A New Bayesian Rating System for Team Competitions

Sergey Nikolenko[1,2] Alexander Sirotkin[1,3]

[1]St. Petersburg Academic University
[2]Steklov Mathematical Institute, St. Petersburg
[3]St. Petersburg Institute for Informatics and Automation of the RAS

ICML 2011, June 30, 2011

Outline

Introduction

- In probabilistic rating models, Bayesian inference aims to find a linear ordering on a certain set given noisy comparisons of relatively small subsets of this set.
- Useful whenever there is no way to compare a large number of entities directly, but only partial (noisy) comparisons are available.
- We will stick to the metaphor of matches and players.

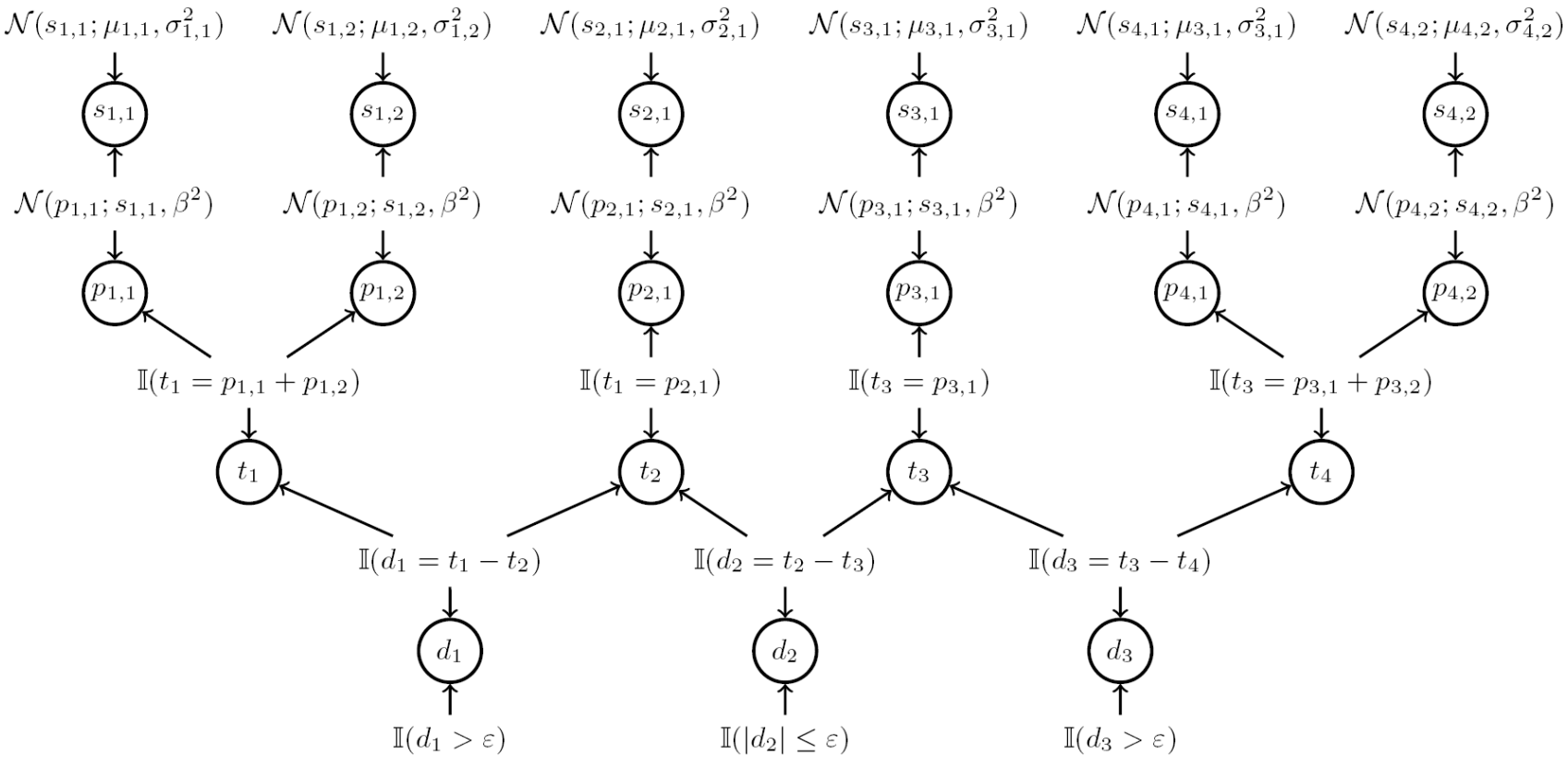
Introduction

- Elo rating system: first probabilistic rating model (chess: two players).
- Bradley–Terry models: assume that each player has a "true" rating γ_i, and the win probability is proportional to this rating: γ_1 wins over γ_2 with probability $\frac{\gamma_1}{\gamma_1+\gamma_2}$.
- Inference: fit this model to the data from matches played.
- Several extensions, but large matches are hard for Bradley–Terry models.

