with incomplete teams support.
(4) TS2 with incomplete teams support and a 10% leader bonus.
(5) TS3 with incomplete teams support.
(6) TS3 with incomplete teams support and a 10% leader bonus.

We have used the following error prediction measure: a prediction system that predicted places y_i, $i = 1..n$, for the teams that finally placed x_i, $i = 1..n$, receives error prediction score (less is better)

$$\sqrt{\sum_{i=1}^{n} \frac{1}{\sqrt{i}} (x_i - y_i)^2}.$$

The $\frac{1}{\sqrt{i}}$ factor is a natural discount given to errors in low places. There are two reasons for this discount: first, we are naturally more interested in correct predictions of the leaders; second, as the place number grows, more and more teams are usually tied, so a small error in the actual result may cause wild changes in the absolute place ranking.

We have performed pairwise comparisons of the predictive power of each of six prediction systems. Our dataset consists of 449 tournaments, in which a total of more than 30000 players have participated. Before each tournament, a probabilistic rating model makes a prediction, and we compare whose prediction was better according to the score introduced above. Giving the models some time to learn, we only count the results from the last 150 tournaments. Table 1 shows the results of pairwise comparisons; cell (i, j) contains the number of tournaments in which model i has had better predictions than model j minus the number of tournaments (the numbers do not sum up to 150 because sometimes predictions coincide completely). Table 2 shows the same data in a more clear way: its cell (i, j) contains the advantage of rating model i over rating model j in our experiments.

Experiments clearly indicate that the basic TrueSkill$^{\text{TM}}$ system has lost to every modification of ours. Among the modifications, TS3 with incomplete teams support and a 10% leader bonus came out on top. Fig. 3 shows a more detailed comparison of the predictive power of these two rating models, showing their relative predictive error score on the last 150 tournaments.

TABLE 2. Pairwise comparisons of the rating models: advantages of one model over the other.

	1	2	3	4	5	6
1	0	-70	-69	-64	-51	-59
2	70	0	-5	6	-4	-22
3	69	5	0	21	4	-20
4	64	-6	-21	0	-4	-20
5	51	4	-4	4	0	-51
6	59	22	20	20	51	0

