Вероятностные методы линейной алгебры и большие данные

В. В. Агишев, Г. И. Горемыкина

Аннотация— **Статья посвящена изучению применения вероятностных методов линейной алгебры для повышения скорости вычислений в условиях работы с большими объемами данных. В связи с тем, что при увеличении объема данных традиционные методы их обработки часто сталкиваются с проблемами масштабируемости и чрезмерной вычислительной сложностью, использование рандомизированных подходов приобретает всё большее значение. В статье исследуются ключевые алгоритмы вероятностных методов. Особо акцентируется внимание на скетчинге – использовании специальных алгоритмов для ускорения обработки больших объёмов данных, позволяющих существенно снижать вычислительные затраты, так как процесс обработки не требует анализа всех данных целиком, что особенно важно для работы с большими массивами информации.**

В статье приводятся примеры практического применения вероятностных методов на тестовых наборах данных, где продемонстрировано их преимущество в виде значительного повышения производительности вычислений по сравнению с традиционными методами. В частности, рассмотрена задача решения системы линейных уравнений методом наименьших квадратов, которая представляет собой классическую задачу оптимизации, часто встречающуюся в приложениях к большим данным. Вероятностные методы в таких задачах позволяют существенно сократить время обработки за счёт снижения размерности системы уравнений при сохранении приемлемого уровня точности.

Результаты проведённых экспериментов демонстрируют, что применение вероятностных методов действительно позволяет добиться существенного ускорения вычислительных процессов с минимальными потерями в точности. Это делает рандомизированные методы перспективным инструментом для обработки больших данных, особенно в областях, где необходимо быстрое и эффективное выполнение сложных вычислений, таких, как машинное обучение, анализ данных и оптимизация.

Ключевые слова— **Randomized Numerical Linear Algebra, большие данные, ускорение вычислений, матричные разложения, рандомизированные алгоритмы, численные методы, оптимизация.**

Статья получена 26 июня 2025.

Владимир Владиславович Агишев – студент Высшей школы кибертехнологий, математики и статистики, Российский экономический университет имени Г. В. Плеханова, ORCID: http://orcid.org/0000-0003-2813-1179 (e-mail: zanzibara61@yandex.ru)

Галина Ивановна Горемыкина – доцент кафедры математических методов в экономике Высшей школы кибертехнологий, математики и статистики, к.ф.-м.н., Российский экономический университет имени Г. В. Плеханова, ORCID: https://orcid.org/0000-0002-8047-5393 (e-mail: Goremykina.GI@rea.ru)

I. Введение

В последние десятилетия развитие технологий хранения данных привело к резкому увеличению объёма информации, которая может быть сохранена и обработана в электронном виде. Ежедневно многие крупные компании работают с петабайтами данных, которые поступают из различных источников и, как правило, плохо структурированы. Однако, несмотря на масштабы таких массивов данных, для аналитики и вычислений используется только их относительно небольшая структурированная часть. Объём даже этого подмножества может создавать значительные вычислительные нагрузки, что особенно актуально в условиях постоянного увеличения объёма данных. В таких ситуациях традиционные методы обработки информации становятся недостаточно эффективными. Вероятностные методы линейной алгебры, активно развивающиеся в последние годы, предлагают решения части этих задач. Они позволяют существенно снизить вычислительные затраты за счёт того, что не всегда требуют обработки всех данных целиком. Применяя случайные проекции и сокращение размерности, эти методы ускоряют выполнение вычислений, сохраняя при этом важные характеристики исходных данных.

Исследования по применению вероятностных алгоритмов для ускорения решения задач линейной алгебры начались в начале 2000-х годов и продолжают активно развиваться. Одной из первых ключевых работ в этой области стала статья 2007 года "Faster Least Squares Approximation" [1], где был представлен вероятностный алгоритм для приближенного вычисления методом наименьших квадратов, а также проанализированы его математические свойства. Эта работа задала вектор дальнейших исследований в данной области. В 2011 году были опубликованы сразу две важные работы. Первая – "Blendenpik: Supercharging LAPACK's Least-Squares Solver" [2], в которой для ускорения вычислений в задаче обработки данных методом наименьших квадратов был предложен алгоритм, основанный на проекциях. Эта статья связала теорию с практикой, так как алгоритм в ней был интегрирован в широко используемый программный пакет LAPACK. Вторая работа – "Randomized Algorithms for Matrices and Data" [3] – систематизировала различные вероятностные алгоритмы для решения задач линейной алгебры, таких как приближенные матричные разложения и обработка данных методом наименьших квадратов.

