СОДЕРЖАНИЕ

Том 61, номер 5, 2021 год

ОБЩИЕ ЧИСЛЕННЫЕ МЕТОДЫ

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ

УРАВНЕНИЯ В ЧАСТНЫХ ПРОИЗВОДНЫХ

МАТЕМАТИЧЕСКАЯ ФИЗИКА

ИНФОРМАТИКА

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 691–695

ОБЩИЕ ЧИСЛЕННЫЕ МЕТОДЫ

УДК 519.6

НОВЫЕ ПРИЛОЖЕНИЯ МАТРИЧНЫХ МЕТОДОВ[1)]

© 2021 г. Н. Л. Замарашкин[1,*], И. В. Оселедец[1,2,**], Е. Е. Тыртышников[1,***]

[1] 119333 Москва, ул. Губкина, 8, Институт вычислительной математики им. Г.И. Марчука РАН, Россия

[2] 121205 Москва, Большой бульвар, 30, с. 1, Сколтех, Россия

*e-mail: nikolai.zamarashkin@gmail.com

**e-mail: ivan.oseledets@gmail.com

***e-mail: eugene.tyrtyshnikov@gmail.com

Поступила в редакцию 24.11.2020 г.
Переработанный вариант 24.11.2020 г.
Принята к публикации 14.01.2021 г.

Представлен очерк современных направлений развития матричных методов и их приложений, отраженных в работах данного тематического выпуска журнала. Особое внимание уделяется методам, связанным с идеей разделения переменных, реализующим ее специальным разложениям матриц и тензоров, основанным на них алгоритмам и их применениям при решении многомерных задач вычислительной математики, анализа данных и машинного обучения. Библ. 32.

Ключевые слова: матрицы малого ранга, тензорные разложения, машинное обучение.

DOI: 10.31857/S0044466921050197

1. МАЛОПАРАМЕТРИЧЕСКИЕ ПРЕДСТАВЛЕНИЯ И ПРИБЛИЖЕНИЯ

Основное направление развития вычислительной математики и прикладного анализа данных в последние десятилетия определяется все увеличивающимся размером данных, с которыми работают алгоритмы. Сама возможность использования таких данных в численных расчетах напрямую зависит от того, найдется ли в них удобная для вычислений структура. С математической точки зрения наличие в данных скрытой структуры означает, что данные описываются моделью с относительно небольшим, приемлемым числом параметров. Естественно предположить, что для оценки параметров в малопараметрическом представлении достаточно знать не все данные, а их незначительную часть. Эта общая идея лежит в основе многих современных алгоритмов.

Анализ существующих приложений показывает, что малопараметрические приближения для матриц больших размеров основаны главным образом на том, что эти матрицы оказываются близкими к матрицам малого ранга. Недавний цикл работ (см. [1]–[3]) можно интерпретировать как попытку осмысления этого весьма общего явления.

Интерес к эффективным алгоритмам представления или приближения матриц матрицами малого ранга с использованием всех или лишь небольшого числа элементов не ослабевает вот уже более 30 лет. Среди разных подходов заметное место занимают *крестовые* приближения, которые строятся по небольшому числу столбцов и строк матрицы. Пусть $m \times n$-матрица

$$A = R + F$$

является суммой матрицы R ранга r и матрицы возмущения F, которое по спектральной норме не превосходит ε. В [4], [5] было показано, что если столбцы $m \times r$-матрицы C и строки $r \times n$-матрицы R выбраны из A таким образом, что $r \times r$-подматрица $\hat{A}$ на их пересечении имеет наибольший объем (модуль определителя) среди всех подматриц порядка r, то выполняется неравенство

$$\left\| A - C\hat{A}^{-1}R \right\|_C \le (r+1)\,\sigma_{r+1}(A) \le (r+1)\varepsilon, \qquad (1)$$

[1)]Работа выполнена при поддержке Московского центра фундаментальной и прикладной математики (соглашение 075-15-2019-1624 с Минобрнауки РФ).

где $\|\cdot\|_C$ – поэлементная (чебышёвская) норма, а $\sigma_{r+1}(A)$ – сингулярное число матрицы A, которое при их упорядочивании по невозрастанию от старшего к младшему имеет номер $r+1$. Приближения вида CGR (иногда пишут CUR) принято называть *крестовыми*, так как они строятся по элементам некоторого креста, составленного из столбцов и строк заданной матрицы.

Крестовые методы для получения тензорных разложений в формате тензорного поезда впервые были предложены в [6]. В работе [7] данного номера получены новые оценки TT-рангов приближенных тензоризаций массивов значений некоторых гладких функций.

