УДК 519.853.62

Вокруг степенного закона распределения компонент вектора PageRank. Часть 2. Модель Бакли–Остгуса, проверка закона для этой модели и устройство реальных поисковых систем*

А.В. Гасников[1,2], П.Е. Двуреченский[2,3], М.Е. Жуковский[1,4], С.В. Ким[6], С.С. Плаунов[5], Д.А. Смирнов[5], Ф.А. Носков[1]

[1]Московский физико-технический институт, Институтский пер., 9, Долгопрудный, Московская обл., 141700

[2]Институт проблем передачи информации им. А.А. Харкевича Российской академии наук, Большой Каретный пер., 19, строение 1, Москва, 127051

[3]Институт прикладного анализа и стохастики им. К. Вейерштрасса, Мореншрассе, 39, Берлин, Германия, 10117

[4]Общество с ограниченной ответственностью "Яндекс", ул. Льва Толстого, 16, Москва, 119034

[5]Государственное бюджетное образовательное учреждение "Физматшкола № 2007", ул. Горчакова, 9, корп. 1, Москва, 117042

[6]Национальный исследовательский университет "Высшая школа экономики", ул. Мясницкая, д. 20, Москва, 101000

E-mails: gasnikov.av@mipt.ru (Гасников А.В.), pavel.dvurechensky@wias-berlin.de (Двуреченский П.Е.), zhukmax@yandex-team.ru (Жуковский М.Е.), kims230599@gmail.com (Ким С.В.), styopa.plaunov@gmail.com (Плаунов С.С.), Daniil.Smirnov.2000@inbox.ru (Смирнов Д.А.), fedor.noskov.99@mail.ru (Носков Ф.А.)

Гасников А.В., Двуреченский П.Е., Жуковский М.Е., Ким С.В., Плаунов С.С., Смирнов Д.А., Носков Ф.А. Вокруг степенного закона распределения компонент вектора PageRank. Часть 2. Модель Бакли–Остгуса, проверка закона для этой модели и устройство реальных поисковых систем // Сиб. журн. вычисл. математики / РАН. Сиб. отд-ние. — Новосибирск, 2018. — Т. 21, № 1. — С. 23–45.

Данная статья является продолжением статьи [13]. В этой, второй части, работы рассматривается модель Бакли–Остгуса формирования сети Интернет. Для сетей, порожденных этой моделью, проводятся численные эксперименты по вычислению вектора PageRank. Обнаруживается степенной закон распределения компонент этого вектора. Обсуждаются вычислительные аспекты этой модели в контексте описанных в первой статье [13] численных способов поиска вектора PageRank. Описаны более общая модель ранжирования web-страниц и подходы к решению задачи оптимизации, возникающей при обучении этой модели.

DOI: 10.15372/SJNM20180102

Ключевые слова: *марковская цепь, эргодическая теорема, мультиномиальное распределение, концентрация меры, оценка максимального правдоподобия, Google problem, градиентный спуск, автоматическое дифференцирование, степенной закон распределения.*

Gasnikov A., Dvurechensky P., Zhukovskii M., Kim S., Plaunov S., Smirnov D., Noskov F. About the power law of the PageRank vector distribution. Part 2. Backley–Osthus model, power law verification for this model and setup of real search engines // Siberian J. Num. Math. / Sib. Branch of Russ. Acad. of Sci. — Novosibirsk, 2018. — Vol. 21, № 1. — P. 23–45.

*Исследование в пункте 3 частично поддержано грантом Президента РФ МК-1806.2017.9. Исследование А.В. Гасникова и П.Е. Двуреченского в пункте 4 выполнено в ИППИ РАН за счет гранта Российского научного фонда (проект № 14-50-00150).

In the second part of this paper, we consider the Buckley–Osthus model for the formation of a web-graph. For the networks generated according to this model, we numerically calculate the PageRank vector. We show that the components of this vector are distributed according to the power law. We also discuss the computational aspects of this model with respect to different numerical methods for the calculation of the PageRank vector, presented in the first part of the paper. Finally, we describe a general model for the web-page ranking and some approaches to solve the optimization problem arising when learning this model.

Keywords: *Markov chain, ergodic theorem, multinomial distribution, measure concentration, maximum likelihood estimate, Google problem, gradient descent, automatic differentiation, power law distribution.*

1. Введение

Данная работа, состоящая из двух статей, посвящена нескольким фундаментальным вопросам, связанным с эргодической теоремой для марковских процессов, с методами Монте-Карло, явлением концентрации меры, понятием равновесия макросистемы, основной теоремой математической статистики (теорема Фишера) о свойствах оценки максимального правдоподобия, ролью степенных законов распределения, машинным обучением, невыпуклой оптимизацией, автоматическим дифференцированием и рядом смежных вопросов. Все темы и вопросы обсуждаются на одном примере — Google problem, также известной как задача поиска вектора PageRank. Это позволяет увидеть как математические результаты, входящие, по нашему мнению, в "золотой фонд" современной математики, соотносятся друг с другом. Умение эффективно и многократно находить вектор PageRank, например, позволяет поисковым системам ранжировать web-страницы в ответ на пользовательский запрос [27, 34, 35, 42, 52, 61]. Схожие постановки задач возникают в моделях поиска консенсуса [1]. Отметим также книгу [33], которая близка по содержанию к настоящей работе. Заметная часть изложенных в статье результатов была специально адаптирована для максимально широкой аудитории. В связи с этим важную роль играют ссылки на литературу, которая позволяет восстановить опущенные в статье детали.

В первой статье [13] предложена оригинальная интерпретация вектора PageRank, согласно которой осуществляется ранжирование web-страниц. Важно отметить, что эта интерпретация позволяет построить параллельный алгоритм для поиска вектора PageRank. В статье также описаны другие численные методы поиска этого вектора. В этой, второй, статье проведена проверка степенного закона убывания компонент вектора PageRank для web-графа, построенного по наиболее популярной сейчас модели роста интернета — модели Бакли–Остгуса [27]. Также обсуждаются вычислительные аспекты ранжирования web-страниц на практике. Очень ценным при этом оказался опыт совместной работы с сотрудниками компании Яндекс [34].

2. Модель Бакли–Остгуса роста сети Интернет и степенные законы

С точки зрения практических нужд (например для разработки алгоритмов борьбы со спамом) модели web-графа оказываются зачастую более востребованными, чем сами web-графы, ввиду того, что web-графы имеют слишком большие для анализа размеры. Кроме того, с помощью моделей можно исследовать различные закономерности, присущие web-графу. Так, Интернету и многим социальным сетям присущи определенные хорошо изученные закономерности: наличие гигантской компоненты, правило пяти рукопожатий, степенной закон для распределения степеней вершин, специальные свойства

кластерных коэффициентов и т. д. [19, 27, 49]. Хотелось бы предложить такую модель формирования этих сетей, которая бы объясняла все эти закономерности. Построив такую модель, с помощью фундаментальной науки мы можем открывать новые свойства, присущие изучаемой сети, исследуя лишь свойства выбранной модели. Такие исследования позволяют в дальнейшем использовать полученные результаты при разработке алгоритмов для реальных сетей, в том числе Интернета.

Одной из лучших на текущий момент моделей web-графа считается модель Бакли–Остгуса (см., например, [27]). Именно ее мы и будем рассматривать. Новым свойством, которое мы хотим показать, будет степенной закон распределения компонент вектора PageRank, посчитанного для графа, сгенерированного по этой модели. Далее мы описываем модель Бакли–Остгуса и выводим[1] степенной закон распределения степеней вершин графа, построенного по этой модели. Именно этот закон приводит к степенному закону распределения компонент вектора PageRank.

Более подробно о том, почему во многих реальных ситуациях так часто возникают степенные законы, можно прочитать в обзорах [24, 53, 58, 59].

Итак, рассмотрим следующую модель роста сети.

База индукции. Сначала имеется всего одна вершина, которая ссылается сама на себя (вершина с петлей).

Шаг индукции. Предположим, что уже имеется некоторый ориентированный граф. Добавим в граф новую вершину. С вероятностью $\beta > 0$ из этой вершины проведем ребро равновероятно в одну из существующих вершин, а с вероятностью $1-\beta$ из этой вершины проводится ребро в одну из существующих вершин не равновероятно, а с вероятностями, пропорциональными входящим степеням вершин[2] — *правило предпочтительного присоединения* (preferential attachment).

Другими словами, если уже построен граф из $n-1$ вершины, то новая n-я вершина сошлется на вершину $i = 1, \ldots, n-1$ с вероятностью

$$\frac{\mathrm{Indeg}_{n-1}(i) + a}{(n-1)(a+1)},$$

где $\mathrm{Indeg}_{n-1}(i)$ — входящая степень вершины i в графе, построенном на шаге $n-1$. Параметры β и a связаны следующим образом:

$$a = \frac{\beta}{1-\beta}.$$

При $a = 1$ получается известная модель Боллобаша–Риордана (см., например, [27]). Интернету наилучшим образом соответствует значение $a = 0.277$ [27].

Далее вводится число m — среднее число web-страниц на одном сайте, и каждая группа web-страниц с номерами $km+1, \ldots, (k+1)m$ объединяется в один сайт. При этом все ссылки, имеющиеся между web-страницами, наследуются содержащими их сайтами, т. е. получается, что с одного сайта на другой[3] может быть несколько ссылок. Пусть, скажем, получилось, что для заданной пары сайтов таких (одинаковых) ссылок оказалось $l \leq m$,

[1] Не строго, а в так называемом *термодинамическом пределе*. Отметим также, что рассматриваемая модель немного отличается от описанной в [27] модели Бакли–Остгуса, в которой новая вершина может сослаться и сама на себя. Однако это никак не отразится на основных статистических свойствах построенного по модели случайного графа.

[2] Выходящая степень всех вершин одинакова и равна 1.

[3] Впрочем, сайты могут совпадать, а внутри одного сайта web-страницы также могут друг на друга ссылаться.

тогда мы превращаем их в одну ссылку, но с весом (вероятностью перехода) l/m. Именно для так построенного взвешенного ориентированного графа мы будем изучать закон распределения компонент вектора PageRank.

К сожалению, строго доказать, что имеет место степенной закон распределения компонент вектора PageRank в этой модели, насколько нам известно, пока никому не удалось. Имеется только один специальный результат на эту тему, касающийся модели, близкой к модели Боллобаша–Риордана [30]. В данной работе мы ограничимся только численными экспериментами. Однако, чтобы у читателей появилась некоторая интуиция, почему такой закон может иметь место в данном случае, мы приведем далее некоторые аргументы.