С течением времени применение вероятностных алгоритмов стало важным инструментом для решения различных задач обработки больших объёмов данных. Существует большое количество работ для различных видов матричного разложения для аппроксимации исходной задачи [4, 5, 6, 7], большинство из которых основаны на статье [8]. С 2019 года язык R включает в себя приближенный вероятностный метод по расчёту сингулярного разложения *(Singular Value Decomposition, SVD)* матрицы [9]. Новый подход пытаются применить и в нейронных сетях, например, для сокращения больших матричные умножений с сохранением при этом определённой точности [10], а также в градиентных методах оптимизации [11, 12, 13, 14, 15].

Данная работа посвящена изучению методов вероятностной линейной алгебры, а также их основным идеям и концепциям. Кроме теоретической части, был проведен эксперимент, сравнивающий скорость вычисления решения системы уравнений с использованием метода наименьших квадратов, вероятностных методов и классического подхода. Цель работы — продемонстрировать потенциал, который приносят новые методы и концепции, и способствовать их внедрению в существующие подходы.

II. Теоретический анализ

Перед тем, как перейти к детальному рассмотрению вероятностных алгоритмов линейной алгебры *(RandNLA)*, важно ввести несколько ключевых понятий, которые лежат в их основе. Эти понятия будут использоваться в соответствии с нотациями из обзорной статьи [16].

Одним из центральных понятий RandNLA является «скетчинг» *(sketching)* — метод уменьшения размерности данных, который позволяет производить вычисления на значительно меньших матрицах и при этом сохранить ключевые свойства исходных данных. Эффективное уменьшение размерности достигается с помощью специальных операторов, которые применяются к данным и создают их компактные представления — «скетчи» *(sketch)*. Такие операторы являются различными матрицами и обозначаются как S. Они бывают двух типов: левосторонние и правосторонние. В данной работе рассматривается только левосторонний скетчинг.

Рассмотрим применения левостороннего скетчинга на примере решения системы линейных уравнений методом наименьших квадратов. Пусть имеется система уравнений $Ax = b$, $A \in R^{m \times n}$, $x \in R^n$, $b \in R^m$, m значительно больше, чем n. Оператор скетчинга имеет размерность $k \times m$, k выбирается значительно меньшим чем m, например $k = 2n$. Применяя оператор скетчинга S к A (рис 1.) и b, получим систему $SAx = Sb$ или в сокращённом виде $\tilde{A}x = \tilde{b}$.

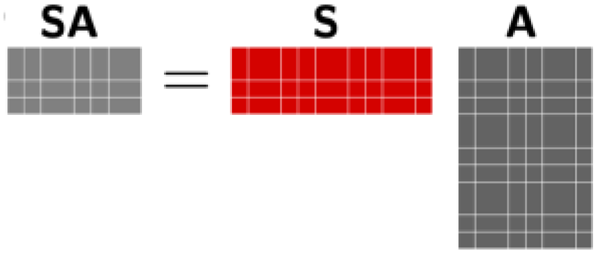

Рис. 1. Результат умножения матрицы S на A [16].

После умножения получается система со значительно меньшей размерностью, вследствие чего поиск решений в ней можно значительно ускорить.

Из приведённого примера становится ясно, что успех скетчинга зависит от того, насколько правильно задан оператор S. Существует различное множество способов задания матрицы S, но в основном она получается путём сэмплирования из распределения всех скетчинговых операторов $\mathcal{D}$, $S \sim \mathcal{D}$. $\mathcal{D}$ задаётся таким способом, чтобы удовлетворять свойствам 1 и 2:

$$M[S] = 0 \tag{1}$$

$$M[S^T S] = 1, \tag{2}$$

где $M[\cdot]$ — математическое ожидание случайной величины.

Свойство 1 является практически повсеместным для RandNLA, а в свойстве 2 есть некоторая доля свободы (подробнее изложено в [16]).