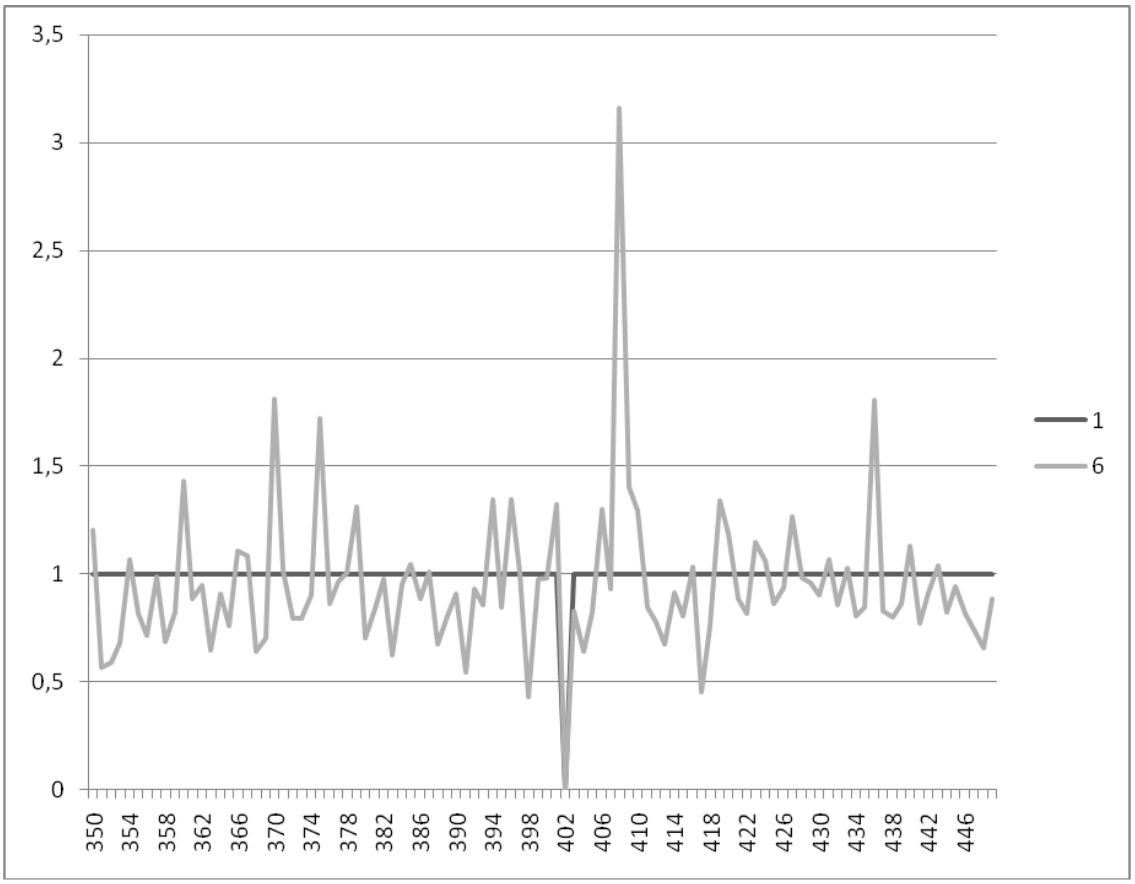

FIGURE 3. TS3 with incomplete teams support and a 10% leader bonus compared to basic TrueSkill™ system prediction error.

6. CONCLUSION

In this paper, we have presented several modification to the TrueSkill™ model that enhance the model on datasets that exhibit certain properties unfavourable for the original model. The most important of these modifications aims to alleviate the problem of multiway ties. We have introduced another level of place performances to the model and devised a new inference algorithm that makes use of this additional level. Experimental results show that our models outperform basic TrueSkill™.

Further work may include both theoretical and practical investigations. The most important theoretical question we are currently facing is to explain the probabilistic sense and describe the properties of our TS3 model. More practical questions include devising procedures for learning the ϵ parameter and the new parameters (incomplete team discount, leader bonus) we have introduced in our modifications.

References

[1] Marden, J.I.: Analyzing and Modeling Rank Data. London: Chapman and Hall (1995)

[2] Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. Journal of Machine Learning Research **5** (2004) 975–1005

[3] Coulom, R.: Computing Elo ratings of move patterns in the game of Go. ICGA Journal **30**(4) (December 2007) 198–208

[4] Stern, D., Herbrich, R., Graepel, T.: Bayesian pattern ranking for move prediction in the game of Go. In: Proceedings of the 23rd International Conference on Machine Learning. (2006)

[5] Zhang, X., Graepel, T., Herbrich, R.: Bayesian online learning for multi-label and multi-variate performance measures. In: Proceedings of the Thirteenth Conference on Artificial Intelligence and Statistics AISTATS 2010. (2010) to appear

[6] Graepel, T., Candela, J.Q., Borchert, T., Herbrich, R.: Web-scale bayesian click-through rate prediction for sponsored search advertising in microsofts bing search engine. In: Proceedings of the 27th International Conference on Machine Learning. (2010) to appear

[7] Elo, A.: The Ratings of Chess Players: Past and Present. New York: Arco (1978)

[8] Bradley, R.A., Terry, M.E.: Rank analysis of incomplete block designs. i. the method of paired comparisons. Biometrika **39** (1952) 324–245

[9] Rao, P.V., Kupper, L.L.: Ties in paired–comparison experiments: a generalization of the Bradley–Terry model. Journal of the American Statistical Association **62** (1967) 194–204

[10] Agresti, A.: Categorical Data Analysis. New York: Wiley (1990)

[11] Hunter, D.R.: MM algorithms for generalized Bradley-Terry models. The Annals of Statistics **32**(1) (2004) 384–406

[12] Huang, T.K., Weng, R.C., Lin, C.J.: Generalized Bradley–Terry models and multi-class probability estimates. Journal of Machine Learning Research **7** (2006) 85–115

[13] Menke, J.E., Martinez, T.R.: A Bradley–Terry artificial neural network model for individual ratings in group competitions. Neural Computing and Applications **17**(2) (2008) 175–186

[14] Graepel, T., Minka, T., Herbrich, R.: Trueskill(tm): A Bayesian skill rating system. In Schölkopf, B., Platt, J., Hoffman, T., eds.: Advances in Neural Information Processing Systems 19, Cambridge, MA, MIT Press (2007) 569–576

[15] MacKay, D.J.: Information Theory, Inference and Learning Algorithms. Cambridge University Press (2003)

[16] Tulupyev, A.V., Nikolenko, S.I., Sirotkin, A.V.: Bayesian networks: a logical probabilistic approach. St. Petersburg, Nauka (2006)

[17] Dangauthier, P., Graepel, T., Minka, T., Herbrich, R.: Trueskill through time: Revisiting the history of chess. In Platt, J., Koller, D., Singer, Y., Roweis, S., eds.: Advances in Neural Information Processing Systems 20, Cambridge, MA, MIT Press (2008) 337–344