Достоинство крестовых приближений заключается в том, что для их построения нужно знать лишь малое число элементов матрицы. Но есть и недостаток, связанный непосредственно с оценкой (1): ее прямое использование в матричных нормах приводит к множителям, зависящим от размеров матрицы. Это обстоятельство несколько ограничивало область применения данного подхода в алгоритмах анализа данных. В то же время активно применялись вероятностные методы построения весьма эффективных алгоритмов приближений вида CGR (см. [8]–[10]). Во многих интересных случаях рандомизированные алгоритмы обладали существенно лучшими оценками точности, но в них, однако, требовалось знать все элементы приближаемой матрицы.

Статья [11] в данном номере рассматривает оценки точности крестовых приближений на основе принципа обобщенного максимального объема в среднем. В ней рассматриваются наиболее употребительные матричные нормы, а полученные результаты подтверждают экспериментально наблюдаемый факт, что точность крестового метода сравнима с точностью рандомизированных алгоритмов. Это означает, что крестовый метод сохраняет свое главное преимущество – возможность строить высокоточные приближения на основе весьма лапидарной информации о матрице.

2. ЗАДАЧИ ВОСПОЛНЕНИЯ

Задача восполнения малоранговых матриц отличается от задачи приближения только в способе выбора элементов. Пусть матрица $A \in \mathbb{R}^{n\times n}$ (для упрощения обозначений рассмотрим случай квадратных матриц) допускает представление $A = R + F$ с матрицей R “малого” ранга r и “малым” по норме возмущением F. Допустим, что множество известных элементов A мало и “равномерно” распределяется в A. Можно ли восстановить матрицу A по некоторому заданному набору ее элементов?

Первоначальной мотивацией для задачи восполнения матриц послужили приложения, связанные с анализом данных: рекомендательные системы, конкурсы типа “Netflix prize”, совместная фильтрация (collabrative filtering) и др. К настоящему времени спектр приложений стал практически необозримым. Более того, восполнение матриц малого ранга является замечательным примером вычислительной задачи, решение которой опирается на глубокие теоретические результаты из разных областей математики.

В [12], [13] доказано, что для успешного восполнения число необходимых элементов не может быть меньше $\mathbb{O}(n\log n)$. Этими же авторами доказано, что при необременительных ограничениях $\mathbb{O}(n\log^2(n))$ элементов достаточно для восполнения. Предложенная конструкция использует идеи релаксации задачи к выпуклой постановке с последующим применением известных методов оптимизации. Алгоритмическая сложность методов выпуклой оптимизации мотивировала дальнейшие исследования по поиску более быстрых вычислительных процедур. В результате появились итерационные алгоритмы SVT (Singular Value Thresholding) (см. [14]) и SVP (Singular Value Projection) (см. [15]). Каждая итерация этих алгоритмов состоит из двух шагов. На первом шаге применяется безусловный градиентный метод, а на втором новое приближение проектируется на многообразие матриц ранга r. Практическая применимость этих методов ограничивается только сложностью шага проектирования, который использует сингулярное разложение матриц.

В работе [16], представленной в данном номере, предлагается на каждом шаге проектирования заменить наилучшую проекцию на приближенную, сложность вычисления которой существенно меньше сложности сингулярного разложения. Для этого применяются крестовые приближения на основе принципа максимального объема. Важно заметить, что для SVP-алгоритма высокая точность приближений малого ранга во фробениусовой норме является критически важным свойством, определяющим сходимость метода. В теоретической части работы показано, что свойства сходимости методов изменятся незначительно. В качестве быстрых приближений

использовались крестовое приближение на основе принципа обобщенного максимального объема (см. [17]) и вероятностные алгоритмы малоранговых приближений (см. [18]). В вычислительных экспериментах на матрицах порядка 1000 достигалось ускорение в сотни раз. Особую ценность данной работе дает описание важных деталей программной реализации алгоритма, таких как адаптивный выбор шага в градиентном методе и адаптивная процедура набора ранга. Без них SVP-алгоритм существенно проигрывает как в скорости вычислений, так и в устойчивости.