Пусть заданы произвольный оператор скетчинга S размерности $d \times m$, евклидово подпространство $L \subseteq \mathbb{R}^m$, $x \in L$ и $\delta \in [0, 1]$. Тогда, согласно [17], оператор S называется δ-встраиванием, а δ – искажением, если выполняется следующее неравенство

$$(1 - \delta)\|x\|_2 \leq \|Sx\|_2 \leq (1 + \delta)\|x\|_2, \tag{3}$$

где $\|\cdot\|_2$ — евклидова норма.

Если подпространство L является образом матрицы A при линейном отображении, то S является δ-встраиванием для L только тогда, когда верно неравенство:

$$(1 - \delta)^2 A^T A \preccurlyeq (SA)^T(SA) \preccurlyeq (1 + \delta)^2 A^T A \tag{4}$$

Условия на искажение, такие, как неравенства 3 и 4, гарантируют, что длины векторов не изменятся слишком сильно при переходе в пространство меньшей размерности, что позволяет сохранить точность результатов последующих вычислений. Все основные операторы скетчинга подбираются таким образом, чтобы соответствовать двум указанным неравенствам 3 и 4.

Количество различных операторов скетчинга велико. Оператор может быть сэмплирован из распределения Радемахера [18], где каждый элемент матрицы с вероятностью 0,5 принимает значение +1 и с вероятностью 0,5 – значение -1. Можно сэмплировать также из равномерного распределения на отрезке $[-1, +1]$. Третьим популярным методом задания $\mathcal{D}$ является нормальное гауссовское распределение.

Далее будет приведена практическая оценка скетчинга и её влияние на итоговый результат. Но даже без неё ясно, что результат будет не столь точным, как при оптимизации на всём объёме данных. В связи с этим выделяют два подхода. В первом, *"sketch-and-solve"* [3, 19], оставляют результат оптимизации после скетчинга таким, какой он есть. Этот подход подойдёт для задач, где не требуется высокая точность, например, для задач машинного обучения. Во втором, *"sketch-and-precondition"* [20], используется приближённое решение в качестве начальной точки оптимизации. Многие современные пошаговые алгоритмы из прикладных пакетов линейной алгебры поддерживают данный функционал. Точность получается практически такой же, как и в случае применения пошагового алгоритма на всем наборе данных, а время сходимости сильно уменьшается за счёт улучшенной начальной точки.

Перейдем к задаче решения систем линейных уравнений вида $Ax = b$ методом наименьших квадратов. Их решение заключается в нахождении такого вектора x, чтобы минимизировать следующую евклидову норму:

$$\|Ax - b\|_2^2 \, . \tag{5}$$

Применив скетчинг к данной задаче, получаем решение системы в виде:

$$x = (SA)^+(Sb) \tag{6}$$

и при этом выполняется:

$$(SA)^+(Sb) \in \ argmin_x \|S(Ax - b)\|_2^2 \, , \tag{7}$$

где $(SA)^+$ обозначает псевдообратную матрицу к матрице SA [21], $argmin_x \|S(Ax - b)\|_2^2$ — решение оптимизационной задачи $\|S(Ax - b)\|_2^2 \to min$.

В этом случае скетчинг используется для уменьшения размерности системы, что позволяет ускорить вычисления, сохраняя при этом ключевые свойства задачи. На основании неравенства (4), можно оценить границу качества решения и получить неравенство:

$$\|A(SA)^+(Sb) - b\|_2 \leq \left(\frac{1+\delta}{1-\delta}\right) \|AA^+b - b\|_2 \tag{8}$$

Неравенство 8 показывает, что точность решения после применения скетчинга зависит от искажения δ, связанного с оператором скетчинга. Чем меньше δ, тем точнее будет приближённое решение относительно исходной системы.

В качестве подхода к решению системы после уменьшения размерности через скетчинг может применяться любой алгоритм, так как размерность задачи была значительно сокращена. Подробнее о том, как оценивать ошибку решения на этом этапе изложено в статье [22].

После применения скетчинга решение можно уточнить через любой алгоритм последовательной оптимизации:

$$x \ = \ iterative\_ls\_solver\,(A, b, \epsilon, T, x_0) \tag{9}$$

поддерживающий начальное приближенное решение x_0 и минимизирующий функцию $\|Ax - b\|_2^2$ с заданным максимальным количеством шагов T и точностью ε. При этом алгоритм будет применяться уже к исходным данным с большой размерностью.

