

С.И. НИКОЛЕНКО, Д.В. СЕРДЮК, А.В. СИРОТКИН
**БАЙЕСОВСКИЕ РЕЙТИНГ-СИСТЕМЫ С УЧЁТОМ
ДОПОЛНИТЕЛЬНОЙ ИНФОРМАЦИИ О
РЕЗУЛЬТАТАХ**

Николенко С.И., Сердюк Д.В., Сироткин А.В. **Байесовские рейтинг-системы с учётом дополнительной информации о результатах.**

Аннотация. Мы рассматриваем задачу построения байесовской рейтинг-системы, в которой из результатов отдельных матчей/турниров (упорядочиваний на небольшом множестве) обучается единая ранжирующая функция. Расширяя результаты [9], мы строим рейтинг-систему, которая может учитывать дополнительную численную информацию о результатах турниров. Статья содержит подробные описания моделей байесовских рейтинг-систем, алгоритмы вывода и результаты экспериментов.

Ключевые слова: байесовские рейтинг-системы, вероятностные графические модели, фактор-граф, байесовский вывод.

Nikolenko S.I., Serdyuk D.V., Sirotkin A.V. **Bayesian rating systems with additional information on tournament results.**

Abstract. We consider the problem of constructing a Bayesian rating system that learns a general ranking function from the results of individual matches and/or tournaments (rankings on small subsets). Extending the results of [9], we construct a rating system that can take into account additional numerical information on tournament results. The paper contains detailed descriptions of Bayesian rating models, inference algorithms, and experimental results.

Keywords: Bayesian rating systems, probabilistic graphical models, factor graph, Bayesian inference.

1. Введение. Байесовские рейтинг-системы изначально возникли в контексте спортивных соревнований. Для спорта всегда было важно понять, каков «истинный уровень» спортсменов или команд, к чему приводит совокупность результатов всевозможных соревнований, как сравнивать спортсменов, которые живут в разных частях света и лично, может быть, и вовсе никогда не встречались. Но это не единственный смысл рейтинг-системы в спорте; честная и истинная оценка силы игрока или команды приводит к тому, что участник рейтинга может легко найти себе соперника, матч с которым был бы максимально интересен и полезен (например, соперника максимально близкой силы). Эта проблема особенно остро встала в последние годы с развитием киберспорта и соревнований по традиционным и новым играм в интернете. На игровом сервере участники вряд ли знают друг друга, но не могут выбирать соперника вслепую, так как слишком велик шанс нарваться

на неравный поединок. Задача осложняется тем, что поединки часто происходят в командах, составы которых от матча к матчу меняются, а также тем, что результат (обновлённый рейтинг) нужно рассчитывать онлайн, сразу после поединка.

Рейтинг-системы, осуществляющие «честную» оценку силы игры, давно были предметом исследований; первой из широко известных таких систем стал рейтинг Эло, внедрённый в шахматном мире в пятидесятые годы. Однако не следует думать, что рейтинг-системы нужны только для развлечений. По своей математической сути рейтинг-система — это просто способ сравнить несколько элементов по тому или иному признаку, не имея возможности сравнивать всех сразу, но имея возможность проводить немного «зашумленные» сравнения попарно или в других небольших группах. Таким образом, модель Брэдли-Терри оказывается полезной, например, для задачи классификации с несколькими категориями: по данным бинарных классификаторов (сравнивающих две категории) определить вероятности попадания в каждую из категорий [4, 12]. Аналогичные задачи — по частичным сравнениям восстановить полную картину — возникают и во многих других приложениях, например, в обработке изображений [10] или сравнении паттернов при обучении [1]. Важным недавним применением стало предсказание популярности интернет-рекламы [2]; это применение было основано на недавно разработанной байесовской рейтинг-системе TrueSkill [3].

В байесовских рейтинг-системах (таких, как TrueSkill) рейтинг участника представляет собой параметрическое задание распределения вероятностей, — оценки его силы игры. Такие параметрические задания для каждого участника подаются на вход системы в качестве априорных распределений, после чего проводится байесовский вывод (он подробно описан в последующих разделах), а в качестве нового рейтинга участника принимается апостериорное распределение оценки силы игры. В TrueSkill входными данными являются упорядоченные результаты командных турниров (в которых допускаются ничьи, а составы команд известны). Однако в реальности может быть доступна и другая, дополнительная информация о результатах турниров; так, может быть известно, «с каким счётом» закончился матч, т.е. может быть доступна информация о том, какой разрыв в силе между участниками продемонстрировал данный конкретный турнир.

