РЕЙТИНГ-СИСТЕМЫ С ТОЧКИ ЗРЕНИЯ БАЙЕСОВСКОГО ВЫВОДА

Николенко С.И., Сироткин А.В.

УДК 004.85

Аннотация. Задача построения «честной» рейтинг-системы, в которой из числовых значений рейтингов можно будет оценить вероятность победы того или иного участника над другими, давно стала предметом статистического анализа, однако результаты этого анализа малоизвестны. Мы рассказываем о статистической основе рейтинга Эло, обобщаем её до вариаций модели Брэдли–Терри, а затем описываем одну из последних разработок на этом пути — систему TrueSkill. — 20 с., 17 источников.

Ключевые слова: рейтинг-системы, байесовский вывод, математическая статистика, модель Брэдли-Терри.

Abstract. The task of creating an "honest" rating system that would allow to estimate probabilities of winning for each its participant has for a long time been a subject of statistical analysis. However, the results of this analysis are much less known than they deserve to be. We describe the statistical basis of the Elo rating system, generalize it to the Bradley–Terry model with variations and, finally, describe one of the latest innovations — the TrueSkill rating system.

Keywords: rating systems, Bayesian inference, mathematical statistics, Bradley-Terry model.

1. Введение. Рейтинг-системы, о которых пойдёт речь в этом обзоре, 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 применяемых таких систем стал рейтинг Эло, внедрённый в шахматном ире в пятидесятые годы. В этом кратком обзоре мы сначала, в разделе 2, изложим сам рейтинг Эло, затем в разделе 3 поговорим о статистическом обосновании рейтинга Эло и его обобщений — модели Брэдли-Терри, а разделы 4 и 5 будут посвящены недавней разработке Microsoft Research, системе TrueSkill.

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 [6, 15]. Аналогичные задачи — по частичным сравнениям восстановить полную картину — возникают и во многих других приложениях, например, в обработке изображений [14] или даже сравнении паттернов в игре го [3].

2. Рейтинг Эло. Основную суть рейтинговых систем с математической точки зрения лучше всего будет изложить на самом известном, но вместе с тем и одном из самых простых примеров: на примере рейтинга Эло. В пятидесятых годах XX века венгерский физик и шахматист Арпад Эло осознал, что существовавший в то время рейтинг федерации шахмат США (USCF), так называемый «рейтинг Харкнесса» (Harkness Rating System получила название в честь своего автора Кеннета Харкнесса и была принята на вооружение USCF в 1950 г.) имеет недостаточно строгое математическое обоснование, а потому не слишком достоверно отражает силу игроков. Эло разработал модель, в которой рейтинги игроков изменялись после партии в зависимости от того, какой результат был ожидаемым и какой на самом деле произошёл [4]. В этом разделе мы рассмотрим модель Эло с точки зрения байесовского вывода; в этой краткой и не предназначенной для специалистов заметке трудно подробно изложить всю теорию байесовской статистики, но основные идеи мы постараемся рассказать достаточно доступно.

Во-первых, нужно ответить на основополагающий вопрос: что такое рейтинг? Мы хотели бы, чтобы рейтинг был мерилом силы игры; однако при этом совершенно очевидно, что сила игры от партии к партии может достаточно сильно меняться под воздействием внешних и внутренних случайных факторов. Таким образом, на самом деле сила игры того или иного участника в конкретной партии (сравнение этих сил и определяет исход партии) — это *случайная величина*, а само значение рейтинга — его среднее, т.е. *математическое ожидание* этой случайной величины. Мы пока будем рассматривать простейший случай, в котором сила игры участника нормально распределена вокруг его рейтинга[1], т.е. плотность вероятности силы игры в конкретной партии равна

$$p(x) = \mathcal{N}(x; s, \beta) = \frac{1}{\beta\sqrt{2\pi}} e^{-\frac{1}{2\beta^2}(x-s)^2}$$

для некоторого $\beta > 0$; здесь и далее буквой s мы будем обозначать рейтинги (от слова skill). Таким образом, сила игры задаётся двумя параметрами: средним s (собственно рейтингом) и дисперсией β^2, которая показывает, насколько точна наша оценка.

Математически задачу рейтинга можно сформулировать так: по имеющимся данным D (результатам всех сыгранных участниками рейтинга партий) получить такие оценки величин s и β^2 для каждого участника, которые максимизируют вероятность

$$p(s, \beta^2 \mid D).$$

Это — классическая постановка задачи поиска максимальной апостериорной гипотезы; по теореме Байеса её можно переформулировать как поиск

$$\arg\max_{s,\beta^2} p(s, \beta^2 \mid D) = \arg\max_{s,\beta^2} \frac{p(D \mid s, \beta^2)p(s, \beta^2)}{p(D)} = \\ = \arg\max_{s,\beta^2} p(D \mid s, \beta^2)p(s, \beta^2).$$

Здесь $p(s, \beta^2)$ — *априорная вероятность* параметров, т.е. плотность распределения вероятностей параметров, которая была до опыта,

[1] FIDE в конце концов пришла к тому, что нормальное распределение убывает слишком быстро, и заменила его на более пологий аналог, *логистическое распределение*; мы поговорим о нём чуть ниже, а пока наша цель — выработать интуицию.

до получения данных D, а $p(s, \beta^2 \mid D)$, которую мы хотим вычислить — *апостериорная вероятность*, плотность распределения после учёта данных D.