В приложениях часто есть дополнительная информация о восполняемой малоранговой матрице (см. [19], [20]), которую было бы полезно учесть при разработке методов восполнения. Например, могут быть известны подпространства, в которых лежат линейные оболочки строк и столбцов. Такие задачи называют задачами *восполнения со сторонней информацией* (side information). Пусть d_1 и d_2 – размерности подпространств, которые содержат пространства строк и столбцов соответственно (очевидно, что $r \le \min\{d_1, d_2\}$). Будем считать, что базисы пространств задаются матрицами $X \in \mathbb{R}^{m\times d_1}$ и $Y \in \mathbb{R}^{d_2\times n}$. В этом случае число элементов, необходимое для восстановления, имеет вид $\mathbb{O}\left(\log(N)\right)$ (см. [21]). Запишем задачу восполнения в виде $A = XWY$, где A – восполняемая матрица, а $W \in \mathbb{R}^{d_1\times d_2}$ – искомая матрица предсказательной модели. В работе [22] данного номера матрица W представляется в факторизованном виде $W = UV$ с матрицами $U \in \mathbb{R}^{d_1\times r}$ и $V \in \mathbb{R}^{r\times d_2}$, разреженными по строкам и столбцам соответственно. Предложена модификация алгоритма восполнения, учитывающая разреженную структуру факторов. При определенных условиях новый алгоритм позволяет дополнительно снизить требования к числу известных элементов. Последнее является критически важным для приложений, где получение необходимой информации затруднено.

3. МАТРИЧНЫЕ И ТЕНЗОРНЫЕ МЕТОДЫ В МАШИННОМ ОБУЧЕНИИ

Подавляющее большинство современных моделей машинного обучения основано на глубинных искусственных нейросетях, которые представляют собой композицию линейных преобразований и поточечных нелинейных преобразований. Линейные преобразования параметризуются матрицами (“весами”), что дает прямую связь с матричным анализом. В частности, можно использовать методы малоранговой аппроксимации матриц и тензоров для сжатия моделей машинного обучения: по заданным весам строятся их аппроксимации, и возникает новая модель, которая содержит меньшее число параметров, но при этом приближает предыдущую. Даже если точности такой сжатой модели недостаточно, то ее можно *дообучить*, используя полученные с помощью матричных и/или тензорных разложений представления. Такой подход стал очень популярным. В данном выпуске тематике сжатия посвящена статья [23], в ней предложена идея использования аппроксимации не параметров модели, а промежуточных значений – активаций. Оказывается, что используя скелетное разложение матриц, составленное из векторов активаций, можно существенно сжать различные модели машинного обучения. Следует отметить, что матричные и тензорные методы сжатия нейросетевых моделей (они называются “факторизационными”) стали отдельным направлением в построении быстрых и компактных моделей машинного обучения и уже используются в коммерческих программных пакетах.

Важным направлением развития матричных и тензорных методов является решение задач оптимизации, где неизвестные естественным образом представляются в виде двумерных или многомерных массивов. Такие оптимизационные задачи не всегда сводятся к классическим, но могут быть решены с использованием специальных оптимизационных методов, где на каждом шаге неободимо использовать эффективные и устойчивые матричные алгоритмы. В частности, подобные постановки возникают в рекомендательных системах, анализе социальных сетей и анализе естественных языков. Иногда даже возникает необходимость по-новому взглянуть на классические алгоритмы матричного анализа с точки зрения методов оптимизации. В данном выпуске этой тематике посвящена статья [22]. В [24] тензорные разложения используются для постановки модели о разделении смесей. Классических тензорных разложений в этом случае оказывается недостаточно, и вводится новое представление. В этом случае задача исследователя состоит в подборе правильного представления, или разработке новой модели факторизации входного тензора, и решении возникающей задачи оптимизации.

Таким образом, матричные методы активно развиваются, применяются в машинном обучении как для создания новых моделей, так и для сжатия существующих. Устойчивые матричные алгоритмы дают основу для более устойчивых методов машинного обучения (в частности, ис-

пользование ортогональных матриц позволяет в ряде случаев решить проблему “затухающего градиента”). При этом высокая надежность и скорость существующих матричных и тензорных разложений позволяют успешно использовать их для решения целого ряда многомерных задач. Интересным направлением видится также развитие новых оптимизационных методов для решения задач с ограничениями на свойства неизвестных матриц и тензоров, в том числе, с использованием методов стохастической оптимизации и теории игр.

4. ДРУГИЕ ПРИЛОЖЕНИЯ И ЗАКЛЮЧЕНИЕ

Традиционные применения матричных методов связаны с решением уравнений математической физики. Это могут быть как сложные многомерные уравнения, такие как уравнение Беллмана (тензорный метод для его решения рассмотрен в [31]), так и решение классических задач, например, электростатики многочастичных систем, с существенно более высокой скоростью (см. [32]). Статья [25] данного номера посвящена весьма актуальной проблеме численного решения объемных интегральных уравнений на неравномерных сетках. При дискретизации таких задач возникают плотные матрицы огромных размеров, которые, однако, обладают скрытой структурой – показано, что они допускают приближенную факторизацию в виде произведения разреженных матриц и многоуровневой тёплицевой матрицы. Этот факт позволяет эффективно выполнять умножение матрицы на вектор и применять итерационные методы. В [26] изучаются алгебры, появившиеся в результате исследования различных обобщений специфики тёплицевых матриц.