III. Методология исследования

Для изучения эффективности применения скетчинга в задаче решения системы линейных уравнений был проведён ряд экспериментов на языке программирования Python. В качестве оператора скетчинга был выбран *"sparse Johnson–Lindenstrauss transforms" (SJLT)* [23], который входит в раздел операторов под названием *"Short-axis-sparse sketching operators"*. Все данное семейство основано на идеи создания разреженной матрицы скетчинга S, где для каждого столбца зафиксировано число p ненулевых значений. Важным условием является линейная независимость столбцов полученной матрицы скетчинга. Поэтому был использован метод равномерного выбора p индексов без замены для каждого столбца. Ненулевые индексы заполняются на основе распределения Радемахера со случайным знаком $\pm\frac{1}{\sqrt{p}}$.

Пример SJLT представлен на рис 2.

0.447214	0.447214	0	0	0	0	-0.44721	0	-0.44721	0	0	0.447214
0	0	-0.44721	0	-0.44721	0	-0.44721	0	0	-0.44721	0	-0.44721
0	0	0	-0.44721	0.447214	0	-0.44721	-0.44721	-0.44721	0.447214	-0.44721	0
0.447214	-0.44721	0.447214	-0.44721	0	-0.44721	0	-0.44721	0.447214	0	0.447214	0
-0.44721	0	0.447214	-0.44721	0	0	0.447214	0	-0.44721	0	0	0
0.447214	0	0.447214	0	-0.44721	0.447214	0	-0.44721	-0.44721	0.447214	-0.44721	-0.44721
0	0.447214	0	0	-0.44721	-0.44721	0	0	0	0.447214	-0.44721	0
-0.44721	-0.44721	0	0.447214	0	0.447214	-0.44721	0.447214	0	0.447214	0	0.447214
0	0	-0.44721	-0.44721	0.447214	0	0	0.447214	0	0	0.447214	0
0	0.447214	0	0	0	0.447214	0	0	0	0	0	-0.44721

Рис. 2. Пример оператора скетчинга SJLT, с $k = 5, S$ имеет размерность 10×12.

После применения к исходной системе $Ax = b$ оператора скетчига с размерностью $k \times m$ система решалась путем QR-разложения [24]. Из матрицы $\tilde{A}(SA)$ после разложения получалось две матрицы: $\tilde{Q}$ — матрица размера $k \times w$ с ортонормированными столбцами и $\tilde{R}$ — верхнетреугольная матрица размера $w \times m$, где $w = \min(k, m)$. Система после разложения принимает вид $\tilde{Q}\tilde{R}x = \tilde{b}$, или $\tilde{R}x = \tilde{Q}^{-1}\tilde{b}$. Далее обозначим $\tilde{Q}^{-1}\tilde{b}$ как z_0. По свойству QR разложения, $\tilde{R}^{-1}z_0$ будет являться решением системы, эквивалентным решению по формуле 6, и далее оно может быть использовано для sketch-and-precondition.

Дополнительно стоит упомянуть о еще одном возможном подходе. Вместо того, чтобы использовать x_0 в качестве начального решения, можно использовать z_0. В таком случае для оптимизации на полном объёме данных используется не матрица A, а $A_{precond} = A\tilde{R}^{-1}$, что помогает улучшить численную стабильность задачи и ускорить решение итеративными методами. Подробное о технике, называемой *"preconditioning"*, описано в статье [16].

В качестве алгоритма последовательной оптимизации был выбран LSQR [25]. Система уравнений составлялась с использованием сгенерированной матрицы A и вектора b, где матрица A имела заранее известный спектр сингулярных значений, что обеспечивало разрешимость системы и известное оптимальное решение. Ошибка полученного решения вычислялась по формуле:

$$error = \frac{\|x - x_{opt}\|_2}{1 + \min(\|x\|_2, \|x_{opt}\|_2)}, \quad (9)$$

где x – найденное решение, x_{opt} – истинное решение системы, а $\|\cdot\|_2$ евклидова норма.