Сначала, следуя работе [53], установим степенной закон распределения входящих вершин в модели Бакли–Остгуса. При этом ограничимся случаем $m = 1$. Обозначим через $X_k(t)$ число вершин с входящей степенью k в момент времени t, т. е. когда в графе имеется всего t вершин. Заметим, что по определению

$$t = \sum_{k\geq 0} X_k(t) = \sum_{k\geq 1} kX_k(t) = \sum_{k\geq 0} kX_k(t).$$

Поэтому для $k \geq 1$ вероятность того, что $X_k(t)$ увеличится на единицу при переходе на следующий шаг $t \to t+1$ по формуле полной вероятности, равна

$$\beta\frac{X_{k-1}(t)}{t} + (1-\beta)\frac{(k-1)X_{k-1}(t)}{t}.$$

Аналогично, для $k \geq 1$ вероятность того, что $X_k(t)$ уменьшится на единицу при переходе на следующий шаг $t \to t+1$:

$$\beta\frac{X_k(t)}{t} + (1-\beta)\frac{kX_k(t)}{t}.$$

Таким образом, "ожидаемое" приращение $\Delta X_k(t) = X_k(t+1) - X_k(t)$ за $\Delta t = (t+1) - t = 1$ будет[4]

$$\frac{\Delta X_k(t)}{\Delta t} = \beta\frac{(X_{k-1}(t) - X_k(t))}{t} + (1-\beta)\frac{(k-1)X_{k-1}(t) - kX_k(t)}{t}. \tag{1}$$

Для $X_0(t)$ уравнение, аналогичное (1), будет иметь вид

$$\frac{\Delta X_0(t)}{\Delta t} = 1 - \beta\frac{X_0(t)}{t}. \tag{2}$$

К сожалению, соотношения (1), (2) не есть точные уравнения, описывающие то, как меняется $X_k(t)$, хотя бы потому, что изменение $X_k(t)$ происходит случайно. Динамика же (1), (2) полностью детерминированная. Однако для больших значений t, когда наблюдается концентрация случайных величин $X_k(t)$ вокруг своих математических ожиданий,

[4]Корректная запись

$$E_{X_{k+1}(t)}\left[\frac{\Delta X_k(t)}{\Delta t}\,\middle|\, X_0(t), \ldots, X_k(t)\right] = \beta\frac{(X_{k-1}(t) - X_k(t))}{t} + (1-\beta)\frac{(k-1)X_{k-1}(t) - kX_k(t)}{t}.$$

Беря от обеих частей математическое ожидание $E_{X_0(t),\ldots,X_k(t)}[\,\cdot\,]$, получим

$$E\left[\frac{\Delta X_k(t)}{\Delta t}\right] = \beta\frac{(E[X_{k-1}(t)] - E[X_k(t)])}{t} + (1-\beta)\frac{(k-1)E[X_{k-1}(t)] - kE[X_k(t)]}{t}.$$

реальная динамика поведения $X_k(t)$ и динамика поведения средних значений $X_k(t)$ становятся близкими[5] — вариация на тему *теоремы Куртца* [41]. Таким образом, на систему (1), (2) можно смотреть как на динамику средних значений, вокруг которых плотно сконцентрированы реальные значения. Под плотной концентрацией имеется в виду, что разброс значений величины контролируется квадратным корнем из ее среднего значения (см. [13, п. 5]).

Будем искать решение системы (1), (2) на больших временах ($t \to \infty$) в виде $X_k(t) \sim c_k t$ (иногда такого вида режимы называют *промежуточными асимптотиками* [3]). Подставляя это выражение в формулы (1), (2), получим

$$c_0 = \frac{1}{1+\beta}, \qquad \frac{c_k}{c_{k-1}} = 1 - \frac{2-\beta}{1+\beta+k\,(1-\beta)} \simeq 1 - \left(\frac{2-\beta}{1-\beta}\right)\frac{1}{k}.$$

Откуда получаем следующий ***степенной закон***:

$$c_k \sim k^{-\frac{2-\beta}{1-\beta}} = k^{-2-a}. \tag{3}$$

Заметим, что если построить на основе (3) *ранговый закон* распределения вершин по их входящим степеням, т. е. отранжировать вершины по входящей степени, начиная с вершины с самой высокой входящей степенью, то также получим степенной закон [24]:

$$\mathrm{in\,deg}(r) \sim r^{-1-\beta}. \tag{4}$$

Действительно, обозначив для краткости $\mathrm{in\,deg}(r)$ через x, получим, что нам нужно найти зависимость $x(r)$, если из формулы (3) известно, что

$$\frac{dr(x)}{dx} \sim -x^{-\frac{2-\beta}{1-\beta}} = -x^{-1-\frac{1}{1-\beta}},$$

где зависимость $r(x)$ получается из зависимости $x(r)$ как решение уравнения $x(r) = x$. Остается только подставить сюда и проверить приведенное соотношение (4). Заметим, что именно ранговый закон распределения компонент вектора PageRank (удовлетворяющего (5) с $\delta > 0$) мы будем численно проверять в п. 3.

Перейдем теперь к пояснению того, почему может иметь место степенной закон распределения компонент вектора PageRank. Для этого предположим, что матрица P имеет вид

$$P \sim \begin{bmatrix} 1^{-\lambda} & 2^{-\lambda} & 3^{-\lambda} & 4^{-\lambda} & 5^{-\lambda} & \dots \\ 1^{-\lambda} & 2^{-\lambda} & 3^{-\lambda} & 4^{-\lambda} & 5^{-\lambda} & \dots \\ 1^{-\lambda} & 2^{-\lambda} & 3^{-\lambda} & 4^{-\lambda} & 5^{-\lambda} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}.$$

Такой вид матрицы означает, что для каждого сайта имеет место точный (т. е. не в вероятностном смысле) степенной закон распределения выходящих степеней вершин, имеющий одинаковый вид для всех сайтов. Конечно, это намного более сильное предположение, чем то, что мы выше получили для модели Бакли–Остгуса. Тогда для выписанной матрицы P выполняются условия единственности вектора PageRank ν, определяющегося формулой (2) (см. [13, п. 2]). Более того, этот вектор неизбежно должен совпадать со строчкой (не важно какой именно — они одинаковые) матрицы P:

$$\nu_k \sim k^{-\lambda}.$$

[5] Последняя динамика уже является детерминированной динамикой.

Но это и означает, что имеет место степенной закон распределения компонент вектора PageRank. Разумеется, проведенные рассуждения ни в какой степени нельзя считать доказательством. Тем не менее, мы надеемся, что некоторую интуицию эти рассуждения читателям все-таки смогли дать.

В контексте написанного выше хотелось бы отметить, что подобно системе (1), (2) можно записать *динамику средних* (говорят также *квазисредних* [7, 12]) и для макросистемы блуждающих по web-графу людей из [13, п. 5]. А именно, предположим, что людей достаточно много и что каждый человек в любом промежутке времени $[t, t + \Delta t)$ независимо от остальных с вероятностью, по порядку равной Δt, совершает переход по одной из случайно выбранных согласно матрице P ссылок. Обозначив через $c_k(t)$ долю людей, находящихся в момент времени t на web-странице с номером k, получим следующую систему:

$$\frac{\Delta c^{\top}(t)}{\Delta t} = c^{\top}(t) P - c^{\top}(t). \tag{5}$$

Формула (5) подтверждает вывод о том, что вектор PageRank ν, удовлетворяющий системе (2), действительно можно понимать как равновесие макросистемы. В самом деле, если существует предел $\nu = \lim_{t\to\infty} c(t)$, то из (5) следует, что этот предел должен удовлетворять (2). Здесь, в отличие от [13, п. 2], предел всегда существует. Но также, как и в [13, п. 2], может зависеть от начального условия. Для того, чтобы предел не зависел от начального условия и был единственным, нужно сделать предположение о наличии в графе "Красной площади".

Имеется глубокая связь между приведенной выше схемой рассуждений и общими моделями макросистем, которые с точки зрения математики можно понимать как разнообразные модели стохастической химической кинетики [2, 6, 7, 9, 10, 12, 18, 20, 62]. В частности, система (5) соответствует *закону действующих масс Гульдберга–Вааге* [4, 12, 18, 20, 25]. При этом важно подчеркнуть, что возможность осуществлять описанный выше канонический скейлинг, по сути заключающийся в замене концентраций их средними значениями, обоснована теоремой Куртца [20, 41] (в том числе и для нелинейных систем, появляющихся, когда имеются не только унарные реакции, как в примере с PageRank'ом) только на конечных отрезках времени. Для бесконечного отрезка[6] требуются дополнительные оговорки, например выполнение условия детального баланса и его обобщений [2, 4, 6, 7, 12, 20]. Хорошим примером тут является известная модель "хищник–жертва" [5], приводящая к системе Лотки–Вольтерра [26] лишь на конечных отрезках времени. На бесконечном отрезке времени либо сначала все "зайцы" будут съедены "волками", после чего все "волки" погибнут от голода, либо сначала все "волки" погибнут из-за нехватки пищи ("зайцев"), после чего "зайцы" неограниченно расплодятся [6, 12, 16]. И в том и в другом случае такая асимптотика никак не соответствует незатухающим (нелинейным) колебаниям численностей "волков" и "зайцев", которые предписывает решение системы Лотки–Вольтерра. Другими словами, в общем случае использованные нами предельные переходы (по времени и числу агентов) не перестановочны! Рассуждения в [13, п. 5] соответствуют следующему порядку предельных переходов: сначала $t \to \infty$ (выходим на инвариантную меру/стационарное распределение), потом $N \to \infty$ (концентрируемся вокруг наиболее вероятного макросостояния инвариантной меры); а рассуждения этого пункта таковы: сначала $N \to \infty$ (переходим на описание макросистемы на языке концентраций, устраняя случайность с помощью законов больших чисел), потом $t \to \infty$ (исследуем аттрактор полученной при скейлинге детерминированной, т. е.

[6] А именно эта ситуация нам наиболее интересна, поскольку, чтобы выйти на равновесие, как правило, необходимо перейти к пределу $t \to \infty$.

не стохастической, системы). Для примера макросистемы из [13, п. 5] получается один и тот же результат. Более того, если посмотреть на то, как именно концентрируется инвариантная мера для этого примера, то получим, что *концентрация экспоненциальная* [2, 6, 12, 28]:

$$P\left(r=k\right)=\frac{N!}{k_1!\cdots k_n!}\nu_1^{k_1}\cdots\nu_n^{k_n}\simeq\exp\left(-N\cdot\mathrm{KL}\left(k/N,\nu\right)\right),$$

где *функция действия* (*функция Санова*) $\mathrm{KL}(x,y)=-\sum_{k=1}^n x_k\ln\left(x_k/y_k\right)$. В других контекстах функцию KL чаще называют *дивергенцией Кульбака–Лейблера* или просто *энтропией*.[7] При этом функция $\mathrm{KL}(c\left(t\right),\nu)$, как функция t, монотонно убывает с ростом t на траекториях системы (5), т. е. является *функцией Ляпунова*. Оказывается, этот факт[8] имеет место и при намного более общих условиях [2, 6, 12, 20]. Точнее говоря, сам факт о том, что *функция, характеризующая экспоненциальную концентрацию инвариантной меры, будет функцией Ляпунова динамической системы, полученной в результате скейлинга из марковского процесса, породившего исследуемую инвариантную меру*, имеет место всегда, а вот то, что именно такая KL-функция будет возникать, соответствует макросистемам, удовлетворяющим обобщенному условию детального баланса (условию Штюкельберга–Батищевой–Пирогова [2, 4, 6, 10, 11, 20]), и только таким макросистемам [12].