В этой работе мы продолжаем исследования, начатые первым и третьим авторами в [8, 9], и обобщаем разработанную ими рейтинговую систему на случай, когда результаты турниров содержат не только упорядочивание. В разделе 2 мы опишем исходную систему TrueSkill [3], в разделе 3 – систему TrueSkill2, разработанную в [9], а в разделе 4 – новую модификацию, использующую дополнительную информацию. Раздел 5 содержит результаты экспериментов, а раздел 6 завершает работу.

2. TrueSkill. Рейтинговая система TrueSkill была разработана в компании Microsoft для игровых серверов Xbox 360. Ключевые сложности при создании подобной рейтинговой системы заключаются в том, что постановка задачи теперь становится максимально общей. Системе TrueSkill приходится вычислять рейтинги игроков, которые объединяются в команды разного размера и участвуют в матчах (турнирах) с несколькими участниками. Задача – после каждого из таких турниров пересчитать апостериорные рейтинги.

Начнём понемногу «разворачивать» то, что происходит в каждом из этих турниров. Во-первых, мы не знаем достоверных априорных значений рейтингов, у нас есть только некоторое априорное распределение (мы считаем его нормальным)

$$f(s_i) = \mathcal{N}(s; \mu_i, \sigma_i).$$

Здесь μ_i – это собственно рейтинг игрока, а σ_i – «показатель достоверности» рейтинга, дисперсия. С каждой сыгранной партией дисперсия уменьшается (если не предпринимать искусственных шагов для её увеличения, но об этом ниже).

Каждый «истинный» скилл¹ является средним значением, вокруг которого распределены конкретные показатели силы игры того или иного игрока в данной конкретной партии (p_i – от слова performance):

$$f(p_i) = \mathcal{N}(p_i; s_i, \beta^2).$$

В системе TrueSkill делается предположение, что β^2 является универсальной константой, общей для всех игроков. Точно так же Арпад Эло, разрабатывая частный случай TrueSkill, предполагал, что каждый игрок может с большой вероятностью сыграть на свой рейтинг плюс-минус 200 очков, или одну «ступень мастерства», и заложил эти 200 очков как универсальную константу рейтинга Эло.

¹Всё-таки трудно нам здесь обойтись без нерусского слова: «сила игры» слишком длинно, а «рейтинг» – это не сам скилл, а наша его оценка.

Практика показывает, что обучать отдельные β^2 слишком сложно (не хватит данных), а предположение это не слишком обременительное.

Легко выразить плотность p_i через исходные параметры, нужно просто проинтегрировать по всем возможным s_i :

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

Затем показатели силы игры игроков в конкретных партиях объединяются и дают оценки на силу игры команд. TrueSkill рассчитывает силу команды как суммарную силу её игроков: $t_i = \sum_j p_j$. Мы ниже будем использовать среднее арифметическое; возможны и другие подходы.

После этого показатели силы команд в данном турнире нужно сравнить друг с другом; их сравнение и должно порождать тот порядок, который записан в результатах турнира. Однако между некоторыми командами может случиться ничья; в этом месте приходится вводить новую константу ϵ . Мы будем считать, что ничья между командами с силой t_1 и t_2 означает, что

$$|t_1 - t_2| < \epsilon.$$

Давайте теперь вспомним, какая задача стоит перед нами. Мы должны подсчитать апостериорные рейтинги команд после получения данных. Данные приходят к нам в виде перестановки команд π : упорядоченных результатов турнира (в которых могут быть ничьи между соседними командами). Иначе говоря, нужно подсчитать

$$p(\mathbf{s} | \pi) = \frac{p(\pi | \mathbf{s})p(\mathbf{s})}{\int p(\pi | \mathbf{s})p(\mathbf{s})d\mathbf{s}}.$$

В нашей системе присутствуют, кроме s_i и π , ещё переменные p_i , t_i и d_i , причём плотность распределения всей системы мы только что представили в виде произведения распределений:

$$p(\pi, \mathbf{d}, \mathbf{t}, \mathbf{p}, \mathbf{s}) = p(\pi | \mathbf{d})p(\mathbf{d} | \mathbf{t})p(\mathbf{t} | \mathbf{p})p(\mathbf{p} | \mathbf{s})p(\mathbf{s}).$$

А нам нужно вычислить

$$p(\pi | \mathbf{s}) = \int \int \int p(\pi, \mathbf{d}, \mathbf{t}, \mathbf{p}, \mathbf{s}) d\mathbf{d}d\mathbf{t}d\mathbf{p}.$$

Следовательно, перед нами обычная задача маргинализации, которая является одним из основных предметов байесовского вывода [5, 11]. В данном случае можно воспользоваться известным алгоритмом маргинализации на графе, представив переменные и функции, участвующие в разложении общего распределения, в виде двудольного графа, вершины одной доли которого представляют собой переменные, другой доли — функции, а ребро проводится тогда и только тогда, когда функция зависит от переменной (см. раздел 2 и ссылки в нём).