Introduction

- TrueSkill was initially developed in MS Research for Xbox Live gaming servers [Graepel, Minka, Herbrich, 2007].
- Given results of team competitions, learn the ratings of players of these teams.
- Direct application – matchmaking: find interesting opponents for a player or team.
- [Graepel et al., 2010]: AdPredictor. Predicts CTRs of advertisements based on a set of features: the features are a team, and the team wins whenever a user clicks the ad.

TrueSkill factor graph



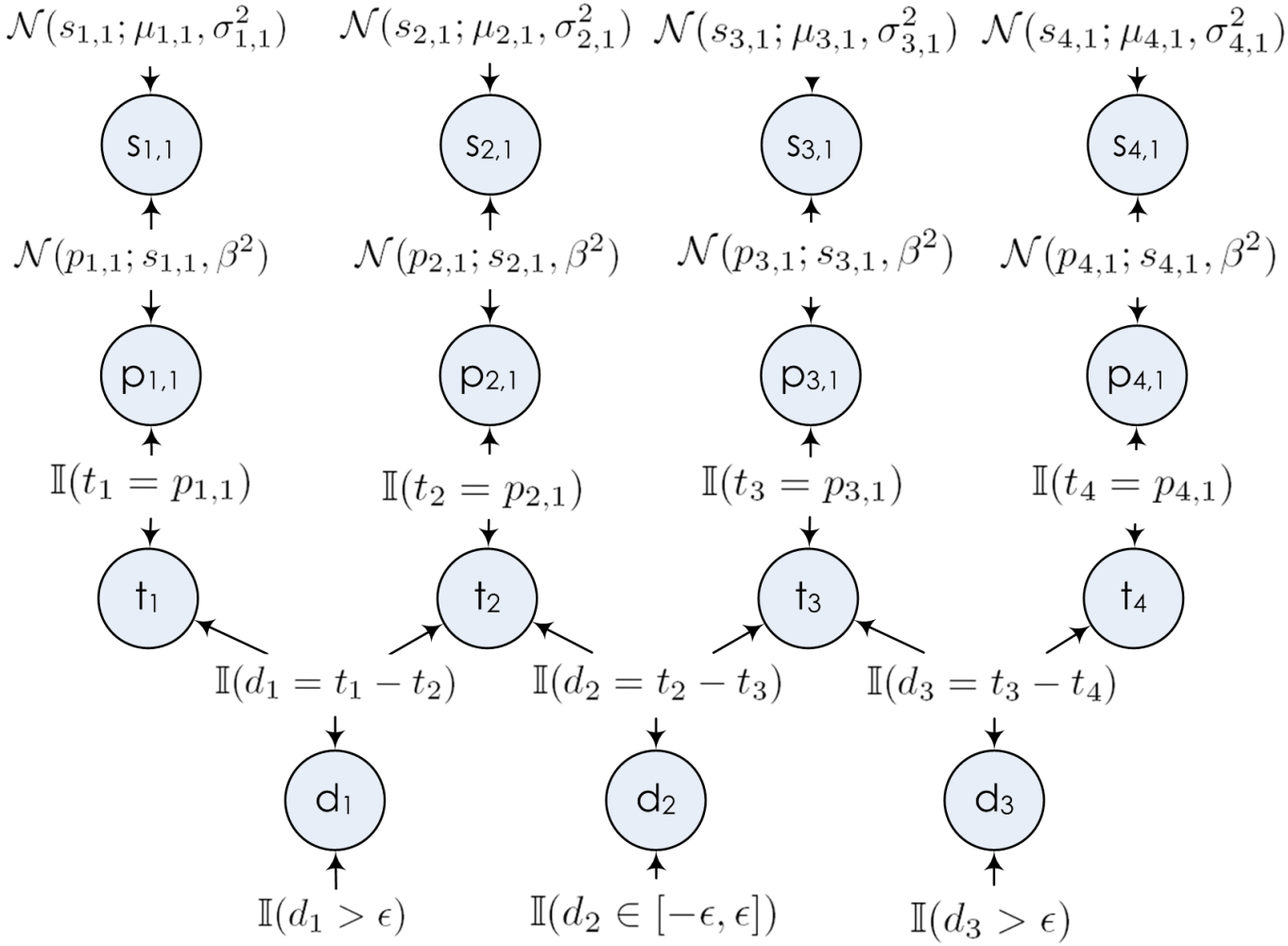
TrueSkill

- Layers of TrueSkill factor graph:
  - $s_{i,j}$ – skill of player i from team j; normally distributed around $\mu_{i,j}$ with variance $\sigma_{i,j}$, where $(\mu_{i,j}, \sigma_{i,j})$ are prior ratings;
  - $p_{i,j}$ – performance of player i from team j in this match; conditionally normally distributed around the skill $s_{i,j}$ with variance β (a global model parameter);
  - t_j – performance of team j; in TrueSkill^{TM}, team performance is the sum of player performances;
  - d_j – difference in performance between teams who took neighboring places in the tournament; a tie between two teams corresponds to $|d_j| \leq \varepsilon$ for some model parameter ε, and a win corresponds to $d_j > \varepsilon$.

TrueSkill

- There is no evidence per se, it is incorporated in the structure of the graph, we just have to marginalize by message passing.
- The marginalization problem is complicated by the step functions at the bottom; solved with Expectation Propagation [Minka, 2001]:
  - approximate messages from $\mathbb{I}(d_i > \epsilon)$ and $\mathbb{I}(|d_i| \leq \epsilon)$ to d_i with normal distributions;
  - repeat message passing on the bottom layer of the graph until convergence.

Example: a match of four players



Motivation

- We started with a practical problem: we tried to apply TrueSkill^{TM} to a Russian game "What? Where? When?".
- Teams of ≤ 6 players answer questions, whoever gets the most correct answers wins.
- It turned out that TrueSkill^{TM} works poorly on this dataset because of its properties:
  - large multiway ties are common; it is common to have 30–40 different places (because there were 35-50 questions in total) in a tournament with a thousand teams;