В заключение этого пункта заметим, что подобно [13, п. 5] можно получить закон (3) в более точных вероятностных категориях. Хотя это можно сделать вполне элементарными комбинаторными средствами (см., например, [46]), тем не менее, соответствующие выкладки оказываются достаточно громоздкие, поэтому мы не приводим их здесь.

3. Численная проверка степенного закона распределения компонент вектора PageRank

В численных экспериментах использовалась модель Бакли–Остгуса, описанная в предыдущем пункте, с разными значениями параметра $a\geq 0$, в которой $m=10$ (среднее число web-страниц на одном сайте), а число сайтов, которое будем обозначать n, было равно $n=10^5$. Эксперименты проводились на нескольких компьютерах с разными операционными системами (частота процессора 2–3 ГГц, размер оперативной памяти 8–16 Гб), все обсуждаемые алгоритмы были реализован на языке Python [29] без привлечения модулей на других языках. Основные временные затраты были связаны с подготовкой web-графа согласно модели, описанной в предыдущем пункте.

Сначала сравнивались методы из таблицы (и не только). Численные эксперименты [29] вполне определенно продемонстрировали, что для web-графа, построенного по модели Бакли–Остгуса с указанными выше параметрами при желаемой точности $\varepsilon\ll 10^{-5}$ (см. столбец “Цель” в таблице), метод простой итерации (МПИ) доминирует над всеми остальными подходами к расчету вектора PageRank по всем разумным критериям. В приведенных далее численных экспериментах выбиралась точность $\varepsilon\simeq 10^{-7}$, т. е. компоненты вектора PageRank ν восстанавливались следующим образом: $\left\|\tilde{p}_T-\nu\right\|_1\leq 10^{-7}$. Время работы МПИ (при $a=1$) составляло порядка 5 минут. С ростом параметра a это время немного увеличивалось.

[7] Отсюда, по-видимому, и пошло, что “равновесие следует искать из *принципа максимума энтропии*” [2, 4, 6, 12, 20, 50, 51].

[8] Известный из курса термодинамики/статистической физики как *H-теорема Больцмана*.

Таблица. Сравнение методов решения задачи поиска вектора PageRank (см. [13])

Метод	**Сложность**	**Цель**
Метод простой итерации (МПИ)	$\frac{sn}{\alpha}\ln\left(\frac{2}{\varepsilon}\right)$	$\left\|\tilde{p}_T-\nu\right\|_1$
Поляка–Трембы	$\frac{2sn}{\varepsilon}$	$\left\|P^\top\tilde{p}_T-\tilde{p}_T\right\|_1$
Д. Спилмана	$C\left(n+\frac{s^2}{\alpha\varepsilon}\ln\left(\frac{1}{\varepsilon}\right)\right)$	$\left\|\tilde{p}_T-\nu\right\|_\infty$
Markov Chain Monte Carlo (MCMC)	$C\left(n+\frac{\log_2 n\cdot\ln(n/\sigma)}{\alpha\varepsilon^2}\right)$	$\left\|\tilde{p}_T-\nu\right\|_2$
Вариация метода условного градиента	$C\left(n+\frac{s^2\ln n}{\varepsilon^2}\right)$	$\left\|P^\top\tilde{p}_T-\tilde{p}_T\right\|_2$

С помощью МПИ была проверена гипотеза о том, что имеет место степенной закон распределения компонент вектора PageRank (рисунок 1). При построении графика на рис. 1 компоненты вектора PageRank предварительно были отсортированы по убыванию, однако, если такую сортировку не производить, то вид графика практически не изменялся. Аналогичные графики получались и для других реализаций случайного web-графа, полученного по модели Бакли–Остгуса, при других значениях параметра $a \geq 0$.

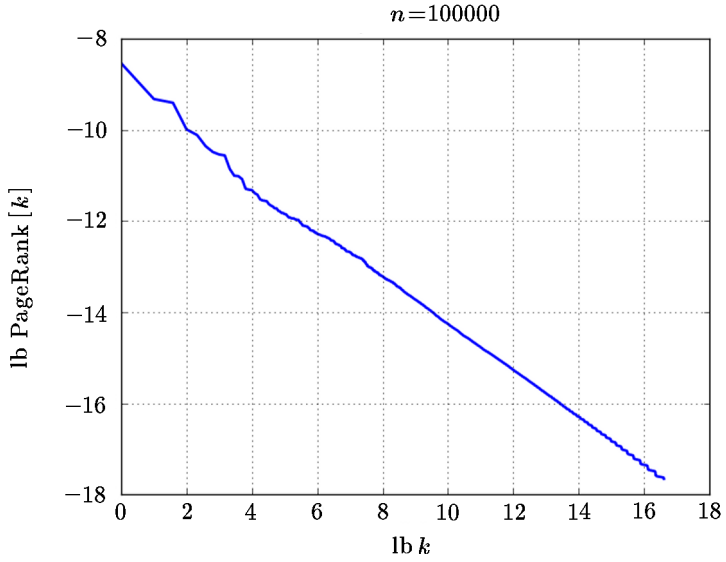

Рис. 1. Зависимость отсортированных по убыванию компонент $\log_2\nu_k$ от $\log_2 k$ при $a=1$

На рисунках 2 и 3 через $g(a)<0$ обозначен показатель степени в предполагаемом степенном законе $\nu_k \sim k^{g(a)}$. Этот показатель определялся по методу наименьших квадратов [48] по данным, отображенным на рис. 1. Для каждого значения $a \geq 0$ по модели Бакли–Остгуса независимо генерировалось 15 web-графов, исходя из разброса посчитанных по этим web-графам значений $g(a)$ около среднего значения, строились приведенные на рис. 2 и рис. 3 три кривые (среднее значение, в легенде обозначенное “average”, и крайние значения, в легенде обозначенные “range”). Отметим, что при увеличении параметра m в несколько раз $g(a)$ изменялся на несколько тысячных, т. е. можно считать, что $g(a)$ не зависит от выбора m. Однако с увеличением m время расчета вектора PageRank также увеличивалось. Отметим также, что при $n=10^4$ получались аналогичные графики только с немного большим разбросом.

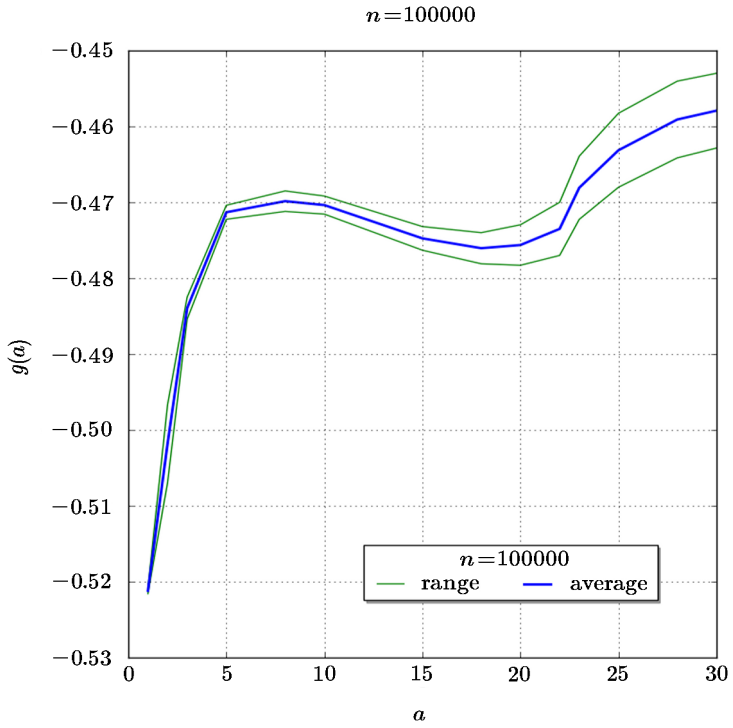

Рис. 2. Зависимость $g(a)$, $a \in [0, 30]$, в предполагаемом законе $\nu_k \sim k^{g(a)}$

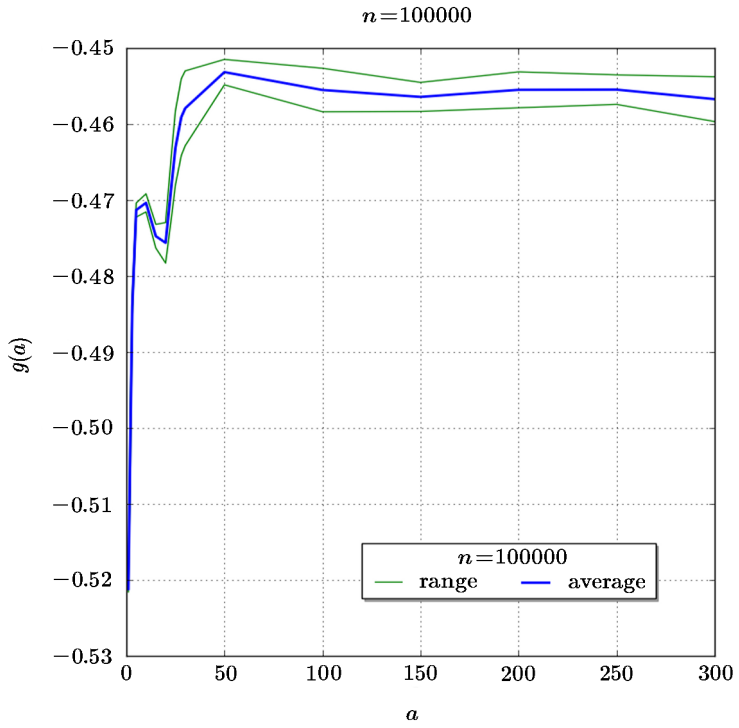

Рис. 3. Зависимость $g(a)$, $a \in [0, 300]$, в предполагаемом законе $\nu_k \sim k^{g(a)}$

4. Немного об устройстве реальных поисковых систем

Концепция вектора PageRank, рассматриваемая нами до этого момента, была нацелена на решение задачи ранжирования web-страниц по запросам пользователей. Заметим, что в данной концепции никак не используется информация о конкретном запросе. Тем не менее в этом пункте будет описан один из существующих способов использования именно классической концепции вектора PageRank для решения реальных задач ранжирования web-страниц.