Пример фактор-графа модели TrueSkill приведён на рис. 1. Заметим, что при передаче сообщений на верхних уровнях графа проблем не возникает — пересчёт параметров распределений может быть произведён точно (это просто свёртки гауссианов). Но на нижних уровнях возникает проблема: распределения, приходящие из узлов-функций $\mathbb{I}(d_i > \varepsilon)$ и $\mathbb{I}(|d_i| \leq \varepsilon)$, представляют собой гауссианы, обрезанные слева, справа или с двух сторон, и передавать их дальше, пересчитывая параметры всех распределений аналитически, не получается. В такой ситуации обычно применяют алгоритм Expectation Propagation, в котором сложные факторы приближаются более простыми, в нашем случае — гауссовскими [6, 7]. Ценой этого приближения является, в частности, то, что одного прохода по графу недостаточно, и в части графа со сложными факторами нужно повторять передачу сообщений до сходимости. В нашем случае сообщения будут передаваться по нижней части графа, между факторами t_i и d_i , а после сходимости можно будет передать полученные сообщения вверх, завершив вывод.

3. TrueSkill2. Система TrueSkill обобщает ранее известные байесовские рейтинг-системы до самой общей ситуации: несколько команд из разного числа игроков в турнире. Однако у системы TrueSkill есть ряд проблем, которые были отчасти решены первым и третьим автором в [9]; в этом разделе мы описываем эти проблемы, а затем представляем решение, которое в этой работе называется TrueSkill2.

Первая и главная проблема TrueSkill — неудачная обработка *ничьих из нескольких команд*. В системе TrueSkill ничьи обрабатываются при помощи узла-функции $\mathbb{I}(|d_{i+1} - d_i| \leq \varepsilon)$, а ничьи, в которых одно место делят несколько команд, представляются как последовательность из нескольких таких узлов. Это может привести к неверному поведению: если $|d_{i+1} - d_i| \leq \varepsilon$ и $|d_{i+2} - d_{i+1}| \leq \varepsilon$,

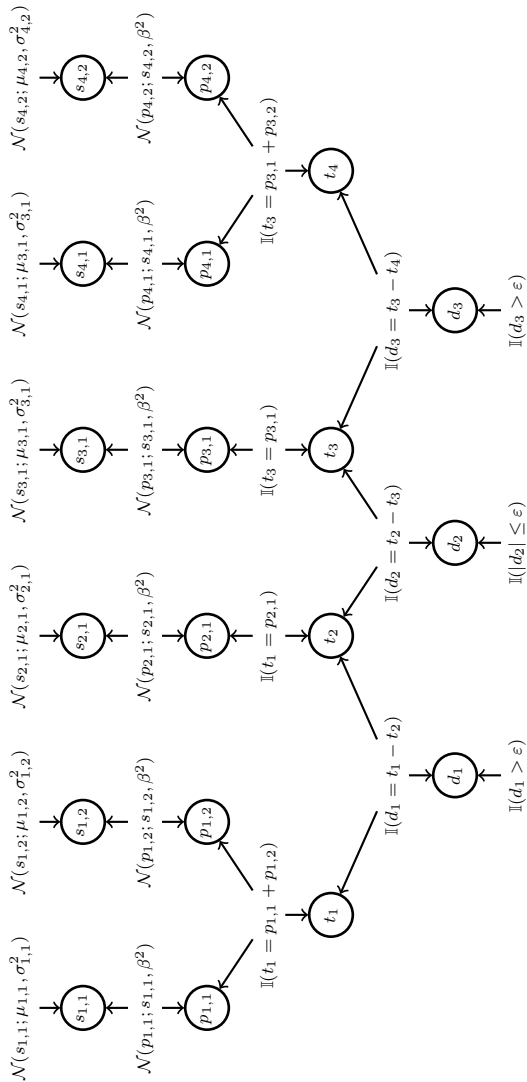


Рис. 1: Пример фактор-графа модели TrueSkill: в матче участвовали четыре команды, команды 1 и 4 состояли из двух игроков, 2 и 3 – из одного. Команда 1 победила, 2 и 3 разделили второе-третье места, а команда 4 заняла последнее место.