IV. Полученные результаты

Сначала был проведён сравнительный анализ по прямому (не являющимся точным) решению через скетчинг. На рис. 3а изображено сравнение ошибок решений по формуле (9) для параметров $m = 100000$ и $n = 500$ по 30 повторным запускам. Оператор скетчинга имел размерность $2n \times m$, с $p = 8$. На рис. 3б представлен график, отображающий рост затрачиваемого времени с увеличением m. Временная шкала опускается ввиду сильной зависимости от спецификации вычислительной машины, на которой запускается программный код. Оператор скетчинга имел размерность 1000×500.

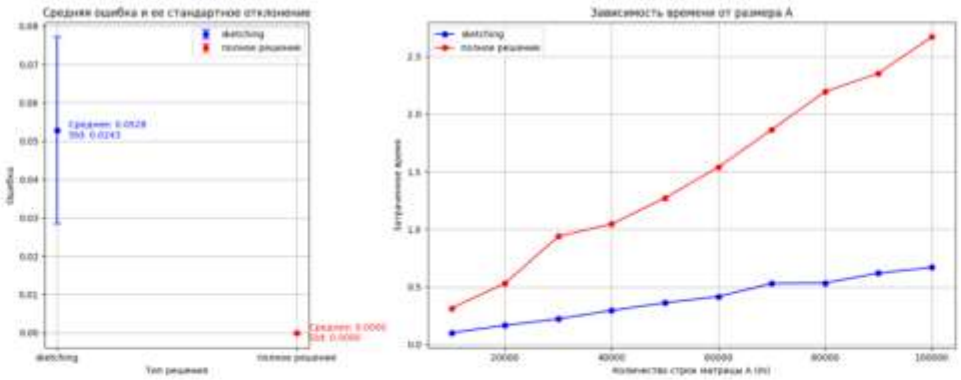

Рис. 3. Сравнение точного (а) и приближенного через скетчинг (б) решений.

Из рис. 3 видно, что у точного решения через скетчинг получается заметная ошибка, которая при этом обладает и большим стандартным отклонением. Оно возникает из-за сэмплирования матрицы S из распределения скетчинговых операторов $\mathcal{D}$. Скорость решение на рис. 3б с увеличением m растет значительно меньше, чем у точного решения системы.

Далее был произведён анализ решения через скетчинг и его доуточнения. Запуск производился в такой же конфигурации и результаты представлены на рис 4.

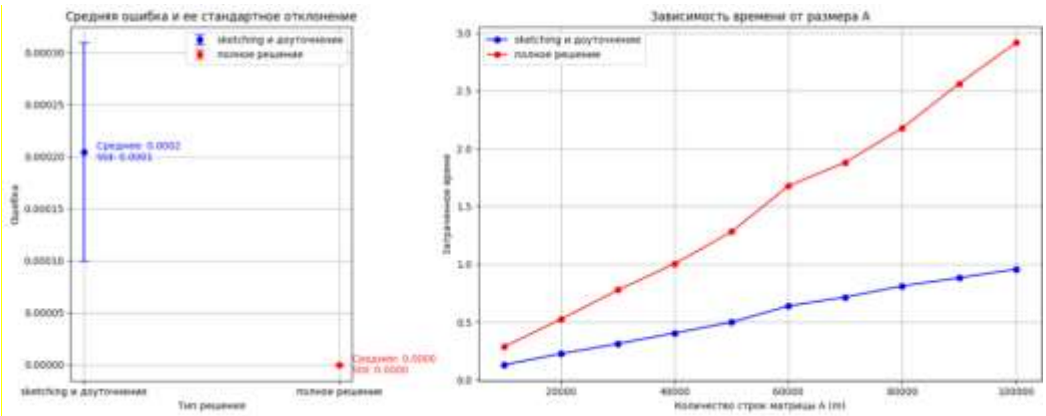

Рис. 4. Сравнение точного решения (а) и приближения через скетчинг с доуточнением (б).

Скорость решения при m, равном 100000, благодаря применению скетчинга, увеличивается в 3 раза, при этом точность практически достигает решения полной системы. На графиках важна также и тенденция быстрого роста затраченного времени у полного решения. Разрыв между ним и скетчингом при увеличении m будет увеличиваться все больше.