Эло предположил (и этим предположением мы тоже будем пользоваться), что дисперсия силы игры в конкретной партии постоянна и от игрока не зависит; таким образом, β^2 в формуле выше — просто константа. Кроме того, в качестве априорного распределения $p(s)$ разумно и естественно выбрать тоже нормальное распределение[2] $\mathcal{N}(s; \mu, \sigma)$. Таким образом, рейтинг игрока на самом деле складывается из двух чисел: его среднего значения μ и его дисперсии σ^2. Значение μ отображается в таблице рейтингов, а σ^2 характеризует, насколько достоверна имеющаяся оценка. Естественно предположить, что для разумного правила пересчёта рейтингов с ростом количества сыгранных партий σ^2 должна убывать, а μ — стремиться к истинному значению s.

Осталось только понять, как вычислить правдоподобие $p(D \mid s, \beta^2)$. Давайте предположим, что встречаются два игрока (обобщениями на команды мы будем заниматься позже, в разделе 3) с некоторыми априорными распределениями на рейтинги $\mathcal{N}(s_1; \mu_1, \sigma_1^2)$ и $\mathcal{N}(s_2; \mu_2, \sigma_2^2)$. Тогда случайная величина, характеризующая силу каждого из них в этой конкретной партии, имеет распределение

$$p(x \mid \mu, \sigma) = \int_{-\infty}^{\infty} p(x \mid s) p(s \mid \mu, \sigma) ds = \int_{-\infty}^{\infty} \mathcal{N}(x; s, \beta) \mathcal{N}(s; \mu, \sigma) ds =$$
$$= \int_{-\infty}^{\infty} \frac{1}{\beta\sqrt{2\pi}} e^{-\frac{1}{2\beta^2}(x-s)^2} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(s-\mu)^2} ds = \mathcal{N}(x; \mu_x, \sigma_x),$$

то есть мы снова приходим к нормальному распределению, но с другими параметрами.

3. Модель Брэдли-Терри и её обобщения. В этом разделе мы излагаем статистическую суть рейтинга Эло, а также переходим от простейшей модели (два игрока в каждой игре, ничьи невозможны) к обобщениям, в том числе к обобщениям, которые приводят к максимально общей модели рейтинга: команды по несколько игроков (причём это количество не обязательно одинаково), возможны ничьи.

[2]Мы не будем здесь подробно на этом останавливаться, но это связано с тем, что нормальное распределение является *сопряжённым* к самому себе; подробнее о сопряжённых распределениях и вообще о байесовском выводе см. [5, 9, 16].

Суть задачи, конечно, можно сформулировать и безо всякого рейтинга. Дано некоторое множество $X = \{x_1, \ldots, x_m\}$ и отображение $f : X \to \mathbb{R}$. Значения $f(x)$ для разных $x \in X$ многократно сравниваются друг с другом, но не полностью, а частично: результатом каждого сравнения является линейный порядок на некотором подмножестве $X' \subseteq X$. Более того, результаты каждого сравнения носят случайный характер: порядок, выданный в результате сравнения, на самом деле определяется не точным значением $f(x)$, а значением некоторой случайной величины, среднее которой равно $f(x)$. По исходным данным такого рода требуется определить наиболее правдоподобные значения $f(x)$ для каждого $x \in X$; обычно на самом деле ищутся все параметры вышеупомянутых случайных величин, т.е., как правило, среднее и дисперсия (эти случайные величины обычно считаются нормально или логистически распределёнными). Мы в дальнейшем будем, как и принято в этой области, называть элементы X «участниками», а значение функции $f(x)$ — «силой (skill) участника x».

Исходная модель Брэдли-Терри [2], как и рейтинг Эло, предназначена для того, чтобы вычислять рейтинги, приближающие истинные силы участников, по имеющимся данным попарных сравнений. Модель основана на крайне простом предположении: она предполагает, что для участников $1, \ldots, n$ можно подобрать такие константы γ_i, $i = 1..n$, что вероятность победы участника i над участником j равна

$$p(i \text{ побеждает } j) = \frac{\gamma_i}{\gamma_i + \gamma_j}.$$

Таким образом, основная задача заключается в том, чтобы найти $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_m)$ максимального правдоподобия из имеющихся данных D. Для этого, если принять априорное распределение равномерным, можно просто максимизировать правдоподобие

$$p(D|\boldsymbol{\gamma}) = \prod_{i=1}^{m} \prod_{j=1}^{m} \left(\frac{\gamma_i}{\gamma_i + \gamma_j} \right)^{w_{ij}},$$

где w_{ij} — то, сколько раз x_i обыграл x_j при их попарном сравнении ($w_{ii} = 0$ по определению), или, что эквивалентно, максимизировать

логарифм правдоподобия

$$l(\boldsymbol{\gamma}) = \sum_{i=1}^{m} \sum_{j=1}^{m} \left(w_{ij} \log \gamma_i - w_{ij} \log(\gamma_i + \gamma_j) \right).$$

На этом математическая суть заканчивается и начинается алгоритмическая. Мы хотим построить итеративную процедуру, которая бы постепенно приходила к максимуму функции $l(\boldsymbol{\gamma})$, начиная с некоторого начального значения. Один из наиболее эффективных способов сделать это — максимизировать $l(\boldsymbol{\gamma})$ при помощи так называемого MM-алгоритма (от слов minorization-maximization). Предположим, что мы хотим максимизировать функцию $f(\theta)$ и сумели найти такую функцию $g(\theta, \theta^n)$ (от двух похожих на θ аргументов), что

$$g(\theta, \theta^n) \leq f(\theta) \text{ для любых } \theta, \theta^n,$$
$$f(\theta^n) = g(\theta^n, \theta^n).$$