это значит, что $|d_{i+2} - d_i| \leq 2\varepsilon$, и этот эффект накапливается. Поскольку победы между соседними командами представляются как $|d_{i+1} - d_i| > \varepsilon$, может даже случиться, что максимальное апостериорное значение силы игры в данном турнире у команды, занявшей более высокое место, окажется ниже, чем у команды, занявшей более низкое место! Так, в примере на рис. 1 максимальное апостериорное значение силы игры, показанной командой 4, t_4 , может быть сколь угодно близко к соответствующему значению для второй команды t_2 , а если бы была ещё одна команда, поделившая места с t_2 и t_3 , это значение могло бы и превысить t_2 . Эта ситуация на самом деле весьма вероятна на практике, если победила команда, от которой этого не ожидалось, и априорное распределение её силы игры было сосредоточено ниже априорного распределения команды 4; мы наблюдали этот эффект на наших данных.

Мы решаем эту проблему при помощи новой структуры фактор-графа. Чтобы справиться с ничьими из нескольких команд, мы добавляем к фактор-графу TrueSkill ещё один уровень. Этот лишний уровень объединяет поделившие место команды в единый узел, который и будет далее участвовать в приближённом выводе. Новые переменные l_k соответствуют «средней» силе игры, показанной в данном турнире поделившими одно и то же место командами. Если команды t_1, \dots, t_s разделили место, мы вводим переменную l и связываем её с t_i при помощи фактора $\mathbb{I}(|l - t_i| \leq \varepsilon)$. Таким образом, на нижнем уровне все d_k теперь соответствуют разным местам (точнее, уровням мест). Чтобы отделить уровни l_k друг от друга, мы вводим факторы $l_{k+1} - l_k = d_k > 2\varepsilon$. Получившийся фактор-граф (для примера из рис. 1) показан на рис. 2.

Фактор-граф, показанный на рис. 2, содержит и решение ещё одной проблемы TrueSkill: предположения о том, что сила команды является суммой сил её игроков. Во многих соревнованиях и задачах сравнения команда с меньшим числом игроков вполне конкурентоспособна против команды полного размера, в то время как сумма сил предсказывает огромную разницу между ними. В результате команда, победившая команду с большим числом игроков, получает несправедливо большой бонус, а проигравшая команда несправедливо сильно падает в рейтинге. К счастью, в рамках модели TrueSkill эту проблему решить несложно – в алгоритмах вывода ничего не изменится, если заменить функцию силы команды на любую линейную комбинацию, например среднюю силу

или, как в результате и было сделано в TrueSkill2, среднюю силу, дополнительно пенализированную за неполный состав; размер штрафа можно дополнительно обучить из данных. Коэффициенты линейной комбинации могут даже зависеть от априорных значений рейтингов; они подаются на вход модели в момент построения фактор-графа, т.е. непосредственно перед обработкой очередного турнира.

В нашем конкретном приложении (см. раздел 6) максимальный размер команды составляет 6 игроков; в [9] был предложен штраф к силе команды в размере 2% за каждого недостающего игрока. Если же списочный состав команды превышает 6 (в наших данных разрешены замены по ходу турнира), сила команды нормализуется к среднему значению лучших шести игроков, т.е. t представляет собой сумму $p_{i,j}$ с одинаковыми весами, равными среднему лучших 6 $\mu_{i,j}$, делённому на сумму всех $\mu_{i,j}$. Предполагая, что $m_{i,j}$ отсортированы по убыванию, получаем следующую (линейную) функцию силы команды:

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

где n_i – число игроков в команде i .

4. TrueSkill2 с дополнительной информацией о результатах турниров. В ситуации реальных турниров (или даже «турниров», которые на самом деле являются сравнениями другого рода) нередко оказывается доступной дополнительная информация о результатах матчей. В частности, очень естественно, что во многих соревнованиях, кроме собственно перестановки, определяющей места по итогам матча или турнира, доступна также информация о том, с каким перевесом победила та или иная команда. Эта информация представляет собой дополнительные сведения о различиях в продемонстрированной силе команд и, таким образом, может и должна быть использована в модели для дальнейшего уточнения апостериорных оценок.

Основная идея учёта дополнительной информации о результатах турнира очень проста: в базовой модели TrueSkill и TrueSkill2