В целом ряде задач требуется вычисление матричной экспоненты. В статье [27] данного тематического номера рассматриваются некоторые вопросы, связанные с применением подпространств Крылова при вычислении эспоненты от несимметричной матрицы. В [28] предлагается новый алгоритм вычисления собственных векторов несимметричных трехдиагональных матриц. В [29] предлагаются и исследуются новые алгоритмы для решения нелинейной проблемы собственных значений. В [30] дается полезный обзор методов экстраполяции Шэнкса и их приложений к ускорению итерационных процессов.

Методы матричного анализа и прикладной линейной алгебры имеют огромное значение для развития наук и технологий. Они обсуждаются на многочисленных семинарах и являются основной темой некоторых серийных конференций. В России – это, прежде всего, регулярная международная конференция “Матричные методы в математике и приложениях”, которая обычно проводится в Москве на базе Института вычислительной математики им. Г.И. Марчука РАН и Сколковского университета науки и технологий. Часть результатов, представленных в данном тематическом выпуске, были анонсированы в докладах 5-й конференции этой серии, состоявшейся в августе 2019 г.

Авторы этого очерка и одновременно редакторы данного тематического выпуска выражают особую благодарность Сергею Александровичу Матвееву, взявшему на себя основную часть труда по его подготовке.

СПИСОК ЛИТЕРАТУРЫ

1. *Udell M., Townsend A.* Why are big data matrices approximately low rank? // SIAM J. Math. Data Sci. 2019. V. 1. No. 1. P. 144–160.
2. *Beckermann B., Townsend A.* On the singular values of matrices with displacement structure // SIAM J. Matrix Analys. Appl. 2017. V. 38. No. 4. P. 1227–1248.
3. *Townsend A., Wilber H.* Near-Optimal Column-Based Matrix Reconstruction // Linear Algebra Appl. 2018. V. 548. P. 19–41.
4. *Goreinov S., Tyrtyshnikov E., Zamarashkin N.* A theory of pseudoskeleton approximations // Linear Algebra Appl. 1997. V. 261. No. 1–3. P. 19–41.
5. *Goreinov S., Tyrtyshnikov E.* The maximal-volume concept in approximation by low-rank matrices // Contemporary Math. 2001. V. 280. P. 47–52.
6. *Oseledets I., Tyrtyshnikov E.* TT-cross approximation for multidimensional arrays // Linear Algebra Appl. 2010. V. 432. P. 70–88.
7. *Высоцкий Л.И.* О TT-рангах приближенных тензоризаций некоторых гладких функций // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5.
8. *Boutsidis C., Drienas P., Magdon-Ismail M.*, Near-Optimal Column-Based Matrix Reconstruction // SIAM J. Comput. 2014. V. 43. No. 2. P. 183–202.

9. *Boutsidis C., Woodruff D.P.* Optimal CUR matrix decompositions // Proceed. 46th Ann. ACM Symp. Theory Comput. 2014. P. 353–362.

10. *Deshpande A., Rademacher L.* Efficient volume sampling for row/column subset selection // 51st Ann. Symp. Foundat. Comput. Sci. 2010. P. 329–338.

11. *Замарашкин Н.Л., Осинский А.И.* О точности крестовых и столбцовых малоранговых MAXVOL-приближений в среднем // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5. 09. V. 2. No. 1. P. 183–202.

12. *Candes E.J., Tao T.* The Power of Convex Relaxation: Near-Optimal Matrix Completion // IEEE Transact. Inform. Theory. 2009. V. 56. No. 5. P. 2053–2080.

13. *Recht B.* A Simpler Approach to Matrix Completion // J. Machine Learn. Res. 2011. V. 12. P. 3413–3430.

14. *Cai J.-F., Candes E.J., Z. Shen Z.* A Singular Value Thresholding Algorithm for Matrix Completion // SIAM J. Optimizat. 2010. V. 20. No. 4. P. 1956–1982.

15. *Meka R., Jain P., Dhillon I.S.*, Guaranteed Rank Minimization via Singular Value Projection // Proceed. 23rd Inter. Conf. Neural Informat. Proc. Syst. 2010. V. 1. No. 1–3. P. 937–945.

16. *Лебедева О.С., Осинский А.И., Петров С.В.* Приближенные алгоритмы малоранговой аппроксимации в задаче восполнения матрицы на случайном шаблоне // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5.