Найдём теперь максимум функции g по первому аргументу (по θ):

$$\theta^{n+1} = \max_{\theta} g(\theta, \theta^n).$$

Тогда верно, что

$$f(\theta^{n+1}) = g(\theta^{n+1}, \theta^n) \geq g(\theta^n, \theta^n) = f(\theta^n).$$

Оказывается, что если сместиться из точки θ^n в точку θ^{n+1}, значение функции $f(\theta)$ при этом увеличится. Это — основная идея MM-алгоритмов: если мы сумеем придумать вспомогательную функцию g, *миноризующую* функцию f ($g \leq f$), а затем её *максимизировать*, то мы тем самым перейдём в точку, где значение f увеличится. Разумеется, надо придумывать такие функции, для которых будет легко максимизировать $g(\theta, \theta^n)$ по θ; как правило, ищут функции, в которых эта максимизация распадается на независимые одномерные задачи максимизации для каждой координаты вектора θ.

В нашем конкретном примере функция g будет вытекать из общего свойства логарифма: поскольку $\log x \leq x - 1$, то (подставляя $\frac{x}{y}$)

$$1 + \log \frac{x}{y} - \frac{x}{y} \leq 0.$$

Рассмотрим теперь вспомогательную функцию

$$Q(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(k)}) = \sum_{i,j} w_{ij} \left[\log \gamma_i - \frac{\gamma_i + \gamma_j}{\gamma_i^{(k)} + \gamma_j^{(k)}} - \log \left(\gamma_i^{(k)} + \gamma_j^{(k)} \right) + 1 \right].$$

Легко видеть, что она действительно миноризует $l(\boldsymbol{\gamma})$ (нужно применить вышеописанное свойство логарифма к $x = \gamma_i + \gamma_j$ и $y = \gamma_i^{(k)} + \gamma_j^{(k)}$). А чтобы найти $\max_{\boldsymbol{\gamma}} Q\left(\boldsymbol{\lambda}, \boldsymbol{\gamma}^{(k)}\right)$, можно просто взять производные Q по каждому из γ_i:

$$\frac{\partial Q}{\partial \gamma_l} = \sum_{i,j} w_{ij} \left[\frac{\delta_{il}}{\gamma_i} - \frac{\delta_{il} + \delta_{jl}}{\gamma_i^{(k)} + \gamma_j^{(k)}} \right] =$$

$$= \frac{1}{\gamma_l} \sum_j w_{lj} - \sum_j \frac{w_{lj}}{\gamma_i^{(k)} + \gamma_j^{(k)}} - \sum_i \frac{w_i l}{\gamma_i^{(k)} + \gamma_j^{(k)}}.$$

Обозначая через w_l общее количество побед игрока l ($w_l = \sum_j w_{lj}$), а через N_{ij} — количество встреч между игроками i и j ($N_{ij} = w_{ij} + w_{ji}$), получаем

$$\frac{w_l}{\gamma_l} - \sum_j \frac{N_{lj}}{\gamma_i^{(k)} + \gamma_j^{(k)}} = 0.$$

Итак, правило пересчёта на одной итерации поиска максимума $l(\boldsymbol{\gamma})$ выглядит так:

$$\gamma_l^{(k+1)} = w_l \left[\sum_j \frac{N_{lj}}{\gamma_i^{(k)} + \gamma_j^{(k)}} \right]^{-1}.$$

После того как значения рейтингов в последовательности $\gamma^{(1)}, \gamma^{(2)}, \ldots$ «устаканятся» (новые итерации перестанут их существенно менять), можно будет

В итоге мы получили алгоритм, достаточно быстро (см. сравнение с итерациями Ньютона-Рапсона в [7]) сходящийся к истинным значениям $\boldsymbol{\gamma}$ в ситуации, когда игроки сравниваются друг с другом попарно, и у каждой пары есть победитель. Этого, конечно, недостаточно даже для шахмат (в шахматах бывают ничьи), не говоря уже о видах спорта, в котором в турнире сравниваются сразу много команд.

К счастью, аналогичные методы легко проходят и для ситуации с ничьими, и для других обобщений, в том числе для ситуации, когда в одной «партии» участвуют более двух игроков [7]. Если, к примеру, результат может зависеть от порядка элементов в паре (скажем, команды проводят «домашние» матчи и «гостевые»), то можно ввести дополнительный параметр $\theta > 0$, характеризующий, насколько большое преимущество дают «родные стены», и рассмотреть модель [1], в которой

$$p(i \text{ побеждает } j) = \begin{cases} \frac{\theta\gamma_i}{\theta\gamma_i+\gamma_j}, & \text{если } i \text{ играет дома,} \\ \frac{\gamma_j}{\theta\gamma_i+\gamma_j}, & \text{если } j \text{ играет дома.} \end{cases}$$

Если возможны ничьи, то их вероятность тоже можно описать дополнительным параметром $\theta > 1$, и это приводит к модели [13], в которой

$$\begin{aligned} p(i \text{ побеждает } j) &= \frac{\gamma_i}{\gamma_i + \theta\gamma_j}, \\ p(j \text{ побеждает } i) &= \frac{\gamma_j}{\theta\gamma_i + \gamma_j}, \\ p(i \text{ и } j \text{ играют вничью}) &= \frac{(\theta^2-1)\gamma_i\gamma_j}{(\gamma_i + \theta\gamma_j)(\theta\gamma_i + \gamma_j)}. \end{aligned}$$