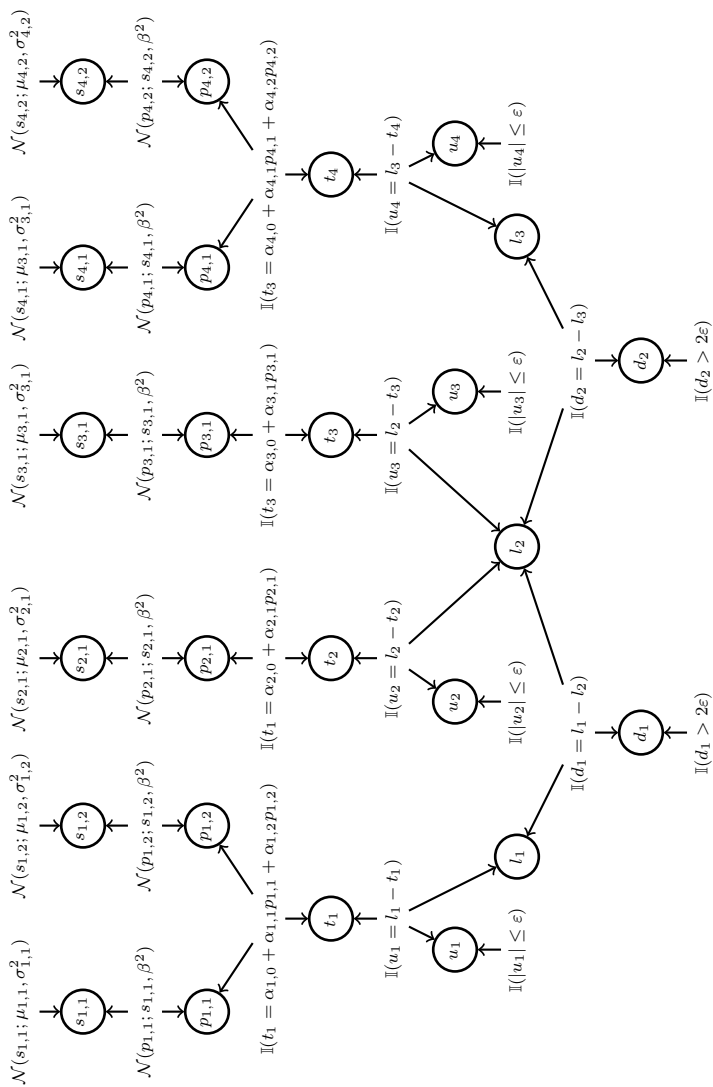


Рис. 2: Фактор-граф системы TrueSkill2 для турнира, изображённого на рис. 1.

мы вводили фактор $\mathbb{I}(d_i > 2\varepsilon)$ на нижнем уровне модели между местами (уровнями мест) i и $i + 1$, а теперь, если известно число очков x_i и x_{i+1} , мы введём вместо этого фактор

$$\mathbb{I}(d_i > 2\varepsilon f(x_i - x_{i+1}))$$

для некоторой положительной функции f , монотонно возрастающей с ростом разницы $x_i - x_{i+1}$. Функцию f можно выбирать любой – какой бы сложной она ни была, в любом случае в самой модели она представляет собой не случайную переменную, а константу, и алгоритм вывода для модели TrueSkill2 переносится на этот случай практически без изменений.

Заметим, однако, что теперь в модели появилась новая степень свободы, которую можно варьировать – собственно вид функции f . Вряд ли возможно выбрать раз и навсегда оптимальный вид функции f , её нужно подбирать в зависимости от природы исходной задачи. В экспериментах на нашем наборе данных (см. описание датасета и результатов экспериментов в разделе 6) достаточно неожиданно оказалось, что тождественная функция f (т.е. решение просто умножить 2ε на разницу очков) привело даже к некоторому небольшому ухудшению результатов, а лучшие результаты из полученных нами показала квадратичная функция $f(x_i - x_{i+1}) = (x_i - x_{i+1})^2$.

Общий вид фактор-графа модели с учётом результатов турнира (эта модель получила название TrueSkill2.5) показана на рис. 3.

5. Результаты экспериментов. Для оценки результатов предсказания моделей мы использовали достаточно естественную метрику – долю пар команд, для которых их относительное положение было предсказано верно. Рис. 4 показывает результаты сравниваемых моделей. Исходный набор данных включал в себя длинный начальный отрезок, для которого точные результаты турниров были недоступны; таким образом, рис. 4 начинается с момента, когда заканчиваются данные турниров без точных результатов и начинаются данные турниров с точными результатами. Видно, что наилучший результат показывает модель TrueSkill2.5 с квадратичной функцией разницы силы игры, а модель TrueSkill2.5 с линейной функцией разницы силы игры незначительно улучшает базовую модель TrueSkill2.