V. Заключение

В ходе работы был проведён анализ применения вероятностных методов линейной алгебры, в частности, скетчинга для решения системы линейных уравнений методом наименьших квадратов. Эксперименты подтвердили, что использование скетчинга позволяет значительно уменьшить размерность задачи и ускорить вычисления, сохраняя приемлемую точность. Особенно эффективным оказался подход с использованием скетчинга в сочетании с методами уточнения решения, который позволяет существенно повысить производительность без значительных потерь в точности.

Результаты тестирования показали, что прямое решение через скетчинг может приводить к некоторой ошибке, что обусловлено случайностью сэмплирования, однако уточнение решения позволяет свести её к минимуму. Кроме того, разница во времени вычислений между точным решением задачи и решением, полученным в результате применения скетчинга, становится всё более значительной по мере увеличения размера данных, что делает предложенный метод особенно перспективным для работы с большими системами уравнений.

Таким образом, вероятностные методы, такие, как, например, скетчинг, демонстрируют свою эффективность в задачах обработки больших объёмов данных и могут применяться для ускорения вычислений в таких областях, как машинное обучение и оптимизация, предлагая значительное сокращение времени решения при минимальных потерях точности.

БИБЛИОГРАФИЯ

[1] Drineas P., Mahoney M. W., Muthukrishnan S., Sarlos T. Faster Least Squares Approximation https://arxiv.org/abs/0710.1435

[2] Avron H., Maymounkov P., Toledo S. Blendenpik: Supercharging LAPACK's Least-Squares Solver https://doi.org/10.1137/090767911

[3] Mahoney M. W. Randomized algorithms for matrices and data https://arxiv.org/abs/1104.5557

[4] N. Benjamin Erichson, Peng Zheng, Krithika Manohar, Steven L. Brunton, J. Nathan Kutz, Aleksandr Y. Aravkin. Sparse Principal Component Analysis via Variable Projection. https://arxiv.org/abs/1804.00341

[5] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. “Fixed-rank approximation of a positive-semidefinite matrix from streaming data”. In: Advances in Neural Information Processing Systems. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. https://arxiv.org/abs/1706.05736

[6] S. Voronin and P.-G. Martinsson. “Efficient algorithms for CUR and interpolative matrix decompositions”. In: Advances in Computational Mathematics 43.3 (Nov. 2016), pp. 495–516. https://arxiv.org/abs/1412.8447

[7] Y. Dong and P.-G. Martinsson. Simpler is better: A comparative study of randomized algorithms for computing the CUR decomposition. 2021. https://arxiv.org/abs/2104.05877

[8] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. https://arxiv.org/abs/0909.4061

[9] Erichson N. B., Voronin S., Brunton S. L., Kutz J. N. Randomized Matrix Decompositions Using R. https://arxiv.org/abs/1608.02148

[10] Ootomo H., Yokota R. Mixed-Precision Random Projection for RandNLA on Tensor Cores https://arxiv.org/abs/2304.04612

[11] Filip Hanzely, Konstantin Mishchenko, and Peter Richtárik. SEGA: Variance reduction via gradient sketching. Advances in Neural Information Processing Systems, 31, 2018. https://arxiv.org/abs/1809.03054

[12] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. Fetchsgd: Communication-efficient federated learning with sketching. In International Conference on Machine Learning, pages 8253–8265. PMLR, 2020. https://arxiv.org/abs/2007.07682

[13] Deanna Needell, Rachel Ward, and Nati Srebro. Gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. Advances in neural information processing systems. https://arxiv.org/abs/1310.5715

[14] Alon Gonen, Francesco Orabona, and Shai Shalev-Shwartz. Solving ridge regression using sketched preconditioned SVRG. https://arxiv.org/abs/1602.02350

[15] Robert Gower, Nicolas Le Roux, and Francis Bach. Tracking the gradients using the Hessian: A new look at variance reducing stochastic methods. In International Conference on Artificial Intelligence and Statistics, pages 707–715. PMLR, 2018. https://arxiv.org/abs/1710.07462

[16] Murray R., Demmel J., Mahoney M. W., Erichson N. B., Melnichenko M., Malik O. A., Grigori L., Luszczek P., Dereziński M., Lopes M. E., Liang T., Luo H., Dongarra J. Randomized Numerical Linear Algebra: A Perspective on the Field With an Eye to Software. https://arxiv.org/abs/2302.11474