Обобщение на несколько игроков в одном турнире несколько сложнее, но тоже достаточно прямолинейно [10, 12]. Оно присваивает данной перестановке π подмножества игроков $A = \{1, \ldots, k\}$ (результату турнира) вероятность

$$p_A(\pi) = \prod_{i=1}^{k} \frac{\gamma_{\pi(i)}}{\gamma_{\pi(i)} + \gamma_{\pi(i+1)} + \ldots + \gamma_{\pi(k)}}.$$

Можно показать, что такая модель эквивалентна весьма естественной аксиоме Люса [8]: для любой модели, в которой вероятности игроков обыграть друг друга в любой паре не равны нулю, для любых подмножеств игроков $A \subset B$ и любого игрока $i \in A$

$$\begin{aligned} &p_B(i \text{ побеждает}) = \\ &\quad = p_A(i \text{ побеждает}) p_B(\text{побеждает кто-то из множества } A). \end{aligned}$$

Однако прямолинейное обобщение на случай, когда команды могут менять свои составы между турнирами, окажется в этом случае слишком сложным; разработать для него MM-схему при таком подходе было бы непросто. Есть, правда, и другие методы обучения моделей Брэдли-Терри; например, недавнее исследование показывает, что модель Брэдли-Терри в весьма высокой степени общности можно параметризовать как искусственную нейронную сеть и соответствующими способами обучить [11]. Мы предлагаем заинтересовавшемуся читателю попытаться всё же решить статистическую задачу и разработать MM-схему, а сами тем временем перейдём к другой схему максимизации правдоподобия, которая производит байесовский вывод на графе — систему TrueSkill.

4. Рейтинг-система TrueSkill. Рейтинг-система TrueSkill была разработана 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. 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» скилл[3] является средним значением, вокруг которого распределены конкретные показатели силы игры того или иного игрока в данной конкретной партии (p_i — от слова performance):

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

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

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. 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:

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

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: $t_i = \sum_i p_i$. 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

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

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:

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

А нам нужно вычислить

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

Следовательно, перед нами обычная задача маргинализации, которая является одним из основных предметов байесовского вывода [9,17]. 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 [9]. В следующем разделе мы рассмотрим несколько конкретных примеров вычислений на таких графах.

5. Несколько примеров. В этом разделе мы на нескольких примерах продемонстрируем работу описываемого метода, а также обратим внимание на некоторые неявные предположения. Начнем с простейшего примера, когда в матче всего два участника. Чтобы соблюсти все формальности, мы будем считать, что каждый из них в одиночку образует команду (в этом примере, конечно, шаг формирования команды можно было бы пропустить, но для полноты не будем).

Если это первое соревнование для каждого из участников, то встаёт вопрос: какие следует использовать начальные данные? Начальное значение для s можно выбирать просто из соображений удобства: какого порядка рейтинги будут удобнее для приложений. Абсолютные значения не важны, важно только соотношение параметров μ, σ, β, и ϵ. Домножение всех этих параметров на любую константу никак не скажется на результатах. Например, Эло нормировал свой рейтинг так, чтобы можно было за начальные данные взять уже имевшиеся в наличии рейтинги Харкнесса. В качестве начального значения для σ имеет смысл взять $\mu/3$; это достаточно много, чтобы иметь возможность в дальнейшем сойтись к каким угодно соотношениям значений. В наших примерах мы будем брать $\beta = 3$, $\epsilon = \frac{1}{2}$.

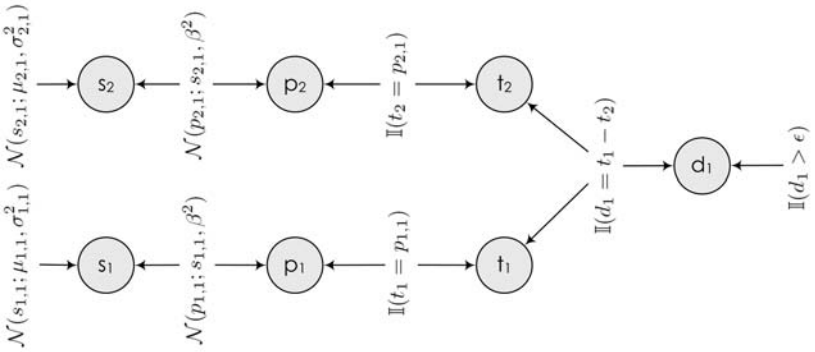

Рис. 1: Граф функций и переменных: матч двух игроков.

Итак, вот подсчёты для первого примера. Граф функций и переменных первого примера показан на рис. 1. Мы будем обозначать нижними индексами $_{i,j}$ данные, относящиеся к j-му игроку i-й команды. Начальные данные:

$$\mu_{1,1} = 100, \quad \sigma_{1,1} = 33.33, \quad \mu_{2,1} = 100, \quad \sigma_{2,1} = 33.33.$$

Начинаем формировать сообщения. Сначала нужно передать сообщение от $s_{i,j}$ к $p_{i,j}$. Как мы уже обсуждали выше,

$$f(p_{i,j} \mid \mu_{i,j}, \sigma_{i,j}) = \int_{-\infty}^{\infty} \mathcal{N}(p_{i,j}; s_{i,j}, \beta^2)\mathcal{N}(s_{i,j}; \mu_{i,j}, \sigma^2_{i,j})ds_{i,j}.$$

Удобные свойства нормального распределения позволяют на данном этапе вычислить интеграл аналитически и получить простую формулу для вычисления сообщения:

$$\int_{-\infty}^{\infty} \mathcal{N}(p_{i,j}; s_{i,j}, \beta^2)\mathcal{N}(s_{i,j}; \mu_{i,j}, \sigma^2_{i,j})ds_{i,j} = \mathcal{N}(p_{i,j}; \mu_{i,j}, \sigma^2_{i,j} + \beta^2).$$