6. Заключение. В этой работе мы ввели и подробно рассмотрели обобщение системы TrueSkill2 на случай дополнительных

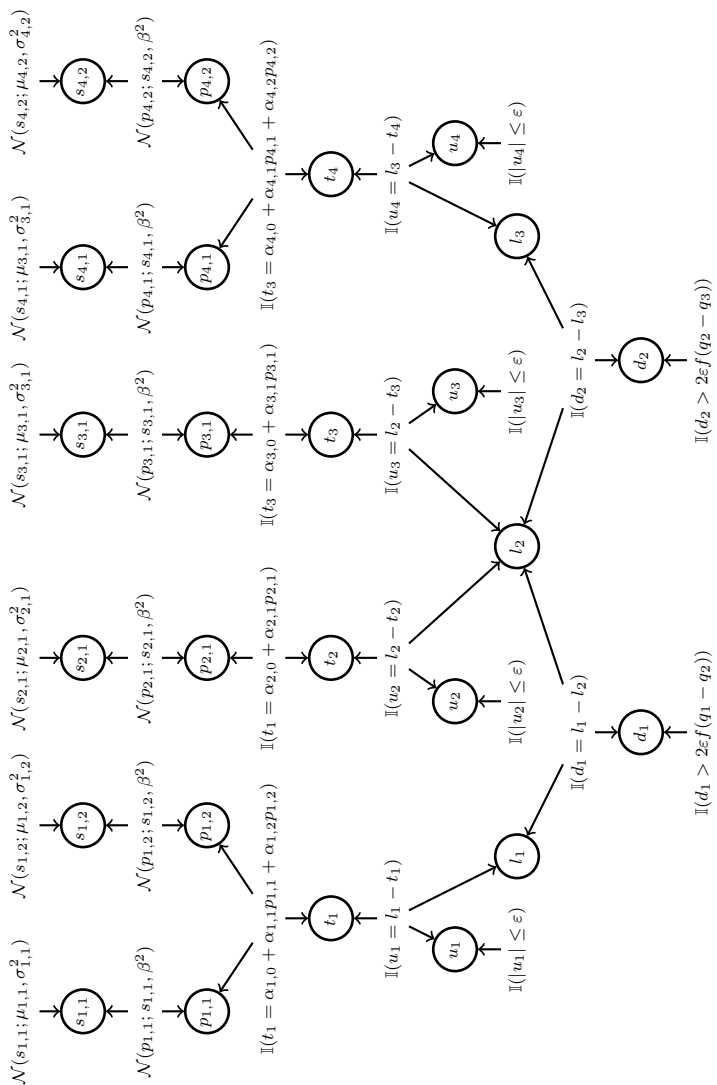


Рис. 3: Пример фактор-графа системы TrueSkill2.5, учитывающей численные результаты, для турнира, изображённого на рис. 1.

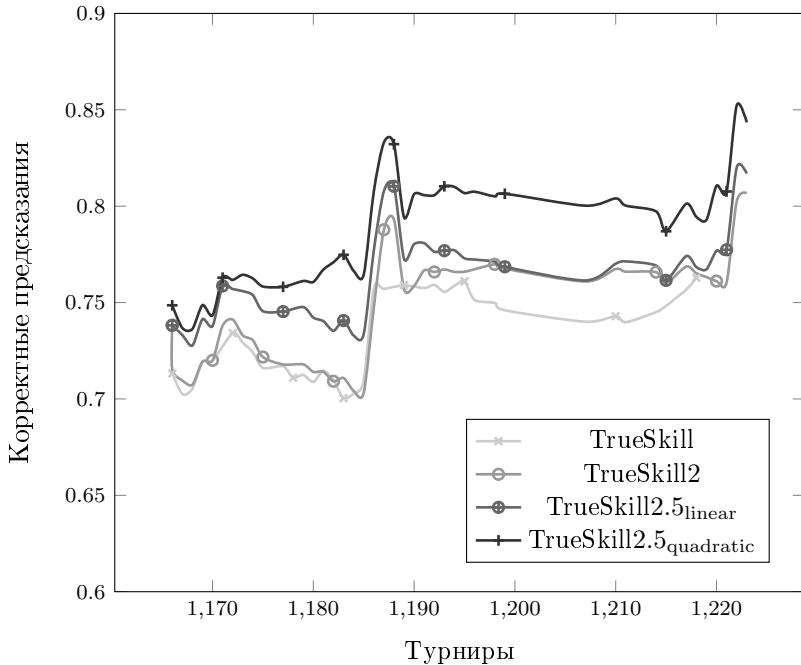


Рис. 4: Доля корректных предсказаний – турниры с известной дополнительной информацией.

имеющихся данных о том, какие численные результаты были показаны теми или иными командами в турнире. Мы показали, что учёт такой дополнительной информации даёт преимущество над базовой системой TrueSkill2, и привели фрагменты программного кода, реализующего полученную систему в среде Infer.NET.