  - teams vary in size (max 6 players, but often incomplete).
- Why is it bad for TrueSkill^{TM} and what do we do about it?

Outline

Variable team size

- An undesired feature of TrueSkill^{TM} is the assumption that a team's performance is the sum of player performances.
- In many competitions (and comparison problems), an undersized team stands a very good chance against a full one, and it would be an unfair boost for the smaller team.
- To alleviate the team performance formula problem, we simply select a different function.
- We can very easily use any affine function, e.g., average (but it would be unfair for smaller teams now).

Variable team size

- Moreover, there is a simple way to approximate nonlinear functions: replace player performances with their estimates provided by the prior ratings μ_i.
- For instance, to approximate a team performance function

$$t = p_1^2 + p_2^2 + \ldots + p_n^2$$

we replace it with

$$t = \mu_1 p_1 + \mu_2 p_2 + \ldots + \mu_n p_n$$

(here p_i are model variables, and μ_i are constants fixed before inference and equal to prior ratings).

Variable team size

- I don't know a universally good team performance function, I can only encourage you to try different ones.
- In the end, for our dataset the function that worked best was (assuming $m_{i,j}$'s are sorted)

$$t_i = \begin{cases} \dfrac{\sum_{j=1}^{n_i} p_{i,j}}{n_i} \cdot (0.88 + 0.02 n_i), & \text{if } n_i \leq 6, \\ \sum_{j=1}^{n_i} p_{i,j} \cdot \dfrac{\sum_{j=1}^{6} \mu_{i,j}}{6 \sum_{j=1}^{n_i} \mu_{i,j}}, & \text{if } n_i > 6. \end{cases}$$

  where n_i is number of players in team i.
- Obviously, it wouldn't work for other applications.

Multiway ties

- Large multiway ties are deadly for TrueSkill^{TM}. Consider four teams in a tournament with performances $p_1, \ldots, p_4$.
- Team 1 has won, while teams 2–4, listed in this order, drew behind the first.
- Then the factor graph tells us that

$$t_2 < t_1 - \epsilon, \quad |t_2 - t_3| \leq \epsilon, \quad |t_3 - t_4| \leq \epsilon.$$

- Team 3's performance t_3 may actually nearly equal t_1, and t_4 may exceed t_1!