Следующий шаг — это переход от показателей силы игроков $(p_{i,j})$ к силе игры команды. В первом примере каждая команда состоит из одного игрока, и поэтому сила игры команды будет равна попросту

$$p(t_i) = \mathcal{N}(t_i; \mu_{i,1}, \sigma^2_{i,1} + \beta^2).$$

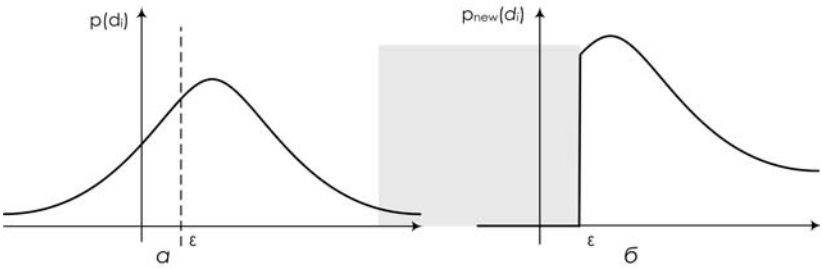

Рис. 2: Изменение распределения в случае победы: а) до; б) после.

Затем нам предстоит сравнить силы двух команд. Мы знаем исход игры и, соответственно, упорядочиваем команды по результату. Перейдём к плотности величины $d_i = t_i - t_{i+1}$. Здесь мы снова воспользуемся тем, что распределения на t_i нормальные, и получим априорную (до обработки непосредственных результатов игры):

$$p(d_i) = \mathcal{N}(d_i, \mu_{i,1} - \mu_{i+1,1}, \sigma_{i,1}^2 + \sigma_{i+1,1}^2 + 2\beta^2).$$

В численном выражении для нашего примера мы получаем

$$p(d_1) = \mathcal{N}(d_1, 0, 84.9448).$$

Теперь (наконец-то!) воспользуемся результатом само́й игры. Предположим, что первый игрок выиграл у второго. Тогда от априорного распределения $p(d_i)$ мы должны отрезать «невозможную часть», а получившееся распределение нормировать.

Остановимся на этом чуть подробнее. Нам нужно оставить только ту часть нормального распределения, которая соответствует значениям $d > \epsilon$, то есть в итоге получить распределение

$$p_{new}(d_1) = \begin{cases} \frac{1}{Z}\mathcal{N}(d_1, 0, 84.9448), & \text{если } d_1 \geq \epsilon, \\ 0, & \text{в противном случае,} \end{cases}$$

$$\text{где нормировка } Z = \int_{\epsilon}^{\infty} \mathcal{N}(d_1; 0, 84.9448) dd_1.$$

См. рис. 2, где мы изобразили этот переход графически. Де-факто же получается, что в нижнем узле, $\mathbb{I}(d_1 > \epsilon)$, стоит не распределе-

ние вероятностей, на которое можно было бы домножить сообщение, а функция Хевисайда с нормировочной константой. На неё и домножают в узле d_1.

В примере с двумя игроками можно было бы честно довести вычисление до верха и получить точные численные значения новых рейтингов. Однако при росте количества игроков в командах и роста количества команд такое распределение в дальнейшем будет очень сложно обработать, поэтому на шаге, передающем это распределение «наверх», его нужно приблизить. Мы приближаем его нормальным распределением; для этого достаточно вычислить математическое ожидание и дисперсию. Для нашего примера получаем (с незначительными округлениями):

$$p_{new}(d_1) = \mathcal{N}(d_1, 38.0798, 808.746426).$$

Теперь поднимемся по графу, начиная с полученного распределения (пользуясь свойствами нормального распределения):

$$\begin{aligned}
p_{new}(t_1) &= \mathcal{N}(t_1, 138.0798, 1928.63536),\\
p_{new}(t_2) &= \mathcal{N}(t_2, 61.9202, 1928.63536),\\
p_{new}(p_{1,1}) &= \mathcal{N}(p_1, 138.0798, 1928.63536),\\
p_{new}(p_{2,1}) &= \mathcal{N}(p_2, 61.9202, 1928.63536),\\
p_{new}(s_{1,1}) &= \mathcal{N}(s_1, 138.0798, 1937.63536),\\
p_{new}(s_{2,1}) &= \mathcal{N}(s_2, 61.9202, 1937.63536).
\end{aligned}$$

Теперь, для того чтобы получить окончательное распределение соответсвующее результату, нам остаётся только вычислить и нормировать произведения

$$\begin{aligned}
p_{new}(s_1)p(s_1) &= \mathcal{N}(p_1, 138.0798, 1937.63536)\mathcal{N}(p_1, 100, 1110.8889),\\
p_{new}(s_2)p(s_2) &= \mathcal{N}(p_1, 61.9202, 1937.63536)\mathcal{N}(p_1, 100, 1110.8889).
\end{aligned}$$

Проделав это, получим новые значения параметров рейтинга игроков (приводим уже округлённые):

$$\mu_{1,1} = 113.876, \quad \sigma_{1,1} = 26.572, \quad \mu_{2,1} = 86.124, \quad \sigma_{2,1} = 26.572.$$

Результаты вполне соответствуют интуитивным ожиданиям: рейтинг победителя стал больше, побеждённого — меньше, а дисперсии уменьшились у обоих.