Модели байесовских рейтинг-систем относятся к бурно развивающемуся разделу машинного обучения – вероятностным графическим моделям. Задачи ранжирования, решаемые такими моделями, могут получить применения в системах оценки качества интернет-поиска, оценки и предсказания качества интернет-рекламы. Огромный объём данных, которые обычно нужно обрабатывать в таких приложениях, делает особенно важными теоретические и практические исследования в области разработки параллельных алгоритмов вывода в рейтинг-моделях, которые мы планируем сде-

лать центром нашей дальнейшей работы в данной области.

Литература

1. *Coulom R.* Computing Elo Ratings of Move Patterns in the Game of Go // ICGA Journal. December 2007. Vol. 30, N. 4. P. 198–208.
2. *Graepel T., Candela J. Q., Borchert T., Herbrich R.* Web-Scale Bayesian Click-Through Rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine // Proceedings of the 27th International Conference on Machine Learning. 2010. P. 13–20.
3. *Graepel T., Minka T., Herbrich R.* TrueSkill(TM): A Bayesian Skill Rating System // Advances in Neural Information Processing Systems 19 / Ed. by B. Schölkopf, J. Platt, T. Hoffman. Cambridge, MA: MIT Press, 2007. P. 569–576.
4. *Huang T.-K., Weng R. C., Lin C.-J.* Generalized Bradley–Terry Models and Multi-class Probability Estimates // Journal of Machine Learning Research. 2006. Vol. 7. P. 85–115.
5. *MacKay D. J.* Information Theory, Inference and Learning Algorithms. Cambridge University Press, 2003.
6. *Minka T.* Expectation Propagation for approximate Bayesian inference // Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence / Ed. by J. S. Breese, D. Koller. San Francisco, CA: Morgan Kaufmann Publishers Inc., 2001. P. 362–369.
7. *Minka T.* A family of algorithms for approximate Bayesian inference: Ph.D. thesis / Massachusetts Institute of Technology. 2001.
8. *Nikolenko S. I., Sirotkin A. V.* Extensions of the TrueSkillTM rating system // Proceedings of the 9th International Conference on Applications of Fuzzy Systems and Soft Computing. 2010. P. 151–160.
9. *Nikolenko S. I., Sirotkin A. V.* A New Bayesian Rating System for Team Competitions // Proceedings of the 28th International Conference on Machine Learning. 2011.
10. *Stein A., Aryal J., Gort G.* Generalized Bradley–Terry Models and Multi-class Probability Estimates // IEEE Transactions on Geoscience and Remote Sensing. 2005. Vol. 43. P. 852–856.
11. *Tulupyyev A. V., Nikolenko S. I., Sirotkin A. V.* Bayesian networks: a logical probabilistic approach. St. Petersburg, Nauka, 2006. 608 p.

12. *Wu T.-F., Lin C.-J., Weng R. C. Probability Estimates for Multi-class Classification by Pairwise Coupling // Journal of Machine Learning Research. 2004. Vol. 5. P. 975–1005.*

Николенко Сергей Игоревич — к.ф.-м.н.; научный сотрудник лаборатории математической логики ПОМИ РАН, доцент кафедры математических и информационных технологий, старший научный сотрудник проблемной лаборатории вычислительной биологии СПбАУ НОЦНТ РАН; snikolenko@gmail.com; СПбАУ НОЦНТ РАН, ул. Хлопина, д. 8, корп. 3, г. Санкт-Петербург, 194021, РФ; р.т. +7(812)297-2145, факс +7(812)448-6998.

Sergey I. Nikolenko — Ph.D.; Researcher at Steklov Mathematical Institute, Adjunct Prof. at the Chair of Mathematical and Information Technologies, Senior Researcher at the Laboratory of Algorithmic Biology at St. Petersburg Academic University; snikolenko@gmail.com; St. Petersburg Academic University, ul. Khlopina, d. 8, korp. 3, St. Petersburg, 194021, Russia; office phone +7(812)297-2145, fax +7(812)448-6998.

Сердюк Дмитрий Владимирович — студент 5 курса СПбАУ НОЦНТ РАН; serdyuk.dmitriy@gmail.com; СПбАУ НОЦНТ РАН, ул. Хлопина, д. 8, корп. 3, г. Санкт-Петербург, 194021, РФ; р.т. +7(812)297-2145, факс +7(812)448-6998. Научный руководитель – С.И. Николенко.

Dmitriy V. Serdyuk — M.Sc. student at St. Petersburg Academic University; serdyuk.dmitriy@gmail.com; St. Petersburg Academic University, ul. Khlopina, d. 8, korp. 3, St. Petersburg, 194021, Russia; office phone +7(812)297-2145, fax +7(812)448-6998. Advisor – S.I. Nikolenko.