Multiway ties

- Moreover, these boundary cases are realized in practice when unexpected results occur.

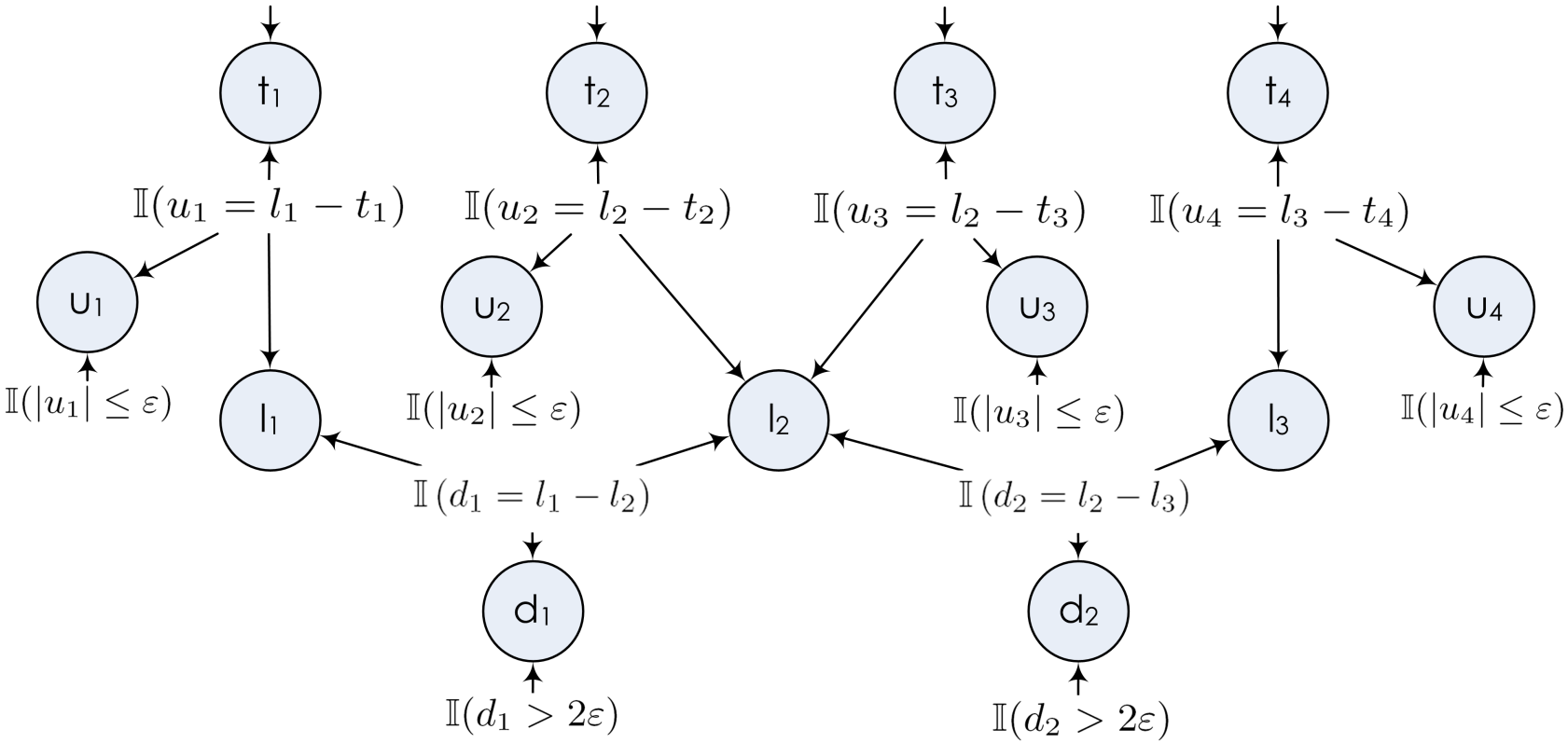
$$t_2 < t_1 - \epsilon, \quad |t_2 - t_3| \leq \epsilon, \quad |t_3 - t_4| \leq \epsilon.$$

- Suppose the winning team t_1 was an underdog, and its prior distribution fell behind the priors of t_2, t_3, and t_4, t_4 being the prior leader of all four.
- Then the maximum likelihood value of t_4 is likely to exceed t_1.