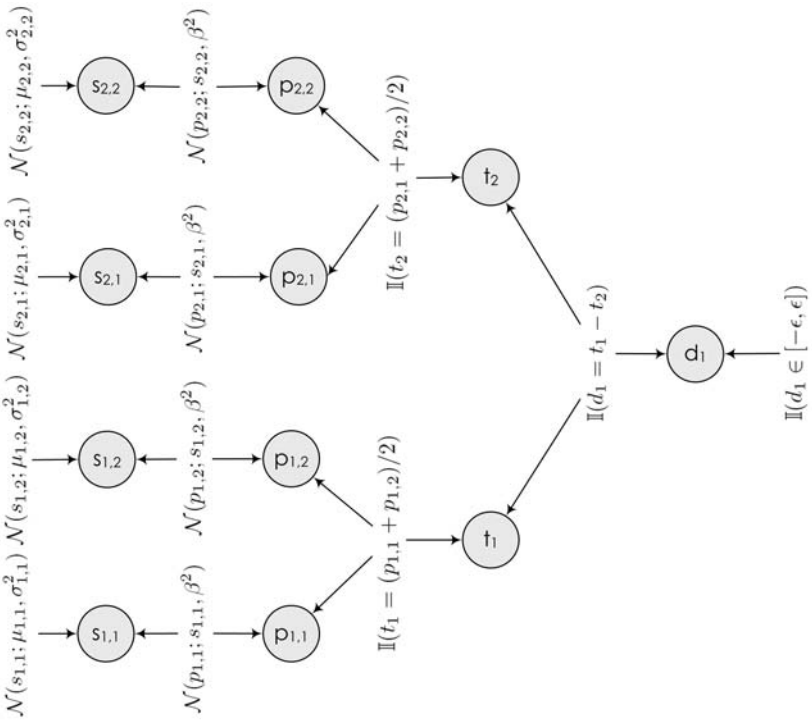

Рис. 3: Граф функций и переменных: две команды по два игрока.

Во втором примере мы рассмотрим ситуацию, в которой в команде несколько игроков (см. рис. 5). В этом случае нужно выбрать, как учитывать переход от силы игроков к силе команды. В этом примере мы будем вычислять силу команды как среднее арифметическое сил игроков; возможны, конечно, и другие подходы. Рассмотрим две команды по два игрока и возьмём в качестве первых игроков в командах данные, полученные в первом примере, а также добавим каждому в команду по одному новому игроку. Тогда исходные данные выглядят как

$$\mu_{1,1} = 113.876, \quad \sigma_{1,1} = 26.572, \quad \mu_{1,2} = 100, \quad \sigma_{1,2} = 33.33,$$
$$\mu_{2,1} = 86.124, \quad \sigma_{2,1} = 26.572, \quad \mu_{2,2} = 100, \quad \sigma_{2,2} = 33.33.$$

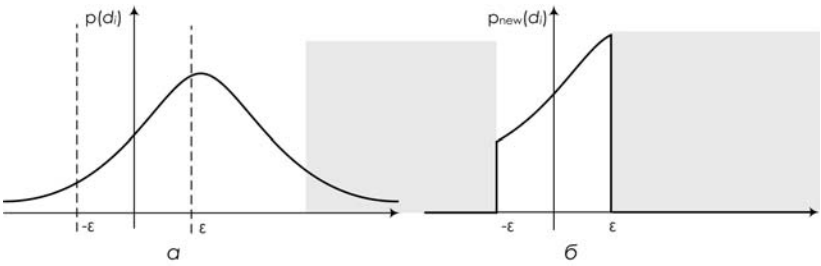

Рис. 4: Изменение распределения в случае ничьей: а) до; б) после.

Пусть результатом стала ничья между командами. Тогда

$$p(p_{i,j}) = \mathcal{N}(p_{i,j}; \mu_{i,j}, \sigma_{i,j}^2 + \beta^2).$$

Для команд в случае двух игроков в каждой получаем

$$p(t_i) = \mathcal{N}\left(p_{i,j}; \frac{1}{2}\left(\mu_{i,1} + \mu_{i,2}\right), \frac{1}{4}\left(\sigma_{i,1}^2 + \beta^2 + \sigma_{i,2}^2 + \beta^2\right)\right).$$

В общем случае среднее есть сумма средних по игрокам, а квадрат дисперсии равен сумме квадратов дисперсий игроков, делённой на квадрат числа игроков. Численно получаем:

$$p(t_1) = \mathcal{N}(t_1; 106.938, 458.74), \quad p(t_2) = \mathcal{N}(t_2; 93.062, 458.74),$$

$$p(d_1) = \mathcal{N}(d_1; 13.876, 917.48).$$

Так как команды сыграли вничью, то на этот раз нужно оставить от распределения $p(d_i)$ только часть от $-\epsilon$ до ϵ (см. рис. 4):

$$p_{new}(d_1) = \begin{cases} \frac{1}{Z}\mathcal{N}(d_1; 13.876, 917.48), & \text{если } -\epsilon \le d_1 \le \epsilon, \\ 0, & \text{в противном случае,} \end{cases}$$

$$\text{где нормировка } Z = \int_{-\epsilon}^{\epsilon} \mathcal{N}(d_1; 13.876, 917.48) dd_1.$$