17. *Osinsky A., Zamarashkin N.* Pseudoskeleton approximations with better accuracy estimates // Linear Algebra Applicat. 2018. V. 537. P. 221–249.

18. *Tropp J.A., Halko N., Martinsson P.G.* Finding structures with randomness: probabilistic algorithms for constructing approximate matrix decompositions // SIAM Rev. 2011. V. 53. No. 2. P. 217–288.

19. *Guo Y.* Convex Co-Embedding for Matrix Completion with Predictive side information // Proceed. 31th AAAI Conf. Artific. Intelligence Symp. Theory Comput. 2017.

20. *Wang H., Wei Y., Cao M., Xu M., Wu W., Xing E.P.* Deep Inductive Matrix Completion for Biomedical Interaction Prediction // IEEE Inter. Conf. Bioinformatics and Biomedicine (BIBM). 2019. P. 520–527.

21. *Xu M., Jin R., Zhou Z.-H.* Speedup matrix completion with side information: Application to multi-label learning // Adv. Neural Informat. Proc. Syst. 2013. P. 2301–2309.

22. *Буркина М., Назаров И., Панов М., Федонин Г., Широких Б.* Индуктивное восстановление матриц с отбором признаков // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5.

23. *Гусак Ю.В., Даулбаев Т.К., Оселедец И.В., Пономарев Е.С., Чихоцкий А.С.* Малоранговое представление нейронных сетей // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5.

24. *Оселедец И.В., Харюк П.В.* Моделирование структуры данных с помощью блочного тензорного разложения: разложение объединенных тензоров и вариационное блочное тензорное разложение как параметризованная модель смесей // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5.

25. *Самохин А.Б., Тыртышников Е.Е.* Численный метод решения объемных интегральных уравнений на неравномерной сетке // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5.

26. *Боццо Э., Диедда П., ди Фиоре К.* Замкнутые относительно J-сопряжения алгебры и формулы смещения // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5.

27. *Бочев М.А.* Точный перезапуск метода подпространства Крылова “сдвиг–обращение” для вычисления действия экспоненты несимметричных матриц // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5.

28. *Ван Дорен П., Лаудадио Т., Мастронарди Н.* Вычисление собственных векторов несимметричных трехдиагональных матриц // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5.

29. *Гандер В.* Новые алгоритмы для решения нелинейной проблемы собственных значений // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5.

30. *Брезински К., Редиво-Дзалья М.* Методы экстраполяции Шэнкса и их приложения // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5.

31. *Бойко А.И., Оселедец И.В., Феррер Г.* T-QI: ускоренная итерация функции ценности в формате тензорного поезда для задач стохастического оптималььного управления // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5.

32. *Хоромская В.Х., Хоромский Б.Н.* Перспективы численного моделирования с использованием тензорных разложений для моделирования электростатики в многочастичных системах // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 5.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 696–705

ОБЩИЕ ЧИСЛЕННЫЕ МЕТОДЫ

УДК 519.61

ЗАМКНУТЫЕ ОТНОСИТЕЛЬНО *J*-СОПРЯЖЕНИЯ АЛГЕБРЫ И ФОРМУЛЫ СМЕЩЕНИЯ[1)]

© 2021 г. Э. Боццо[1,*], П. Диедда[2,], К. ди Фиоре[3,***]**

[1] *Удине, Отделение математических, физических и компьютерных наук, университет Удине, Италия*

[2] *Падуа, Отделение математики им. Туллио Леви–Чивиты, Падуанский университет, Италия*

[3] *Рим, Отделение математики, Римский университет "Tor Vergata", Италия*

**e-mail: enrico.bozzo@uniud.it*

***e-mail: deidda@math.unipd.it*

****e-mail: difiore@mat.uniroma2.it*

Поступила в редакцию 24.11.2020 г.
Переработанный вариант 24.11.2020 г.
Принята к публикации 14.01.2021 г.

Вводится понятие J-эрмитовости матрицы как обобщение классической эрмитовости, и, более общо, замкнутости множества матриц относительно J-сопряжения. Многие известные алгебры, такие как нижние и верхние тёплицевы, циркулятные и τ-матрицы, а также некоторые алгебры, чья размерность больше размера матриц, оказываются замкнутыми относительно J-сопряжения. В качестве приложения мы обобщаем теоремы о смещенных разложениях в предположении о замкнутости алгебры относительно J-сопряжения. Несмотря на то что предположение о структуре не является необходимым для алгебр, порожденных одной матрицей, было показано, что вышеупомянутый результат приводит к формулам смещения низкой сложности для алгебр, которые не порождаются одним элементом. Библ. 11.

Ключевые слова: матричные алгебры, матричный анализ, формулы смещения.