Может показаться, что в поисковой системе либо есть огромная база соответствий между запросами q и web-страницами w, либо есть огромный штат сотрудников, которые за долю секунды успевают просмотреть миллиарды страниц и выбрать нужные. Конечно же, обе гипотезы далеки от истины. Тем не менее некоторая доля истины есть в обеих. Разумеется, в современных поисковых системах (таких как Яндекс и Google) есть постоянно обновляемая огромная база соответствий страниц запросам. Эти соответствия выражаются векторами *признаков* $y(q,w)$. Каждый элемент такого вектора $y_k(q,w)$ вычисляется как мера обладания пары (q,w) (запрос, web-страница) некоторым, как правило, интуитивным свойством k, которым обязана такая пара обладать, если страница должна быть показана в выдаче по запросу (говорят, что страница *релевантна* запросу). К таким свойствам, например, относят "свежесть", "соответствие тематике" и т. п. Для хорошей работы поисковой системы необходимо иметь много признаков, поэтому векторы признаков в огромных компаниях, занимающихся поиском, имеют, как правило, большую размерность (от 500 и выше), т. е. $k = 1, \ldots, l$; $l \sim 10^3$. Кроме того, в компаниях, предлагающих пользователям качественный поиск, работают специально обученные люди, называемые асессорами. Эти специалисты выставляют оценки некоторым парам (запрос, web-страница) в соответствии со степенью релевантности страницы запросу. Оценки парам выставляются по пятибалльной шкале (от "0" за наименьшую степень релевантности до "4" за наибольшую степень релевантности).

Если задать некоторый запрос поисковой системе, то это еще не значит, что, во-первых, в базе поисковой системы такой запрос имеется, а во-вторых, что, если такой запрос в базе и имеется, то найдутся web-страницы, размеченные асессорами по запросу. Вообще говоря, вероятность того, что два последних события выполнены, близка к нулю. В этой связи работники поисковых систем прибегают к решению специальной задачи *машинного обучения* [48, 63], которую можно сформулировать следующим образом [34, 36, 37, 38].

Прежде всего опишем параметрическую модель, положенную в основу ранжирования. Пусть $x \in \mathrm{R}^d$, $d \simeq 2l \sim 10^3$ — вектор параметров, задающих модель. Для каждого запроса q будем считать, что ранжирование осуществляется согласно вектору $p = p(q,x)$, где i-я координата вектора p — значение функции ранжирования для i-го документа, размеченного (оцененного асессорами) по запросу q. Таким образом, ранжирование можно осуществить, если известен вектор x.

Для определения вектора x используется информация, полученная от асессоров. Для этого для каждого запроса $q \in Q$ формируется вектор $\nu^{\exp}(q)$, отражающий мнение экспертов о релевантности страниц запросу. Считается, что также задана "мера несоответствия" $\mu(p, \nu^{\exp})$, согласно которой можно измерить насколько соответствующая часть вектора $p(q,x)$ близка к вектору $\nu^{\exp}(q)$. Вектор параметров x определяется, исходя из решения следующей задачи оптимизации:

$$F_Q(x) = \sum_{q \in Q} \mu\Big(p(q,x), \nu^{\exp}(q)\Big) \to \min_{x \in \mathrm{R}^d}. \tag{6}$$

Множество Q разбивается на три подмножества Q_{L} (обучающее/Learning), Q_{T} (тестовое/Test) и Q_{V} (контрольное/Validation). Процесс обучения параметров модели происходит на множестве Q_{L} (этот процесс происходит до тех пор, пока ошибка $F_Q(x)$ на Q_{V} не станет увеличиваться). Если ранжирование выбирается из нескольких моделей, то этот выбор происходит с помощью множества Q_{V}. Предположим, что есть две модели: 1 и 2. Для каждой модели в процессе обучения получены векторы x^1, x^2. Далее в зависимости от того, что меньше $F_{Q_{\mathrm{V}}}\left(x^1\right)$ или $F_{Q_{\mathrm{V}}}\left(x^2\right)$, отдается предпочтение одной из этих моделей [48]. Множество Q_{T} используется для определения итогового качества ранжирования. Для удобства обозначений в оставшейся части этого пункта будем опускать нижний индекс у $F_Q(x)$.

Далее в этом пункте речь пойдет о функции ранжирования $p = p(q, x)$, построенной на основе модели PageRank, а именно являющейся решением системы линейных уравнений:

$$p^\top = (1-\delta)\, p^\top P\big(\mathrm{y}(q), x\big) + \delta \pi^\top \big(\mathrm{y}(q), x\big), \tag{7}$$

где $\mathrm{y}(q) = \{y(q,w)\}_{w=1}^n$ — вектор признаков, соответствующих запросу q, а зависимости $P\big(\mathrm{y}(q), x\big)$ (матрица) и $\pi\big(\mathrm{y}(q), x\big)$ (вектор) считаются известными (каждый элемент этой матрицы или вектора может быть вычислен за $\mathrm{O}(d)$). Таким образом, ранжирование можно осуществить, решив (7) МПИ (см. [13, п. 3]):

$$p^\top(t+1) = (1-\delta)\, p^\top(t)\, P\big(\mathrm{y}(q), x\big) + \delta \pi^\top \big(\mathrm{y}(q), x\big). \tag{8}$$

В действительности, не совсем ясно, как выбирать зависимости $P\big(\mathrm{y}(q), x\big)$, $\pi\big(\mathrm{y}(q), x\big)$ и $\mu(p, \nu^{\exp})$. Обычно их стараются выбирать так, чтобы задача (6) была гладкой (добиться еще и выпуклости задачи (6), т. е. выпуклости $F(x)$, к сожалению, пока не удавалось) с липшицевым градиентом:

$$\left\|\nabla F(y) - \nabla F(x)\right\|_2 \le L \|y - x\|_2 \quad \forall\, x, y. \tag{9}$$

В этом случае для решения задачи (6) можно использовать обычный *метод градиентного спуска* (МГС) [22, 60], восходящий к пионерским работам Б.Т. Поляка начала 60-х годов [23]:

$$x^{k+1} = x^k - \frac{1}{L}\nabla F\big(x^k\big), \tag{10}$$

или его *адаптивный вариант*[9] [55]:

1. $L^k = \dfrac{L^{k-1}}{2}$.
2. $x^{k+1} = x^k - \dfrac{1}{L^k}\nabla F\left(x^k\right)$.

[9]Адаптивный вариант МГС является полезным на практике, поскольку значение константы L, как правило, неизвестно, а ее грубые оценки сверху оказываются завышенными, что замедляет сходимость. Кроме того, в МГС с фиксированным шагом никак не учитывается, что по мере приближения к экстремуму L уменьшается. Если это учитывать, то метод будет быстрее сходиться. В адаптивном МГС происходит настройка на параметр L, отвечающий текущему участку пребывания метода (итерационной последовательности), а не на худшую точку, как в обычном МГС. Эксперименты показывают, что за счет адаптивности метод ускоряется на порядок.

3. Если[10] имеет место

$$F\left(x^{k+1}\right) > F\left(x^k\right) + \left\langle \nabla F\left(x^k\right), x^{k+1} - x^k \right\rangle + \frac{L^k}{2}\left\|x^{k+1} - x^k\right\|_2^2, \tag{11}$$

то $L^k := 2L^k$ и переход на шаг 2, иначе переход на следующую итерацию: $k := k+1$.

Среднее в расчете на одну итерацию число вычислений значения функции $F(x)$ и градиента в таком методе не превосходит 4 [55].

Используя (9), можно оценить скорость глобальной сходимости МГС (10) и его адаптивного варианта, исходя из следующей оценки (см., например, [64]):

$$F\left(x^{k+1}\right) = F\left(x^k - \frac{1}{L}\nabla F\left(x^k\right)\right) \le F\left(x^k\right) - \frac{1}{2L}\left\|\nabla F\left(x^k\right)\right\|_2^2. \tag{12}$$

Выберем в качестве *критерия останова метода* условие $\left\|\nabla F\left(x^k\right)\right\|_2 \le \varepsilon$. Тогда, согласно (12), на каждой итерации метода (10) происходит уменьшение целевой функции не менее чем на $\varepsilon^2/(2L)$. Отсюда можно заключить, что число итераций, которые необходимо сделать до остановки, оценивается как $N \sim 2L/\varepsilon^2$. Несмотря на грубость проведенных рассуждений, оказывается, что оценка

$$N \sim L/\varepsilon^2 \tag{13}$$

в общем случае является неулучшаемой [20, 38] с точностью до мультипликативной константы, зависящей только от $F\left(x^0\right) - F\left(x_*\right)$, где x_* — стационарная точка (т. е. $\nabla F\left(x_*\right) = 0$), к которой сходится метод, а x^0 — точка старта метода.

Заметим, что МГС для любой точки старта сходится к одной из стационарных точек функции $F(x)$, вообще говоря, зависящей от точки старта, но не обязательно к локальному минимуму[11] [22]. Рассчитывать на возможность отыскания глобального минимума $F(x)$ без дополнительных предположений не приходится, поскольку для невыпуклых задач даже с единственным локальным минимумом (являющимся глобальным минимумом) существует *нижняя оценка* $N \sim \varepsilon^{-(d-1)}$ на необходимое число итераций [21]. Причем на каждой итерации можно вычислять производные $F(x)$ сколь угодно высокого порядка (если последние существуют) в одной выбранной на этой итерации точке. На практике с отмеченной проблемой часто помогает бороться мультистарт [15]: независимый запуск траекторий метода из разных точек. Однако такой способ поиска глобального минимума является в общем случае очень трудозатратным [15]. Недавние исследования, связанные с *обучением глубоких нейронных сетей* (Deep Learning) [45, 47, 63], показали, что совсем не обязательно всегда пытаться найти именно глобальный минимум. Например, если функция имеет много локальных минимумов приблизительно с одним значением целевой функции, то с точки зрения качества обучения не так важно в каком из минимумов окажется точка, сгенерированная методом. По-видимому, такая ситуация часто возникает в Deep Learning и ряде других задач машинного обучения.

Описанный выше МГС и его адаптивный вариант предполагают возможность вычисления точного градиента $\nabla F(x)$ функции $F(x)$. Однако итерационный процесс (8) позволяет вычислять только приближенно лишь значение $p(q, x)$, а следовательно, лишь значение функции $F(x)$. Если забыть на некоторое время про неточность вычисления $F(x)$,

[10] Из (9) следует, что неравенство $F\left(x^{k+1}\right) \le F\left(x^k\right) + \left\langle \nabla F\left(x^k\right), x^{k+1} - x^k \right\rangle + \frac{L^k}{2}\left\|x^{k+1} - x^k\right\|_2^2$ выполнено при $L^k \ge L$. Значит, цикл проверки неравенства (11) будет конечным.