Приблизим это распределение с помощью нормального:

$$p_{new}(d_1) = \mathcal{N}(d_1; 0.00126, 0.083329).$$

Снова поднимаемся по дереву, пересчитывая плотности:

$$\begin{aligned}
p_{new}(t_1) &= \mathcal{N}(t_1; 93.0633, 458.82333),\\
p_{new}(t_2) &= \mathcal{N}(t_2; 106.9367, 458.82333),\\
p_{new}(p_{1,1}) &= \mathcal{N}(p_{1,1}; 86.12652, 2955.1823),\\
p_{new}(p_{1,2}) &= \mathcal{N}(p_{1,1}; 72.25052, 2550.3646),\\
p_{new}(p_{2,1}) &= \mathcal{N}(p_{2,1}; 113.8735, 2955.1823),\\
p_{new}(p_{2,2}) &= \mathcal{N}(p_{2,2}; 127.7495, 2550.3646),\\
p_{new}(s_{1,1}) &= \mathcal{N}(s_{1,1}; 86.12652, 2964.1823),\\
p_{new}(s_{1,2}) &= \mathcal{N}(s_{1,1}; 72.25052, 2559.3646),\\
p_{new}(s_{2,1}) &= \mathcal{N}(s_{2,1}; 113.8735, 2964.1823),\\
p_{new}(s_{2,2}) &= \mathcal{N}(s_{2,2}; 127.7495, 2559.3646).
\end{aligned}$$

В результате получаются новые значения для игроков:

$$\begin{array}{llll}
\mu_{1,1} = 108.538, & \sigma_{1,1} = 23.88, & \mu_{1,2} = 91.6, & \sigma_{1,2} = 27.833,\\
\mu_{2,1} = 91.462, & \sigma_{2,1} = 23.88, & \mu_{2,2} = 108.399, & \sigma_{2,2} = 27.833.
\end{array}$$

Всё логично: первый и второй игроки как представители априори более сильной команды в результате ничьей потеряли в рейтинге, а третий и четвёртый — заработали.

И последний пример. Пусть четыре «команды» по одному игроку заняли первое, разделили второе и третье и заняли четвёртое место соответственно, то есть вторая и третья сыграли вничью (см. рис. 5). Возьмём исходные данные из предыдущего примера, слегка их переупорядочив:

$$\begin{array}{llll}
\mu_{1,1} = 108.538, & \sigma_{1,1} = 23.88, & \mu_{2,1} = 91.6, & \sigma_{2,1} = 27.833,\\
\mu_{3,1} = 91.462, & \sigma_{3,1} = 23.88, & \mu_{4,1} = 108.399, & \sigma_{4,1} = 27.833.
\end{array}$$

Вычисляя силу команд, как в первом примере, получаем

$$\begin{aligned}
p(t_1) &= \mathcal{N}(t_1; 108.538, 579.2544),\\
p(t_2) &= \mathcal{N}(t_2; 91.6, 579.2544),\\
p(t_3) &= \mathcal{N}(t_3; 91.462, 458.74),\\
p(t_4) &= \mathcal{N}(t_4; 108.833, 783.6759).
\end{aligned}$$

Поступить так, как в предыдущих примерах, мы здесь не сможем. Дело в том, что на плотность величины d_1 будут влиять плотности величин d_2 и d_3, и наоборот. Мы будем учитывать это влияние, итеративно обмениваясь сообщениями. На первом шаге приблизим плотности d_i, как в предыдущем примере. Это исходые значения — $p_1(d_i)$. Введём обозначения:

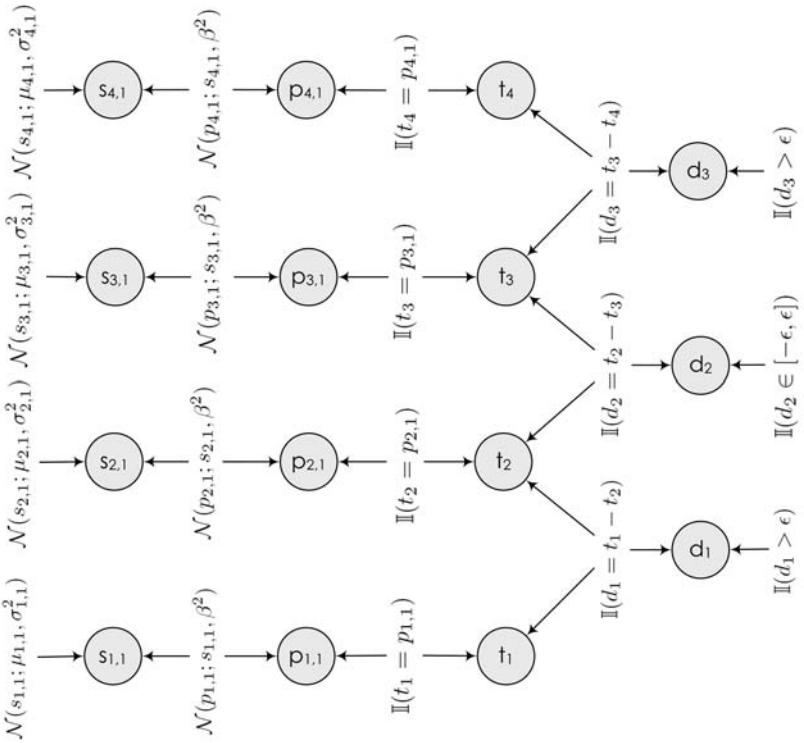

Рис. 5: Граф функций и переменных: матч четырёх игроков.

$$
\begin{aligned}
R_j(t_i) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(d_i - (t_i - t_{i+1}))p(t_{i+1})p_j(d_i)dd_i dt_{i+1},\\
L_j(t_i) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(d_{i-1} - (t_{i-1} - t_i))p(t_{i-1})p_j(d_{i-1})dt_{i-1}dd_{i-1},\\
M_j(d_i) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} R_j(t_{i+1})L_j(t_i)\delta(d_{i-1} - (t_i - t_{i+1}))dt_i dt_{i+1}.
\end{aligned}
$$