[17] T. Sarlos. “Improved approximation algorithms for large matrices via random projections”. In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS). FOCS ’06. USA: IEEE Computer Society, 2006, pp. 143–152. isbn: 0769527205. DOI: 10.1109/FOCS.2006.37

[18] Hitczenko P., Kwapień S. On the Rademacher series. Probability in Banach Spaces. Progress in Probability. T. 35. C. 31–36. DOI: 10.1007/978-1-4612-0253-0_2

[19] D. P. Woodruff. “Sketching as a tool for numerical linear algebra”. In: Found. Trends Theor. Comput. Sci. 10.1–2 (Oct. 2014), pp. 1–157. https://arxiv.org/abs/1411.4357

[20] V. Rokhlin and M. Tygert. “A fast randomized algorithm for overdetermined linear least-squares regression”. In: Proceedings of the National Academy of Sciences 105.36 (Sept. 2008), pp. 13212–13217. DOI: 10.1073/pnas.080486910

[21] A. Ben-Israel and T.N.E. Greville. Generalized Inverses: Theory and Applications. SpringerVerlag, New York, 2003. DOI: 10.1007/b97366

[22] M. E. Lopes, S. Wang, and M. Mahoney. “Error estimation for randomized least-squares algorithms via the bootstrap”. In: Proceedings of the 35th International Conference on Machine Learning (ICML). Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 3217–3226. https://arxiv.org/abs/1803.08021

[23] D. M. Kane and J. Nelson. “Sparser Johnson-Lindenstrauss Transforms”. In: Proceedings of the 2012 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). https://arxiv.org/abs/1012.1577

[24] Lloyd N. Trefethen и David Bau III. “Numerical Linear Algebra”. (1997) DOI: https://doi.org/10.1137/1.9780898719574

[25] C. C. Paige and M. A. Saunders. “LSQR: an algorithm for sparse linear equations and sparse least squares”. In: ACM Trans. Math. Softw. 8.1 (Mar. 1982), pp. 43–71. DOI: 10.1145/355984.355989

Probabilistic methods of linear algebra and Big Data

Vladimir Agishev, Galina Goremykina

***Abstract*— This study explores the application of probabilistic methods in linear algebra to enhance computational efficiency in big data processing. As traditional data processing methods face challenges in scalability and computational complexity, this research investigates the potential of randomized approaches (RandNLA), that have been applied in many linear algebra application packages in recent years, to address these issues, specifically focusing on the benefits of dimensionality reduction.**

The research employs key probabilistic algorithms, particularly sketching techniques, to significantly reduce computational costs without processing entire datasets. Various test datasets were utilized to demonstrate the performance of these algorithms, with a special emphasis on least squares problems, which are prevalent in optimization tasks.

The results indicate that the implementation of probabilistic methods leads to considerable acceleration in computational processes, achieving high performance with minimal loss of accuracy. Furthermore, the study highlights refinement techniques that enhance the precision of approximate solutions obtained through randomized algorithms.

This research provides valuable insights into the effectiveness of probabilistic methods as a promising tool for big data analysis, particularly in areas such as machine learning and optimization. It demonstrates the potential for future research to further explore the integration of these methods into various applications, paving the way for innovative solutions to complex data challenges.

***Keywords* - Randomized Numerical Linear Algebra, Big Data, computation acceleration, matrix decompositions, randomized algorithms, numerical methods, optimization.**

REFERENCES

[1] Drineas P., Mahoney M. W., Muthukrishnan S., Sarlos T. Faster Least Squares Approximation https://arxiv.org/abs/0710.1435

[2] Avron H., Maymounkov P., Toledo S. Blendenpik: Supercharging LAPACK's Least-Squares Solver https://doi.org/10.1137/090767911

[3] Mahoney M. W. Randomized algorithms for matrices and data https://arxiv.org/abs/1104.5557

[4] N. Benjamin Erichson, Peng Zheng, Krithika Manohar, Steven L. Brunton, J. Nathan Kutz, Aleksandr Y. Aravkin. Sparse Principal Component Analysis via Variable Projection. https://arxiv.org/abs/1804.00341

[5] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. "Fixed-rank approximation of a positive-semidefinite matrix from streaming data". In: Advances in Neural Information Processing Systems. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. https://arxiv.org/abs/1706.05736