[11] Впрочем, путем усложнения описанной процедуры, допуская возможность вычисления гессиана $\nabla^2 F\left(x^k\right)$, можно добиться, чтобы сходимость была именно к локальному минимуму (см., например, [64]).

то при весьма общих условиях (см., например, [14, 32, 60]) на основе "графа вычисления" $F(x)$ можно построить обратный граф вычисления $\nabla F(x)$ за время, не превышающее[12] 4·[время расчета $F(x)$], причем в большинстве случаев число 4 может быть уменьшено до числа из интервала $(2,3)$ [32]. Соответствующая общая техника называется[13] (быстрым) *автоматическим дифференцированием* (БАД) [14, 32, 60] (automatic differentiation), а в литературе по нейронным сетям — *методом обратного распространения ошибки* [32, 45] (back propagation). Интересно при этом заметить, что расчет матрицы Якоби $[\partial p(q,x)/\partial x]$ не может быть в общем случае осуществлен быстрее, чем за время $2\,d$·[время расчета $p(q,x)$].

Вернемся к неточности вычисления $p(q,x)$, а следовательно, и $F(x)$. Проблема в том, что в БАД работа должна происходить с точным алгоритмом вычисления значения функции, так как именно по этому алгоритму строится соответствующая "обратная" процедура расчета градиента. Если алгоритм, вычисляющий функцию $F(x)$, в свою очередь, является неточным, то не понятно, к чему приведет в этом случае техника БАД. Собственно, именно такая проблема и возникает в подходе к решению различных вариационных задач, задач оптимального управления и ряда обратных задач, предполагающем сначала дискретизацию задачи, например за счет замены системы дифференциальных уравнений разностной схемой, а затем использование техники БАД для вычисления градиента дискретизированного функционала [8, 14]. Насколько нам известно, на данный момент в общем случае не существует теоретических обоснований у описанного подхода. Впрочем, в важных частных случаях или в общем случае, но без точных оценок, обоснование имеется в книге [8] (см. также цитированную там литературу).

Однако можно пойти и по другому пути, основанному на концепции неточного оракула [11, 23, 34, 39, 40] неточности в вычислении $F(x)$ и $\nabla F(x)$ на скорость сходимости численного метода для задачи (6) (сейчас будет удобно уже рассматривать сразу адаптивный МГС). Будем рассматривать сразу адаптивный МГС. Предположим, что на каждой итерации мы имеем доступ к $\left(\tilde{\delta}, L\right)$-оракулу (в нашем случае численной процедуре, построенной на базе (8); не стоит путать $\delta = 0.15$ в (7), (8) с введенным здесь $\tilde{\delta}$), который для любого $x \in \mathrm{R}^d$ возвращает такие $F_{\tilde{\delta}}(x)$ и $\nabla F_{\tilde{\delta}}(x)$, что

$$\left|F_{\tilde{\delta}}(x) - F(x)\right| \le \tilde{\delta}, \qquad \left\|\nabla F_{\tilde{\delta}}(x) - \nabla F(x)\right\|_2 \le \sqrt{8L\tilde{\delta}},$$

и для любого $y \in \mathrm{R}^d$ имеет место неравенство

$$F(y) \le F_{\tilde{\delta}}(x) + \left\langle \nabla F_{\tilde{\delta}}(x), y - x\right\rangle + \frac{L}{2}\|y-x\|_2^2 + \tilde{\delta}.$$

Если адаптивный МГС с заменой условия (11) на условие

$$F_{\tilde{\delta}}\left(x^{k+1}\right) > F_{\tilde{\delta}}\left(x^k\right) + \left\langle \nabla F_{\tilde{\delta}}\left(x^k\right), x^{k+1} - x^k\right\rangle + \frac{L^k}{2}\left\|x^{k+1} - x^k\right\|_2^2 + 2\tilde{\delta}$$

и критерием останова $\left\|\nabla F_{\tilde{\delta}}\left(x^k\right)\right\|_2 \le \varepsilon/2$ работает с описанным выше $\left(\tilde{\delta}, L\right)$-оракулом, где $\tilde{\delta} \simeq \varepsilon^2/(32L)$, то после $N \sim 16L/\varepsilon^2$ итераций $\left\|\nabla F\left(x^N\right)\right\|_2 \le \varepsilon$. При доказательстве

[12] Чтобы проще было это понять, можно представить себе, например, функцию $F(x) = \langle c, x\rangle$.

[13] Техника автоматического дифференцирования предполагает гладкость всех функций. В нашем случае это требует некоторых оговорок для существования гладкой зависимости $p(q,x)$ (теорема о неявной функции).

этой оценки используется то, что всегда $L^k \le 2L$. В общем случае[14] $\left(\tilde{\delta}, L\right)$-оракул определяется, как правило, согласованной дискретизацией процедур расчета $F(x)$ и $\nabla F(x)$. Подчеркнем еще раз, что в предлагаемом подходе $\nabla F_{\tilde{\delta}}(x)$ получается непосредственно из $\nabla F(x)$, а не из $F_{\tilde{\delta}}(x)$.

С точки зрения практической реализации описанного выше подхода всегда встает вопрос о том, как по заданному ε строить $\left(\tilde{\delta}, L\right)$-оракул, минимизируя объем вычислений. В рассматриваемом в этом пункте примере такая проблема практически не стоит, поскольку время вычисления $\left(\tilde{\delta}, L\right)$-оракула пропорционально $\ln\left(\tilde{\delta}^{-1}\right)$ [34]. В частности, в данном случае можно просто пользоваться теоретическими оценками из [34]. К сожалению, в общем случае возникающие тут теоретические оценки (например оценки аппроксимации и устойчивости разностной схемы $\tilde{\delta}(h)$, использованной при дискретизации с шагом h системы дифференциальных уравнений в задаче оптимального управления), как правило, оказываются сильно завышенными, поэтому сложность вычисления $\left(\tilde{\delta}, L\right)$-оракула, пропорциональная $\tilde{\delta}^{-\rho}$, не позволяет надеяться на минимизацию объема вычислений при таком подходе. Разумный выход из данной ситуации заключается в *рестартах* по h [60]: запускаем метод при некотором h, затем при $h := h/2$ и т. д. до тех пор, пока при переходе на следующий шаг не будет наблюдаться заметных отличий в результатах. Последнее нуждается в пояснениях и уточнениях и зависит от специфики рассматриваемой задачи. Этому планируется посвятить отдельную работу.

Выше много внимания было уделено проблеме неточности вычислений, однако главная проблема БАД в рассматриваемом здесь приложении — это ресурсы памяти. Точнее говоря, проблема большой памяти — это общая проблема БАД [60]: требуется хранить весь граф вычислений значения функции в памяти, чтобы можно было построить обратный граф и использовать его для расчета градиента. В нашем случае затраты памяти оказываются слишком большими. Действительно, используя обозначения [13, п. 3], можно получить следующую, довольно грубую, оценку на требуемую память для одного запроса $q \in Q$:

$$\sim sn \cdot \left(d + \ln\left(\varepsilon^{-1}\right)/\delta\right) \ge 10^{13} \cdot 10^3 \simeq 10^{16},$$

т. е. получается порядка 10^4 терабайт. Получить доступ к таким ресурсам достаточно быстрой памяти (тем более, оперативной) на деле не представляется возможным. В качестве возможного решения отмеченной проблемы можно предложить использовать вместо МГС его покомпонентный вариант ПМГС (см., например, [11, 54] в выпуклом случае), который описан ниже в упрощенном варианте: *равновероятно и независимо от предыдущих розыгрышей выбрать* $i_k \in [1, \dots, d]$:

$$x_{i_k}^{k+1} = x_{i_k}^{k+1} - \frac{1}{L_{i_k}} \frac{\partial F\left(x^k\right)}{\partial x_{i_k}}, \quad x_i^{k+1} = x_i^k, \ i \ne i_k,$$

где предполагается, что для всех $i \in [1, \dots, d]$ $\left|\partial F(x + he_i)/\partial x_i - \partial F(x)/\partial x_i\right| \le L_i h$, e_i — i-й орт.

[14]Под “общим случаем” понимаются всевозможные приложения описанного подхода к решению различных вариационных задач, задач оптимального управления и ряда обратных задач [8, 14, 17]. Отметим, что конечномерность пространства, в котором происходит оптимизация ($d < \infty$), была не существенна (не использовалась при получении оценки на N): x может принадлежать бесконечномерному гильбертову пространству. Предлагаемый подход заключается в том, что решать исходную задачу стоит в этом самом пространстве (вообще говоря, бесконечномерном), не дискретизируя исходную постановку, однако при организации вычислительного процесса каждый раз использовать не идеальные значения функции и градиента, а их приближенные значения. Полученная выше зависимость $\tilde{\delta}(\varepsilon)$ позволяет оптимальным образом подбирать параметры дискретизации/аппроксимации, исходя из желаемой в итоге точности $h(\varepsilon)$.

Анализ скорости сходимости в среднем ПМГС, а также его адаптивного варианта и вариантов этих методов, работающих с неточным оракулом, аналогичен приведенным выше рассуждениям. В частности, можно получить (для упрощения изложения все огрублено) следующую оценку числа итераций:

$$N \sim \bar{L}d/\varepsilon^2, \tag{14}$$

где $\bar{L} \simeq \frac{1}{n}\sum_{i=1}^n L_i \leq L$.

Отметим, что условие $\left\|\nabla F\left(x^k\right)\right\|_2 \leq \varepsilon$ теперь уже нельзя проверять на каждой итерации, поскольку тогда теряется смысл в использовании на каждой итерации только одной компоненты градиента, поскольку все равно требуется вычисление полного градиента. Предлагается проверять $\left\|\nabla F\left(x^k\right)\right\|_2 \leq \varepsilon$ через каждые $\sim d$ итераций. Однако этот способ также упирается в отмеченную выше проблему нехватки памяти, поскольку все равно приходится считать полный градиент $\nabla F\left(x^k\right)$. Другой способ состоит в приближенном вычислении $\nabla F\left(x^k\right)$, заменяя в $\nabla F\left(x^k\right)$ недоступные на данной k-й итерации компоненты градиента последними на данный момент известными их значениями.