DOI: 10.31857/S0044466921050057

1. ВВЕДЕНИЕ

Некоторые важные классы матриц характеризуются в терминах такого оператора $\mathfrak{C}$, что для любой матрицы A из заданного класса ранг матрицы $\mathfrak{C}(A)$ (который называется рангом смещения матрицы A) достаточно мал. Формула смещения позволяет выразить исходную матрицу A как сумму произведений матриц, принадлежащих двум структурированным матричным алгебрам, причем количество слагаемых совпадает с рангом смещения матрицы A.

Известны формулы, позволяющие разложить семейства матриц, расширяющие классы тёплицевых, ганкелевых, сумм тёплицевых и ганкелевых матриц, а также матриц Вандермонда, в короткие суммы произведений нижних и верхних тёплицевых, циркулянтных, и τ-матриц. Соответствующие формулы смещения можно найти в [1]–[8].

Матричные алгебры, использующиеся в формулах смещения, обладают свойствами симметричности, эрмитовости или персимметричности. Для обобщения перечисленных свойств мы вводим и исследуем понятия J-эрмитовости матрицы и замкнутости матричной алгебры относительно J-сопряжения. Они естественно обобщают классическое понятие эрмитовости матриц. В качестве приложения мы приводим теорему о формулах смещения с использованием алгебр, замкнутых относительно J-сопряжения. Она обобщает теоремы о формулах смещения в [1], [2], поскольку предположения о симметричности или персимметричности матриц алгебры содержатся в предположении о замкнутости относительно J-сопряжения. На самом деле, для алгебр, чьи элементы суть полиномы от фиксированной циклической матрицы (матрицы, мини-

[1)]К. ди Фиоре выражает благодарность за частичное финансирование итальянскому исследовательскому институту INdAM–GNCS и проекту Отдела передового опыта Министерства просвещения, университетов и научных исследований Италии, предоставленному Отделению математики Римского университета "Tor Vergata", CUP E83C18000100006.

мальный многочлен которой совпадает с характеристическим), в [3] было показано, что ограничения на структуру алгебры не нужны. Однако из нашей теоремы следуют формулы низкой сложности, связанные с алгебрами, которые не порождаются одним элементом. Более того, алгебры, использованные в [9], имеют размерность большую, чем размер матриц, что может быть использовано для уменьшения числа слагаемых в формуле смещения. Эти алгебры являются замкнутыми относительно J-сопряжения, но не порождаются одним элементом. Мы планируем обобщить теорему для работы с подобными алгебрами.

В разд. 2 вводится понятие J-эрмитовости, и показывается, каким образом оно обобщает классическую эрмитовость. В разд. 3 мы исследуем свойства алгебр, замкнутых относительно J-сопряжения, и показываем, что многие известные алгебры тёплицевых, циркулянтных, антициркулянтных и $\tau_{\varepsilon,\varphi}$-матриц являются замкнутыми относительно J-сопряжения. В разд. 4 теоремы из [1], [2] обобщаются на случай замкнутых относительно J-сопряжения алгебр, а известные формулы смещения получаются как следствия. В разд. 5 подведены итоги работы.

2. J-ЭРМИТОВЫ МАТРИЦЫ

Определение 1. Пусть J – унитарная эрмитова комплексная матрица размера $n \times n$. Комплексная матрица A размера $n \times n$ называется J-эрмитовой, если $JAJ = A^h$.

Отметим, что A является J-эрмитовой тогда и только тогда, когда $JA = A^h J = (JA)^h$. Иначе говоря, A является J-эрмитовой тогда и только тогда, когда JA эрмитова.

Рассмотрим несколько примеров. Любая эрмитова матрица является J-эрмитовой для $J = I$. Если J является матрицей перестановки следующего вида (обменной матрицей)

$$J = \begin{pmatrix} 0 & \dots & 0 & 1 \\ \vdots & & \iddots & 0 \\ 0 & \iddots & & \vdots \\ 1 & 0 & \dots & 0 \end{pmatrix},$$

то J-эрмитовость матрицы является обобщением понятия персимметричности для комплексных матриц. Например, матрица Z является J-эрмитовой для вышеописанной матрицы J:

$$Z = \begin{pmatrix} 0 & \mathbf{i} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & & & \vdots \\ \vdots & & 0 & \mathbf{i} & \ddots & & \vdots \\ \vdots & & & \ddots & -\mathbf{i} & \ddots & \vdots \\ \vdots & & & & 0 & \ddots & 0 \\ \vdots & & & & & 0 & -\mathbf{i} \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \end{pmatrix}. \tag{1}$$

J-эрмитовы матрицы обладают свойствами, аналогичными свойствам эрмитовых матриц. Например, если две эрмитовы матрицы коммутируют, то их произведение также является эрмитовым. Обобщение этого утверждения на случай J-эрмитовых матриц тривиально.