Для случаев, когда индексы в формулах выходят за границы множества игроков, будем $R_j(t_n) = 1$ и $L_j(t_1) = 1$. Тогда в случае победы i-той команды над $(i-1)$-ой имеем

$$
p_{j+1}(d_i) = \begin{cases} \frac{1}{Z}M_j(d_i)dd_i, & \text{если } d_i \geq \epsilon, \text{ где } Z = \int_{\epsilon}^{\infty} M_j(d_i)dd_i, \\ 0, & \text{в противном случае,} \end{cases}
$$

а в случае ничьей

$$p_{j+1}(d_i) = \begin{cases} \frac{1}{Z} M_j(d_i) dd_i, & \text{если } d_i \in [-\epsilon, \epsilon], \text{ где } Z = \int_{-\epsilon}^{\epsilon} M_j(d_i) dd_i, \\ 0, & \text{в противном случае.} \end{cases}$$

Полученные плотности приближаем нормальным распределением, делаем следующую итерацию и так до сходимости средних и дисперсий. После этого получаются достаточно точные приближения $p_{new}(d_i)$. Затем пересчитываем $p_{new}(t_i) = R(t_i)L(t_i)$, нормируем полученные значения и, как и в предыдущих примерах, идём вверх по дереву до начала. Итогом подобных вычислений станут следующие результаты:

$$\mu_{1,1} = 117.37, \quad \sigma_{1,1} = 17.198, \quad \mu_{2,1} = 96.48, \quad \sigma_{2,1} = 14.242,$$
$$\mu_{3,1} = 96.462, \quad \sigma_{3,1} = 14.185, \quad \mu_{4,1} = 84.399, \quad \sigma_{4,1} = 17.516.$$

Опять же, результаты выглядят вполне логично.

6. Заключение. В этой работе мы рассмотрели, от простого к сложному, три метода построения рейтинг-систем, каждый из которых основан на том или ином варианте модели Брэдли-Терри. Желающим организовать свою собственную рейтинг-систему или применить модели Брэдли-Терри в других областях мы рекомендуем в первую очередь реалистично оценить, насколько общий случай рейтинга нужен; система TrueSkill, на данный момент одна из наиболее общих постановок, требует значительных вычислительных затрат, и реализовать её нелегко, а рейтинг Эло — весьма несложно. Но в любом случае рейтинг-система сводится к той или иной задаче байесовского вывода; решив её, можно обучить рейтинги участников.

Литература

1. *Agresti A.* Categorical Data Analysis. New York: Wiley, 1990.
2. *Bradley R. A., Terry M. E.* Rank Analysis of Incomplete Block Designs. I. The Method of Paired Comparisons // Biometrika. 1952. Vol. 39. P. 324–245.
3. *Coulom R.* Computing Elo Ratings of Move Patterns in the Game of Go // ICGA Journal. December 2007. Vol. 30, N. 4. P. 198–208.
4. *Elo A.* The Ratings of Chess Players: Past and Present. New York: Arco, 1978.
5. *Gelman A., Carlin J. B., Stern H. S., Rubin D. B.* Bayesian Data Analysis. CRC Press, 2003.

6. *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.

7. *Hunter D. R.* MM Algorithms for Generalized Bradley-Terry Models // The Annals of Statistics. 2004. Vol. 32, N. 1. P. 384–406.

8. *Luce R. D.* Individual Choice Behaviour. New York: Wiley, 1959.

9. *MacKay D. J.* Information Theory, Inference and Learning Algorithms. Cambridge University Press, 2003.

10. *Marden J. I.* Analyzing and Modeling Rank Data. London: Chapman and Hall, 1995.

11. *Menke J. E., Martinez T. R.* A Bradley–Terry artificial neural network model for individual ratings in group competitions // Neural Computing and Applications. 2008. Vol. 17, N. 2. P. 175–186.

12. *Plackett R. L.* The Analysis of Permutations // Journal of Applied Statistics. 1975. Vol. 24. P. 193–202.

13. *Rao P. V., Kupper L. L.* Ties in paired–comparison experiments: a generalization of the Bradley-Terry model // Journal of the American Statistical Association. 1967. Vol. 62. P. 194–204.

14. *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.

15. *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.

16. *Леман Э.* Теория точечного оценивания. М., Наука, Физматлит, 1991.

17. *Тулупьев А. Л., Николенко С. И., Сироткин А. В.* Байесовские сети: логико-вероятностный подход. СПб.: Наука, 2006. 608 с.

Николенко Сергей Игоревич — к.ф.-м.н.; младший научный сотрудник Учреждения Российской академии наук Санкт-Петербургского отделения Математического института им. В. А. Стеклова РАН (ПОМИ РАН); область научных интересов: теоретическая информатика, криптография, высшая алгебра, машинное обучение, распознавание речи, системы представления знаний. Число научных публикаций — 43. `sergey@logic.pdmi.ras.ru`, ПОМИ РАН, 191023, Санкт-Петербург, наб. р. Фонтанка, 27, тел. +7(812)3124058.

Сироткин Александр Владимирович — младший научный сотрудник Учреждения Российской академии наук Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН); область научных интересов: байесовский вывод, байесовские сети, машинное обучение. Число научных публикаций — 40. `avs@iias.spb.su`, СПИИРАН, 199178, Санкт-Петербург, 14 линия В.О., 39, тел. +7(812)3283411. Научный руководитель — А. Л. Тулупьев.

Поддержка исследований. В публикации представлены результаты исследований, поддержанных грантом РФФИ 09-01-00861-а (рук. А. Л. Тулупьев).