Несложно построить на базе (8) метод, вычисляющий с требуемой точностью частную производную $\partial F(x)/\partial x_i$ за число итераций $\sim \ln\left(\varepsilon^{-1}\right)/\delta$ с такой же по порядку стоимостью итерации как у метода (8) и с затратами памяти $\sim sn \simeq 10$ Тб [34]. Чтобы лучше это понять, можно использовать конечную разность $\left(F(x+he_i)-F(x)\right)/h$ при специальном выборе $h = h(\varepsilon)$ вместо $\partial F(x)/\partial x_i$ [11, 34, 56]. Все это, конечно, приводит к дополнительным неточностям и, как следствие, замедлению скорости сходимости ПМГС, однако все эти поправки имеют характер логарифмических множителей [34], что не может принципиально изменить качество метода. Таким образом, сопоставляя полученный выигрыш в ресурсах памяти в d раз с потенциально аналогичной потерей в числе итераций (следует сопоставить формулы (13), (14)) можно сказать, что множитель d перешел из затрат памяти в затраты времени. На самом деле, в (14) может быть $\bar{L} \ll L$. Следуя Ю.Е. Нестерову, рассмотрим следующий пример [11]:

$$F(x) = \frac{1}{2}\langle x, Sx\rangle - \langle b, x\rangle, \tag{15}$$

где S — симметричная матрица, все элементы которой являются числами от 1 до 2. Тогда $L = \lambda_{\max}(S) \geq \lambda_{\max}\left(1_d 1_d^\top\right) = d$, а $L_i = S_{ii} \leq 2$, т. е. $\bar{L} \leq 3$.

Отметим также, что описанный выше ПМГС и его вариации во многих конкретных интересных на практике случаях (см., например, [11]) работают намного эффективнее, чем в рассматриваемом в этом пункте общем случае за счет того, что стоимость пересчета $\partial F\left(x^{k+1}\right)/\partial x_i$ для ПМГС может быть осуществлена в $\sim d$ раз быстрее чем расчет $\nabla F(x)$. Это не сложно понять на примере функции (15): полноценный расчет градиента $\nabla F(x)$ стоит d^2, а пересчет градиента $\nabla F\left(x^{k+1}\right) = Sx^{k+1} - b$ с учетом того, что Sx^k уже известно с предыдущей итерации и $x_{i_k}^{k+1} = x_{i_k}^{k+1} - L_{i_k}^{-1}\partial F\left(x^k\right)/\partial x_{i_k}$, стоит $2\,d$.

Если в функции (6) очень много слагаемых и возможности распараллеливания вычислений ограничены, то использовать описанный выше МГС не разумно из-за огромных вычислительных затрат на каждой итерации. В этом случае у задачи (6) ярко проявляется специальная структура функции вида суммы с огромным числом слагаемых, которую можно использовать за счет введения специальных рандомизаций в процедуру МГС [31].

Для улучшения свойств решения задачи (6) можно вносить различные композиты в функционал [48, 55] (регуляризации, штрафы за сложность модели — попытка контроля переобучения [63]). Описанные выше подходы несложно распространить и на этот случай.

Если задача

$$F(x) \to \min_x,$$

рассматриваемая в гильбертовом пространстве, оказывается выпуклой (μ-сильно выпуклой), то можно говорить о поиске глобального минимума [11, 21, 22, 60]. Будем считать, что на каждой итерации доступен $\left(\tilde{\delta}, L, \mu\right)$-оракул [11, 39, 40], который выдает на запрос, в котором указывается только одна точка x, такую пару $\left(F_{\tilde{\delta}}(x), \nabla F_{\tilde{\delta}}(x)\right)$, что для любых y имеет место неравенство

$$\frac{\mu}{2}\|y-x\|^2 \le F(y) - F_{\tilde{\delta}}(x) - \left\langle \nabla F_{\tilde{\delta}}(x), y-x\right\rangle \le \frac{L}{2}\|y-x\|^2 + \tilde{\delta}.$$

Существует такая однопараметрическая (параметр $p \in [0,1]$: значение $p=0$ отвечает обычному методу градиентного спуска [39, 60], а значение $p=1$ отвечает быстрому градиентному методу Ю.Е. Нестерова [22, 55]) линейка методов (см. [11, 39, 40]), получающих на вход только параметры L и μ, которые работают по следующим (неулучшаемым [39]) оценкам:

$$\begin{aligned} &F\left(x^N\right) - F_* \le \varepsilon, \\ &N = \mathrm{O}\left(\min\left\{\left(\frac{LR}{\varepsilon}\right)^{\frac{1}{1+p}}, \left(\frac{L}{\mu}\right)^{\frac{1}{1+p}} \left\lceil \ln\left(\frac{\mu R^2}{\varepsilon}\right)\right\rceil\right\}\right), \\ &\tilde{\delta} \le \mathrm{O}\left(\frac{\varepsilon}{N^p}\right), \end{aligned} \tag{16}$$

где можно считать (см. [11]), что R — расстояние от точки старта до решения, а точнее, ближайшего к этой точке решения. Если дополнительно на одной итерации разрешается обращаться несколько раз за значением $F_{\tilde{\delta}}(x)$, то, подобно написанному ранее в этом пункте, можно предложить адаптивные варианты описываемой линейки методов, которым не требуется подавать на вход параметр L и для которых в оценку (16) входит не худшая (по всем шагам итерационного процесса) константа L, а некоторая средняя (пространственно средняя). Если оптимизация происходит в конечномерном пространстве $x \in \mathrm{R}^d$, то при весьма общих условиях [11] можно предложить еще покомпонентные варианты описанной адаптивной линейки методов, для которых в оценки числа итераций добавляется множитель $\sim d$, но при этом и стоимость итерации уменьшается также в $\sim d$ раз. Однако при этом константа L усредняется не только по пространству, но и по направлениям всех ортов, а не берется по худшему направлению, как было раньше. Выгода от такой замены также может быть порядка $\sim d$ раз. То есть в целом, если условия позволяют [11], то лучше, в конечном итоге, использовать именно покомпонентные методы. Начиная с работы Ю.Е. Нестерова [54], эти методы стали повсеместно использоваться для решения всевозможных выпуклых задач [11], приходящих из Big Data Science.

Можно распространить приведенные выше результаты и на случай негладких (не обязательно выпуклых) постановок задач [11, 44, 57]. Также можно исследовать вопрос о том, когда и каким образом стоит рандомизировать [11] описанные выше процедуры с целью сокращения общего времени работы метода. Интересно также заметить, что теория регуляризации выпуклых постановок задач довольно неплохо разработана к настоящему моменту (см., например, [8, 11, 39]), что позволяет получать сходящиеся по аргументу последовательности, что оказывается полезно на практике. Выпуклость задачи также обеспечивает возможность замены исходной задачи двойственной к ней. Решение последней задачи в ряде случаев бывает проще осуществить. А с учетом того,

что линейка методов (16) — это линейка прямо-двойственных методов [11], то по последовательности, полученной при решении двойственной задачи, можно без существенных дополнительных затрат восстановить с такой же точностью, с которой решалась двойственная задача, и решение исходной (прямой) задачи.

Возвращаясь к затронутой ранее проблеме оптимального выбора шага дискретизации в зависимости от желаемой точности решения исходной задачи $h(\varepsilon)$, возникающей при численном поиске градиента функционала, можно заметить, что для выпуклых постановок задач ситуация заметно интереснее, чем для невыпуклых. А именно, из формулы (16) можно получить зависимость $\tilde{\delta}_p(\varepsilon)$. Пускай имеется оценка $\tilde{\delta}(h)$ (обычно такие оценки как-то можно получать, однако, как уже отмечалось ранее, часто они оказываются завышенными). Наконец, пускай имеется оценка $T(h)$ того, сколько стоит итерация в зависимости от h. Исходя из зависимостей $\tilde{\delta}_p(\varepsilon)$ и $\tilde{\delta}(h)$, можно построить зависимость $h_p(\varepsilon)$. Тогда общее время работы метода будет равно $T\big(h_p(\varepsilon)\big)\,N_p(\varepsilon)$. Последнее выражение зависит от параметра $p \in [0,1]$. Исходя из минимизации этого выражения по $p \in [0,1]$, можно подобрать оптимальное значение этого параметра. Заметим, что для невыпуклых задач такой степени свободы не было.

В заключение заметим, что в довольно большом числе приложений, в действительности, имеют дело с функционалами вида $F(x) = \frac{1}{2}\|Ax-b\|_2^2$ (см., например, [13, п. 3]). В частности, очень популярны такого типа функционалы при решении обратных задач, в которых x является элементом пространства достаточно гладких функций с заданными краевыми условиями, а A является дифференциальным оператором [17]. Описанные выше довольно общие подходы, как ни странно, хорошо подходят и для решения таких специальных задач.

Благодарности. Авторы выражают благодарность рецензенту за замечания, позволившие улучшить качество статьи.

Литература

1. **Агаев Р.П., Чеботарев П.Ю.** Сходимость и устойчивость в задачах согласования характеристик (обзор базовых результатов) // Управление большими системами. — 2010. — Т. 30, № 1. — С. 470–505.
2. **Баймурзина Д.Р., Гасников А.В., Гасникова Е.В.** Теория макросистем с точки зрения стохастической химической кинетики // Тр. МФТИ. — 2015. — Т. 7, № 4. — С. 95–103.
3. **Баренблатт Г.И.** Автомодельные явления — анализ размерностей и скейлинг. — Долгопрудный: Изд. дом “Интеллект”, 2009.
4. **Батищева Я.Г., Веденяпин В.В.** II-й закон термодинамики для химической кинетики // Мат. моделирование. — 2005. — Т. 17, № 8. — С. 106–110.
5. **Богданов К.Ю.** Хищник и жертва: уравнение сосуществования // Квант. — 2014. — № 4–5. — С. 13–17.
6. **Бузун Н.О., Гасников А.В., Гончаров Ф.О., Горбачев О.Г., Гуз С.А., Крымова Е.А., Натан А.А., Черноусова Е.О.** Стохастический анализ в задачах. Часть 1 / А.В. Гасников. — М.: Изд-во МФТИ, 2016.
7. **Вайдлих В.** Социодинамика: системный подход к математическому моделированию в социальных науках. Пер. с англ. 3-е изд. — М.: УРСС, 2010.
8. **Васильев Ф.П.** Методы оптимизации. Т. 2. — М.: Изд-во МЦНМО, 2011.
9. **Гардинер К.В.** Стохастические методы в естественных науках. — М.: Мир, 1986.

10. Введение в математическое моделирование транспортных потоков. 2-е изд. / А.В. Гасников. — М.: Изд-во МЦНМО, 2013.

11. **Гасников А.В.** Эффективные численные методы поиска равновесий в больших транспортных сетях: Дис. ... докт. физ.-мат. наук: 05.13.18. — М.: Изд-во МФТИ, 2016.

12. **Гасников А.В., Гасникова Е.В.** Об энтропийно-подобных функционалах, возникающих в стохастической химической кинетике при концентрации инвариантной меры и в качестве функций Ляпунова динамики квазисредних // Мат. заметки. — 2013. — Т. 94, № 6. — С. 816–824.

13. **Гасников А.В., Гасникова Е.В., Двуреченский П.Е., Мохаммед А.А.М., Черноусова Е.О.** Вокруг степенного закона распределения компонент вектора PageRank. Часть 1. Численные методы поиска вектора PageRank // Сиб. журн. вычисл. математики / РАН. Сиб. отд-ние. — Новосибирск, 2017. — Т. 20, № 4. — С. 359–378.