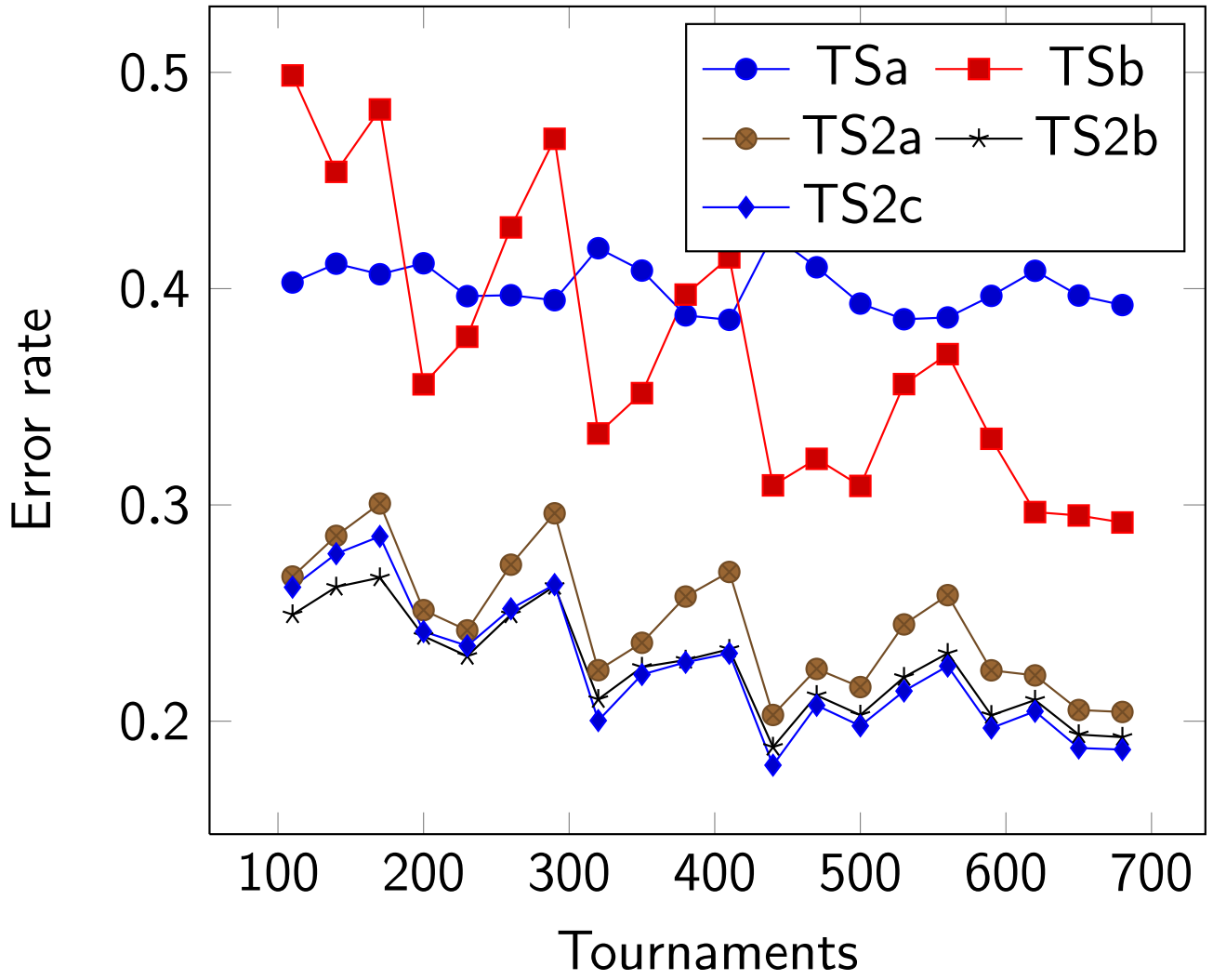
Changes in the factor graph

- For the multiway tie problem, we add another layer in the factor graph, namely the layer of *place performances* l_i.
- Each team performs in the ϵ-neighborhood of its place performance, and place performances relate to each other with strict inequalities like $l_2 < l_1 - 2\epsilon$.
- Then it's inference as usual. We have not experienced any slowdown in convergence.

New factor graph



Experimental results



Average error rate over a sliding window of 50 tournaments.

Thank you!

Thank you for your attention!