Сироткин Александр Владимирович — к.ф.-м.н.; младший научный сотрудник лаборатории теоретических и междисциплинарных проблем информатики СПИИРАН, научный сотрудник проблемной лаборатории вычислительной биологии СПбАУ НОЦНТ РАН; alexander.sirotkin@gmail.com; СПбАУ НОЦНТ РАН, ул. Хлопина, д. 8, корп. 3, г. Санкт-Петербург, 194021, РФ; р.т. +7(812)297-2145, факс +7(812)448-6998.

Alexander V. Sirotkin — Ph.D.; Researcher at the Laboratory of Theoretical and Interdisciplinary Problems of Computer Science at SPIIRAS, Junior Researcher at the Laboratory of Algorithmic Biology at St. Petersburg Academic University; alexander.sirotkin@gmail.com; St. Petersburg Academic University, ul. Khlopina, d. 8, korp. 3, St. Petersburg, 194021, Russia; office phone +7(812)297-2145, fax +7(812)448-6998.

Поддержка исследований. Работа была поддержана грантами РФФИ 11-01-00760-а и 11-01-12135-офи-м-2011, грантом Президента РФ для молодых кандидатов наук МК-6628.2012.1, грантом Президента РФ для ведущих научных школ НШ-3229.2012.1 и Федеральной целевой программой «Научные и научно-педагогические кадры инновационной России» за 2009–2013 гг.

Рекомендовано ТИМПИ СПИИРАН, зав.лаб. Тулупьев А.Л., д.ф.-м.н.

Статья поступила в редакцию 17.04.2012.

РЕФЕРАТ

Николенко С.И., Сердюк Д.В., Сироткин А.В. **Байесовские рейтинг-системы с учётом дополнительной информации о результатах.**

Мы рассматриваем задачу построения байесовской рейтинг-системы, в которой из результатов отдельных матчей/турниров (упорядочиваний на небольшом множестве) обучается единая ранжирующая функция.

Рейтинг-системы, осуществляющие «честную» оценку силы игры, известны давно; первой из широко известных таких систем стал рейтинг Эло, внедрённый в 1950-е годы. Одной из наиболее общих таких систем является рейтинг-система TrueSkill, разработанная в компании Microsoft для игровых серверов Xbox 360. Ключевые сложности при создании подобной рейтинг-системы заключаются в том, что постановка задачи теперь становится максимально общей: в матчах участвуют команды из заранее неизвестного числа игроков, а рейтинг нужно в итоге вывести индивидуальный. Однако у системы TrueSkill есть ряд проблем: выбранная в ней функция силы команды плохо обобщается на случай матча команд с разным числом игроков, а главное, TrueSkill очень неудачно обрабатывает делёж мест большим числом команд.

Эти проблемы были отчасти решены первым и третьим автором ранее; в этой работе мы продолжаем наши предыдущие исследования. Расширяя построенную нами систему TrueSkill2, мы строим рейтинг-систему TrueSkill2.5, которая может учитывать дополнительную численную информацию о результатах турниров: очки, набранные командами в турнире. При этом в системе добавляется новый параметр – функция зависимости разницы силы игры от разницы в результатах – который нужно подбирать для конкретного датасета.

Работа содержит подробные описания всех упомянутых моделей байесовских рейтинг-систем, алгоритмы вывода и результаты экспериментов на репрезентативном наборе данных.

SUMMARY

Nikolenko S.I., Serdyuk D.V., Sirotkin A.V. **Bayesian rating systems with additional information on tournament results.**

We consider the problem of constructing a Bayesian rating system that learns a general ranking function from the results of individual matches and/or tournaments (rankings on small subsets).

Rating systems that perform an “honest” estimate of the skills have been known for quite some time: the first widely known of this kind system was the Elo rating introduced in the 1950s. One of the most general systems of this kind is the TrueSkill rating system developed in Microsoft for Xbox 360 gaming servers. The key problem in such a rating system is that the problem setting now becomes the most general: matches/tournaments can contain teams with varying number of players not known in advance, and the system has to learn individual rating. However, the TrueSkill system has a number of problems: its team performance function does not generalize well to teams with varying number of players, and, most importantly, TrueSkill fails on large multiway ties that are characteristic for many applications.

These problems have been previously partially solved by the first and third author; in this work we continue our previous research. Extending the TrueSkill2 system, we construct a new rating system TrueSkill2.5 that can incorporate additional numerical information regarding tournament results, namely the points a team has scored in a tournament. The system also gets a new parameter, the function of how performance difference depends on the difference in results, that needs to be tuned for a specific application.

The work contains detailed descriptions of all above-mentioned Bayesian rating models, inference algorithms, and experimental results on a representative dataset.