14. **Евтушенко Ю.Г.** Оптимизация и быстрое автоматическое дифференцирование. — М.: Изд-во ВЦ РАН, 2013. — http://www.ccas.ru/personal/evtush/p/198.pdf.

15. **Жиглявский А.А., Жилинскас А.Г.** Методы поиска глобального экстремума. — М.: Наука, 1991.

16. **Занг В.-Б.** Синергетическая экономика: время и перемены в нелинейной экономической теории. — М.: Мир, 1999.

17. **Кабанихин С.И.** Обратные и некорректные задачи. — Новосибирск: Сибирское научное изд-во, 2009.

18. **Калинкин А.В.**Марковские ветвящиеся процессы с взаимодействием // Успехи мат. наук. — 2002. — Т. 57, № 2 (344). — С. 23–84.

19. **Литвак Н., Райгородский А.** Математика информационного века. — М.: МИФ, 2017.

20. **Малышев В.А., Пирогов С.А.** Обратимость и необратимость в стохастической химической кинетике // Успехи мат. наук. — 2008. — Т. 63, № 1 (379). — С. 4–36.

21. **Немировский А.С., Юдин Д.Б.** Сложность задач и эффективность методов оптимизации. — М.: Наука, 1979. — Перевод: Nemirovsky A.S., Yudin D.B. Problem Complexity and Method E_ciency in Optimization. — New York: J. Wiley & Sons, 1983.

22. **Нестеров Ю.Е.** Введение в выпуклую оптимизацию. — М.: Изд-во МЦНМО, 2010. — Перевод: Nesterov Yurii. Introductory Lectures on Convex Optimization: a basic course. — Massachusetts: Kluwer Academic Publishers, 2004.

23. **Поляк Б.Т.** Введение в оптимизацию. — М.: Наука, 1983. — Перевод: Polyak Boris. Introduction to Optimization. — New York: Optimization Software, 1987.

24. **Подлазов А.В.** Закон Ципфа и модели конкурентного роста // Новое в синергетике. Нелинейность в современном естествознании / Г.Г. Малинецкий. — М.: Изд-во "Либроком", 2009. — С. 229–256.

25. **Пурмаль А.П., Слободецкая Е.М., Травин С.О.** Как превращаются вещества. — М.: Наука, 1984. — (Серия "Библиотечка Квант"; вып. 36).

26. **Разжевайкин В.Н.** Анализ моделей динамики популяций. Учебное пособие. — М.: Изд-во МФТИ, 2010.

27. **Райгородский А.М.** Модели Интернета. — Долгопрудный: Изд. дом "Интеллект", 2013.

28. **Санов И.Н.** О вероятности больших отклонений случайных величин // Мат. сборник. — 1957. — Т. 42 (84), № 1. — С. 11–44.

29. Численная проверка степенного закона распределения компонент вектора PageRank. — https://github.com/KoIIdun/PageR. — https://github.com/KoIIdun/PageR/blob/master/data_exps/data_exp.txt. — https://github.com/KoIIdun/PageRank-gradient.

30. **Avrachenkov K., Lebedev D.** PageRank of scale-free growing networks // Internet Math. — 2006. — Vol. 3, № 2. — P. 207–232.

31. **Allen-Zhu Z., Hazan E.** Variance Reduction for Faster Non-Convex Optimization. — Ithaca, 2016. — (Preprint / Cornell University Library; arXiv:1603.05643).

32. **Baydin A.G., Pearlmutter B.A., Radul A.A., Siskand J.M.** Automatic Differentiation in Machine Learning: a Survey. — Ithaca, 2016. — (Preprint / Cornell University Library; arXiv:1502.05767).

33. **Blum A., Hopcroft J., and Kannan R.** Foundations of Data Science. — 2017. — http://www.cs.cornell.edu/jeh/book.pdf.

34. **Bogolubsky L., Dvurechensky P., Gasnikov A., Gusev G., Nesterov Yu., Raigorodskii A., Tikhonov A., Zhukovskii M.** Learning supervised PageRank with gradient-based and gradient-free optimization methods // Advances in Neural Information Processing Systems 29. / D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, and R. Garnett. — Curran Associates, Inc., 2016. — P. 4914–4922.

35. **Brin S., Page L.** The anatomy of a large-scale hypertextual web search engine // Comput. Network ISDN Syst. — 1998. — Vol. 30, № 1–7. — P. 107–117.

36. **Buffoni D., Gallinari P., Usunier N., and Calauzènes C.** Learning scoring functions with order-preserving losses and standardized supervision // Proc. of the 28th International Conference on Machine Learning (ICML–11). — 2011. — P. 825–832.

37. **Burges C.J.C.** From RankNet to LambdaRank to LambdaMART: An Overview. — 2010. — (Microsoft Research Technical Report; MSR-TR-2010-82).

38. **Calauzènes C., Usunier N., and Gallinari P.** On the (non-) existence of convex, calibrated surrogate losses for ranking // Advances in Neural Information Processing Systems. — 2012. — P. 197–205.

39. **Devolder O.** Exactness, Inexactness and Stochasticity in First-Order Methods for Large-Scale Convex Optimization: PhD thesis. — Louvain: CORE UCL, 2013.

40. **Dvurechensky P., Gasnikov A.** Stochastic intermediate gradient method for convex problems with inexact stochastic oracle // J. Optim. Theory Appl. — 2016. — Vol. 171, № 1. — P. 121–145.

41. **Ethier N.S., Kurtz T.G.** Markov Processes. — New York: John Wiley & Sons Inc., 1986.

42. **Franceschet M.** PageRank: Standing on the shoulders of giant // Communication of ACM. — 2011. — Vol. 54, № 6. — P. 92–101.

43. **Ghadimi S., Lan G.** Accelerated gradient methods for nonconvex nonlinear and stochastic programming // Math. Program. — 2016. — Vol. 156, № 1. — P. 59–99.

44. **Ghadimi S., Lan G., and Zhang H.** Generalized Uniformly Optimal Methods for Nonlinear Programming. — Ithaca, 2015. — (Preprint / Cornell University Library; arXiv:1508.07384).

45. **Goodfellow I., Bengio Y., and Courville A.** Deep Learning. — MIT Press, 2016. — http://www.deeplearningbook.org/.

46. **Grechnikov E.A.** The degree distribution and the number of edges between nodes of given degrees in the Buckley–Osthus model of a random web graph // Internet Math. — 2012. — Vol. 8, № 3. — P. 257–287.

47. **Hardt M., Ma T.** Identity Matters in Deep Learning. — Ithaca, 2015. — (Preprint / Cornell University Library; arXiv: 1611.04231).

48. **Hastie T., Tibshirani R., and Friedman J.** The Elements of Statistical Learning. — Springer, 2014.

49. **Jackson M.O.** Social and Economics Networks. — Princeton Univ. Press, 2008.

50. **Jaynes E.T.** Probability Theory. The Logic of Science. — Cambridge Univ. Press, 2003.

51. **Kapur J.N.** Maximum-Entropy Models in Science and Engineering. — New York: John Wiley & Sons Inc., 1989.

52. **Langville A.N., Meyer C.D.** Google's PageRank and Beyond: The Science of Search Engine Rankings. — Princeton Univ. Press, 2006.

53. **Mitzenmacher M.** A brief history of generative models for power law and lognormal distributions // Internet Math. — 2003. — Vol. 1, № 2. — P. 226–251.

54. **Nesterov Yu.** Efficiency of coordinate descent methods on large scale optimization problem // SIAM J. Optim. — 2012. — Vol. 22, № 2. — P. 341–362.

55. **Nesterov Yu.** Gradient methods for minimizing composite functions // Math. Progam. — 2013. — Vol. 140, № 1. — P. 125–161.

56. **Nesterov Yu.** Random Gradient-Free Minimization of Convex Functions // CORE Discussion Paper. — 2011. — Vol. 2011/1.

57. **Nesterov Yu.** Universal gradient methods for convex optimization problems // Math. Program. Ser. A. — 2015. — Vol. 152. — P. 381–404.

58. **Newman M.E.J.** Networks: An Introduction. — Oxford Univ. Press, 2010.

59. **Newman M.E.J.** Power laws, Pareto distributions and Zipf's law // Contemporary physics. — 2005. — Vol. 46, № 5. — P. 323–351.

60. **Nocedal J., Wright S.** Numerical Optimization. — Springer, 2006.

61. **Pandurangan G., Raghavan P., and Upfal E.** Using PageRank to characterize web structure // Internet Math. — 2006. — Vol. 3, № 1. — P. 1–20.

62. **Sandholm W.** Population Games and Evolutionary Dynamics. Economic Learning and Social Evolution. — Cambridge: MIT Press, 2010.

63. **Shalev-Shwartz S., Ben-David S.** Understanding Machine Learning: From Theory to Algorithms. — Cambridge Univ. Press, 2014.

64. **Wright S.J.** Optimization algorithms for data science // IAS/Park City Math. Ser. — 2016. — http://www.optimization-online.org/DB_FILE/2016/12/5748.pdf.

Поступила в редакцию 7 марта 2017 г.,
в окончательном варианте 16 июня 2017 г.

Литература в транслитерации

1. **Agaev R.P., Chebotarev P.Yu.** Skhodimost' i ustoychivost' v zadachah soglasovaniya harakteristik (obzor bazovyh rezul'tatov) // Upravlenie bol'shimi sistemami. — 2010. — T. 30, № 1. — S. 470–505.

2. **Baymurzina D.R., Gasnikov A.V., Gasnikova E.V.** Teoriya makrosistem s tochki zreniya stohasticheskoy himicheskoy kinetiki // Tr. MFTI. — 2015. — T. 7, № 4. — S. 95–103.

3. **Barenblatt G.I.** Avtomodel'nye yavleniya — analiz razmernostey i skeyling. — Dolgoprudnyy: Izd. dom "Intellekt", 2009.

4. **Batishcheva Ya.G., Vedenyapin V.V.** II-y zakon termodinamiki dlya himicheskoy kinetiki // Mat. modelirovanie. — 2005. — T. 17, № 8. — S. 106–110.

5. **Bogdanov K.Yu.** Hishchnik i zhertva: uravnenie sosushchestvovaniya // Kvant. — 2014. — № 4–5. — S. 13–17.

6. **Buzun N.O., Gasnikov A.V., Goncharov F.O., Gorbachev O.G., Guz S.A., Krymova E.A., Natan A.A., CHernousova E.O.** Stohasticheskiy analiz v zadachah. Chast' 1 / A.V. Gasnikov. — M.: Izd-vo MFTI, 2016.