Предложение 1. *Если две J-эрмитовых матрицы коммутируют, то их произведение J-эрмитово.*

Известно, что любая матрица представляется в виде суммы эрмитовой и косоэрмитовой матриц. Следующее предложение содержит обобщение этого факта, использующее J-эрмитовость.

Предложение 2. *Любая матрица H представима в виде $H = H_1 + iH_2$, где H_1 и H_2 – J-эрмитовы.*

Доказательство. Матрицы

$$H_1 = \frac{H + JH^h J}{2}, \quad H_2 = \frac{H - JH^h J}{2i},$$

являются J-эрмитовыми и, очевидно,

$$H = H_1 + iH_2.$$

3. АЛГЕБРЫ, ЗАМКНУТЫЕ ОТНОСИТЕЛЬНО J-СОПРЯЖЕНИЯ

Определение 2. Пусть J – унитарная эрмитова комплексная $n\times n$ матрица. Множество комплексных $n\times n$ матриц H называется замкнутым относительно J-сопряжения, если $(JXJ)^h \in \mathcal{H}$ для любого $X \in \mathcal{H}$. В частном случае $J = I$ множество, замкнутое относительно J-сопряжения, называется замкнутым относительно сопряжения.

Пример 1. Рассмотрим несколько примеров.

• Очевидно, любое множество эрмитовых матриц замкнуто относительно сопряжения.

• Пусть U – унитарная матрица. Тогда множество $\mathcal{U} = \{UDU^h \,|\, D$ диагональна$\}$ является алгеброй матриц, одновременно диагонализуемых матрицей $\mathcal{U}$ (SDU-алгеброй). Алгебра U замкнута относительно сопряжения. Действительно, если $X = U\,\mathrm{diag}(\lambda_i)U^h$, то

$$X^h = U\,\mathrm{diag}(\overline{\lambda}_i)U^h \in \mathcal{U}.$$

• Пусть U – унитарная матрица, а P – симметричная матрица перестановки. Тогда множество $\mathcal{U} = \{U(D_1 + D_2P)U^h \,|\, D_1$ и D_2 диагональны$\}$ является алгеброй матриц, замкнутой относительно сопряжения. Действительно, если $X = U(D_1 + D_2P)U^h$, то

$$X^h = U(D_1^h + PD_2^hPP)U^h \in \mathcal{U},$$

• так как PD_2^hP диагональна.

• Алгебры $\tau_{\alpha,\beta}$, введенные в [2], замкнуты относительно сопряжения. Они определяются как алгебры, состоящие из полиномов от матрицы

$$T_{\alpha,\beta} = \begin{pmatrix} \alpha & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & 0 & 1 \\ 0 & \cdots & 0 & 1 & \beta \end{pmatrix},$$

для некоторых вещественных α и β.

• Если J – это обменная матрица, то каждая замкнутая относительно сопряжения персимметричная алгебра является замкнутой относительно J-сопряжения. Рассмотрим несколько примеров:

– Множество тёплицевых матриц.

– Множество верхних (нижних) треугольных тёплицевых матриц.

– Множества циркулянтных и антициркулянтных матриц.

– Матричная τ-алгебра.

– Алгебры $\mathcal{C}_e = \{C_1 + JC_2 \,|\, C_1$ и C_2 циркулянтны$\}$ и $\mathcal{S}_e = \{S_1 + JS_2 \,|\, S_1$ и S_2 антициркулянтны$\}$ (они имеют размерность больше, чем n, см. [9]).

Предложение 3. *Если H есть J-эрмитова матрица, то верно следующее:*

• *алгебра $\mathcal{H}$, порожденная матрицей H, замкнута относительно J-сопряжения;*

• *коммутант множества $\{H\}$ (множество матриц, коммутирующих с H) замкнут относительно J-сопряжения.*

Тривиальное доказательство оставляется читателю. Пусть, например, J является обменной матрицей. Рассмотрим алгебру $\mathcal{H}$, порожденную матрицей Z из (1). Поскольку Z является J-эрмитовой, алгебра $\mathcal{H}$ замкнута относительно J-сопряжения. Если Z – циклическая матрица, известно, что коммутант множества $\{Z\}$ равен $\mathcal{H}$.

Из предложения 2 легко вывести следующее полезное утверждение.