[6] S. Voronin and P.-G. Martinsson. "Efficient algorithms for CUR and interpolative matrix decompositions". In: Advances in Computational Mathematics 43.3 (Nov. 2016), pp. 495–516. https://arxiv.org/abs/1412.8447

[7] Y. Dong and P.-G. Martinsson. Simpler is better: A comparative study of randomized algorithms for computing the CUR decomposition. 2021. https://arxiv.org/abs/2104.05877

[8] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. https://arxiv.org/abs/0909.4061

[9] Erichson N. B., Voronin S., Brunton S. L., Kutz J. N. Randomized Matrix Decompositions Using R. https://arxiv.org/abs/1608.02148

[10] Ootomo H., Yokota R. Mixed-Precision Random Projection for RandNLA on Tensor Cores https://arxiv.org/abs/2304.04612

[11] Filip Hanzely, Konstantin Mishchenko, and Peter Richtárik. SEGA: Variance reduction via gradient sketching. Advances in Neural Information Processing Systems, 31, 2018. https://arxiv.org/abs/1809.03054

[12] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. Fetchsgd: Communication-efficient federated learning with sketching. In International Conference on Machine Learning, pages 8253–8265. PMLR, 2020. https://arxiv.org/abs/2007.07682

[13] Deanna Needell, Rachel Ward, and Nati Srebro. Gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. Advances in neural information processing systems. https://arxiv.org/abs/1310.5715

[14] Alon Gonen, Francesco Orabona, and Shai Shalev-Shwartz. Solving ridge regression using sketched preconditioned SVRG. https://arxiv.org/abs/1602.02350

[15] Robert Gower, Nicolas Le Roux, and Francis Bach. Tracking the gradients using the Hessian: A new look at variance reducing stochastic methods. In International Conference on Artificial Intelligence and Statistics, pages 707–715. PMLR, 2018. https://arxiv.org/abs/1710.07462

[16] Murray R., Demmel J., Mahoney M. W., Erichson N. B., Melnichenko M., Malik O. A., Grigori L., Luszczek P., Dereziński M., Lopes M. E., Liang T., Luo H., Dongarra J. Randomized Numerical Linear Algebra: A Perspective on the Field With an Eye to Software. https://arxiv.org/abs/2302.11474

[17] T. Sarlos. "Improved approximation algorithms for large matrices via random projections". In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS). FOCS '06. USA: IEEE Computer Society, 2006, pp. 143–152. isbn: 0769527205. DOI: 10.1109/FOCS.2006.37

[18] Hitczenko P., Kwapień S. On the Rademacher series. Probability in Banach Spaces. Progress in Probability. T. 35. C. 31–36. DOI: 10.1007/978-1-4612-0253-0_2

[19] D. P. Woodruff. "Sketching as a tool for numerical linear algebra". In: Found. Trends Theor. Comput. Sci. 10.1–2 (Oct. 2014), pp. 1–157. https://arxiv.org/abs/1411.4357

[20] V. Rokhlin and M. Tygert. "A fast randomized algorithm for overdetermined linear least-squares regression". In: Proceedings of the National Academy of Sciences 105.36 (Sept. 2008), pp. 13212–13217. DOI: 10.1073/pnas.080486910

[21] A. Ben-Israel and T.N.E. Greville. Generalized Inverses: Theory and Applications. SpringerVerlag, New York, 2003. DOI: 10.1007/b97366

[22] M. E. Lopes, S. Wang, and M. Mahoney. "Error estimation for randomized least-squares algorithms via the bootstrap". In: Proceedings of the 35th International Conference on Machine Learning (ICML). Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 3217–3226. https://arxiv.org/abs/1803.08021

[23] D. M. Kane and J. Nelson. "Sparser Johnson-Lindenstrauss Transforms". In: Proceedings of the 2012 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). https://arxiv.org/abs/1012.1577

[24] Lloyd N. Trefethen и David Bau III. "Numerical Linear Algebra". (1997) DOI: https://doi.org/10.1137/1.9780898719574

[25] C. C. Paige and M. A. Saunders. "LSQR: an algorithm for sparse linear equations and sparse least squares". In: ACM Trans. Math. Softw. 8.1 (Mar. 1982), pp. 43–71. DOI: 10.1145/355984.355989