7. **Vaydlih V.** Sotsiodinamika: sistemnyy podhod k matematicheskomu modelirovaniyu v sotsial'nyh naukah. Per. s angl. 3-e izd. — M.: URSS, 2010.
8. **Vasil'ev F.P.** Metody optimizatsii. T. 2. — M.: Izd-vo MTSNMO, 2011.
9. **Gardiner K.V.** Stohasticheskie metody v estestvennyh naukah. — M.: Mir, 1986.
10. Vvedenie v matematicheskoe modelirovanie transportnyh potokov. 2-e izd. / A.V. Gasnikov. — M.: Izd-vo MTSNMO, 2013.
11. **Gasnikov A.V.** Effektivnye chislennye metody poiska ravnovesiy v bol'shih transportnyh setyah: Dis. ... dokt. fiz.-mat. nauk: 05.13.18. — M.: Izd-vo MFTI, 2016.
12. **Gasnikov A.V., Gasnikova E.V.** Ob entropiyno-podobnyh funktsionalah, voznikayushchih v stohasticheskoy himicheskoy kinetike pri kontsentratsii invariantnoy mery i v kachestve funktsiy Lyapunova dinamiki kvazisrednih // Mat. zametki. — 2013. — T. 94, № 6. — S. 816–824.
13. **Gasnikov A.V., Gasnikova E.V., Dvurechenskiy P.E., Mohammed A.A.M., Chernousova E.O.** Vokrug stepennogo zakona raspredeleniya komponent vektora PageRank. CHast' 1. CHislennye metody poiska vektora PageRank // Sib. zhurn. vychisl. matematiki / RAN. Sib. otd-nie. — Novosibirsk, 2017. — T. 20, № 4. — S. 359–378.
14. **Evtushenko YU.G.** Optimizatsiya i bystroe avtomaticheskoe differentsirovanie. — M.: Izd-vo VTS RAN, 2013. — http://www.ccas.ru/personal/evtush/p/198.pdf.
15. **Zhiglyavskiy A.A., Zhilinskas A.G.** Metody poiska global'nogo ekstremuma. — M.: Nauka, 1991.
16. **Zang V.-B.** Sinergeticheskaya ekonomika: vremya i peremeny v nelineynoy ekonomicheskoy teorii. — M.: Mir, 1999.
17. **Kabanihin S.I.** Obratnye i nekorrektnye zadachi. — Novosibirsk: Sibirskoe nauchnoe izd-vo, 2009.
18. **Kalinkin A.V.**Markovskie vetvyashchiesya protsessy s vzaimodeystviem // Uspekhi mat. nauk. — 2002. — T. 57, № 2 (344). — S. 23–84.
19. **Litvak N., Raygorodskiy A.** Matematika informatsionnogo veka. — M.: MIF, 2017.
20. **Malyshev V.A., Pirogov S.A.** Obratimost' i neobratimost' v stohasticheskoy himicheskoy kinetike // Uspekhi mat. nauk. — 2008. — T. 63, № 1 (379). — S. 4–36.
21. **Nemirovskiy A.S., Yudin D.B.** Slozhnost' zadach i effektivnost' metodov optimizatsii. — M.: Nauka, 1979. — Perevod: Nemirovsky A.S., Yudin D.B. Problem Complexity and Method E_ciency in Optimization. — New York: J. Wiley & Sons, 1983.
22. **Nesterov Yu.E.** Vvedenie v vypukluyu optimizatsiyu. — M.: Izd-vo MTSNMO, 2010. — Perevod: Nesterov Yurii. Introductory Lectures on Convex Optimization: a basic course. — Massachusetts: Kluwer Academic Publishers, 2004.
23. **Polyak B.T.** Vvedenie v optimizatsiyu. — M.: Nauka, 1983. — Perevod: Polyak Boris. Introduction to Optimization. — New York: Optimization Software, 1987.
24. **Podlazov A.V.** Zakon Tsipfa i modeli konkurentnogo rosta // Novoe v sinergetike. Nelineynost' v sovremennom estestvoznanii / G.G. Malinetskiy. — M.: Izd-vo "Librokom", 2009. — S. 229–256.
25. **Purmal' A.P., Slobodetskaya E.M., Travin S.O.** Kak prevrashchayutsya veshchestva. — M.: Nauka, 1984. — (Seriya "Bibliotechka Kvant"; vyp. 36).
26. **Razzhevaykin V.N.** Analiz modeley dinamiki populyatsiy. Uchebnoe posobie. — M.: Izd-vo MFTI, 2010.
27. **Raygorodskiy A.M.** Modeli Interneta. — Dolgoprudnyy: Izd. dom "Intellekt", 2013.
28. **Sanov I.N.** O veroyatnosti bol'shih otkloneniy sluchaynyh velichin // Mat. sbornik. — 1957. — T. 42 (84), № 1. — S. 11–44.

29. Chislennaya proverka stepennogo zakona raspredeleniya komponent vektora PageRank. — https://github.com/KoIIdun/PageR. — https://github.com/KoIIdun/PageR/blob/master/data_exps/data_exp.txt. — https://github.com/KoIIdun/PageRank-gradient.

30. **Avrachenkov K., Lebedev D.** PageRank of scale-free growing networks // Internet Math. — 2006. — Vol. 3, № 2. — P. 207–232.

31. **Allen-Zhu Z., Hazan E.** Variance Reduction for Faster Non-Convex Optimization. — Ithaca, 2016. — (Preprint / Cornell University Library; arXiv:1603.05643).

32. **Baydin A.G., Pearlmutter B.A., Radul A.A., Siskand J.M.** Automatic Differentiation in Machine Learning: a Survey. — Ithaca, 2016. — (Preprint / Cornell University Library; arXiv:1502.05767).

33. **Blum A., Hopcroft J., and Kannan R.** Foundations of Data Science. — 2017. — http://www.cs.cornell.edu/jeh/book.pdf.

34. **Bogolubsky L., Dvurechensky P., Gasnikov A., Gusev G., Nesterov Yu., Raigorodskii A., Tikhonov A., Zhukovskii M.** Learning supervised PageRank with gradient-based and gradient-free optimization methods // Advances in Neural Information Processing Systems 29. / D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, and R. Garnett. — Curran Associates, Inc., 2016. — P. 4914–4922.

35. **Brin S., Page L.** The anatomy of a large-scale hypertextual web search engine // Comput. Network ISDN Syst. — 1998. — Vol. 30, № 1–7. — P. 107–117.

36. **Buffoni D., Gallinari P., Usunier N., and Calauzènes C.** Learning scoring functions with order-preserving losses and standardized supervision // Proc. of the 28th International Conference on Machine Learning (ICML–11). — 2011. — P. 825–832.

37. **Burges C.J.C.** From RankNet to LambdaRank to LambdaMART: An Overview. — 2010. — (Microsoft Research Technical Report; MSR-TR-2010-82).

38. **Calauzènes C., Usunier N., and Gallinari P.** On the (non-) existence of convex, calibrated surrogate losses for ranking // Advances in Neural Information Processing Systems. — 2012. — P. 197–205.

39. **Devolder O.** Exactness, Inexactness and Stochasticity in First-Order Methods for Large-Scale Convex Optimization: PhD thesis. — Louvain: CORE UCL, 2013.

40. **Dvurechensky P., Gasnikov A.** Stochastic intermediate gradient method for convex problems with inexact stochastic oracle // J. Optim. Theory Appl. — 2016. — Vol. 171, № 1. — P. 121–145.

41. **Ethier N.S., Kurtz T.G.** Markov Processes. — New York: John Wiley & Sons Inc., 1986.

42. **Franceschet M.** PageRank: Standing on the shoulders of giant // Communication of ACM. — 2011. — Vol. 54, № 6. — P. 92–101.

43. **Ghadimi S., Lan G.** Accelerated gradient methods for nonconvex nonlinear and stochastic programming // Math. Program. — 2016. — Vol. 156, № 1. — P. 59–99.

44. **Ghadimi S., Lan G., and Zhang H.** Generalized Uniformly Optimal Methods for Nonlinear Programming. — Ithaca, 2015. — (Preprint / Cornell University Library; arXiv:1508.07384).

45. **Goodfellow I., Bengio Y., and Courville A.** Deep Learning. — MIT Press, 2016. — http://www.deeplearningbook.org/.

46. **Grechnikov E.A.** The degree distribution and the number of edges between nodes of given degrees in the Buckley–Osthus model of a random web graph // Internet Math. — 2012. — Vol. 8, № 3. — P. 257–287.

47. **Hardt M., Ma T.** Identity Matters in Deep Learning. — Ithaca, 2015. — (Preprint / Cornell University Library; arXiv: 1611.04231).

48. **Hastie T., Tibshirani R., and Friedman J.** The Elements of Statistical Learning. — Springer, 2014.

49. **Jackson M.O.** Social and Economics Networks. — Princeton Univ. Press, 2008.

50. **Jaynes E.T.** Probability Theory. The Logic of Science. — Cambridge Univ. Press, 2003.

51. **Kapur J.N.** Maximum-Entropy Models in Science and Engineering. — New York: John Wiley & Sons Inc., 1989.

52. **Langville A.N., Meyer C.D.** Google's PageRank and Beyond: The Science of Search Engine Rankings. — Princeton Univ. Press, 2006.

53. **Mitzenmacher M.** A brief history of generative models for power law and lognormal distributions // Internet Math. — 2003. — Vol. 1, № 2. — P. 226–251.

54. **Nesterov Yu.** Efficiency of coordinate descent methods on large scale optimization problem // SIAM J. Optim. — 2012. — Vol. 22, № 2. — P. 341–362.

55. **Nesterov Yu.** Gradient methods for minimizing composite functions // Math. Progam. — 2013. — Vol. 140, № 1. — P. 125–161.

56. **Nesterov Yu.** Random Gradient-Free Minimization of Convex Functions // CORE Discussion Paper. — 2011. — Vol. 2011/1.

57. **Nesterov Yu.** Universal gradient methods for convex optimization problems // Math. Program. Ser. A. — 2015. — Vol. 152. — P. 381–404.

58. **Newman M.E.J.** Networks: An Introduction. — Oxford Univ. Press, 2010.

59. **Newman M.E.J.** Power laws, Pareto distributions and Zipf's law // Contemporary physics. — 2005. — Vol. 46, № 5. — P. 323–351.

60. **Nocedal J., Wright S.** Numerical Optimization. — Springer, 2006.

61. **Pandurangan G., Raghavan P., and Upfal E.** Using PageRank to characterize web structure // Internet Math. — 2006. — Vol. 3, № 1. — P. 1–20.

62. **Sandholm W.** Population Games and Evolutionary Dynamics. Economic Learning and Social Evolution. — Cambridge: MIT Press, 2010.

63. **Shalev-Shwartz S., Ben-David S.** Understanding Machine Learning: From Theory to Algorithms. — Cambridge Univ. Press, 2014.

64. **Wright S.J.** Optimization algorithms for data science // IAS/Park City Math. Ser. — 2016. — http://www.optimization-online.org/DB_FILE/2016/12/5748.pdf.