Следствие 1. Пусть $\mathcal{H}$ – алгебра, замкнутая относительно J-сопряжения. Тогда любая матрица $H \in \mathcal{H}$ представима в виде $H = H_1 + iH_2$, где H_1 и H_2 суть J-эрмитовы матрицы из $\mathcal{H}$. Более того, $\mathcal{H}$ имеет базис, состоящий из J-эрмитовых матриц.

Предложение 4. *Пусть $\mathcal{H}$ есть n-мерная коммутативная алгебра, замкнутая относительно J-сопряжения, порожденная матрицей H. Тогда найдется J-эрмитова матрица H', которая порождает алгебру $\mathcal{H}$.*

Доказательство. Рассмотрим разложение $H = H_1 + iH_2$, где H_1 и H_2 суть J-эрмитовы. Рассмотрим жорданово разложение матрицы H. H является циклической матрицей, а значит, на каждое собственное значение приходится ровно один жорданов блок:

$$XHX^{-1} = \begin{pmatrix} K_1 & 0 & \cdots & 0 \\ 0 & K_2 & \cdot & \cdot \\ \cdot & \cdot & \ddots & 0 \\ 0 & \cdots & 0 & K_m \end{pmatrix}, \quad K_i = \begin{pmatrix} \lambda_i & 1 & 0 & \cdot \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & \lambda_i \end{pmatrix}, \quad \lambda_i \neq \lambda_j.$$

Поскольку матрицы XH_1X^{-1} и XH_2X^{-1} коммутируют с XHX^{-1}, они являются блочно-диагональными матрицами с верхними треугольными тёплицевыми блоками:

$$XH_1X^{-1} = \begin{pmatrix} T_1 & 0 & \cdots & 0 \\ 0 & T_2 & \cdot & \cdot \\ \cdot & \cdot & \ddots & 0 \\ 0 & \cdots & 0 & T_m \end{pmatrix}, \quad XH_2X^{-1} = \begin{pmatrix} \tilde{T}_1 & 0 & \cdots & 0 \\ 0 & \tilde{T}_2 & \cdot & \cdot \\ \cdot & \cdot & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{T}_m \end{pmatrix}.$$

Для доказательства утверждения достаточно доказать существование таких $\alpha, \beta \in \mathbb{R}$, что $\forall i = 1, \ldots, n$ матрица $\alpha T_i + \beta \tilde{T}_i = T_{i,\alpha,\beta}$ является циклической, и $(T_{i,\alpha,\beta})_{11} \neq (T_{j,\alpha,\beta})_{11}$ $\forall i \neq j$.

Мы знаем, что матрица $T_i + j\tilde{T}_i = K_i$ циклическая для всех i. Значит, $\forall i$, для почти всех $(\alpha, \beta) \in \mathbb{R} \times \mathbb{R}$, $\alpha T_i + \beta \tilde{T}_i$ — тёплицева матрица с ненулевыми числами над главной диагональю. Эта матрица является циклической, так как $T_{i,\alpha,\beta} - (T_{i,\alpha,\beta})_{11} I$ содержит $(n-1) \times (n-1)$ невырожденную подматрицу.

Более того, поскольку $(K_i)_{11} \neq (K_j)_{11}$ $\forall i \neq j$, для почти всех $(\alpha, \beta) \in \mathbb{R} \times \mathbb{R}$ выполнено $(\alpha T_i + \beta \tilde{T}_i)_{11} \neq (\alpha T_j + \beta \tilde{T}_j)_{11}$, так что существует такое $(\alpha, \beta) \in \mathbb{R} \times \mathbb{R}$, что матрица $\alpha XH_1X^{-1} + \beta XH_2X^{-1}$ циклическая.

Теперь мы вводим важное для вывода формул смещения понятие.

Определение 3. Пусть $\mathcal{H}$ есть n-мерная матричная алгебра.

• Говорят, что вектор $w \in \mathbb{C}^n$ характеризует $\mathcal{H}$ строчно, если $\forall x \in \mathbb{C}^n$ существует единственная матрица $\mathcal{H}_{(w)}(x) \in \mathcal{H}$, удовлетворяющая равенству $w^t \mathcal{H}_{(w)}(x) = x^t$.

• Говорят, что вектор $\exists v \in \mathbb{C}^n$ характеризует $\mathcal{H}$ столбцово, если $\forall y \in \mathbb{C}^n$ существует единственная матрица, $\mathcal{H}^{(v)}(y) \in \mathcal{H}$, удовлетворяющая равенству $\mathcal{H}^{(v)}(y)v = y$.

Два типа характеризации для замкнутых относительно J-сопряжения алгебр связаны между собой следующим тривиальным утверждением.

Предложение 5. *Если $\mathcal{H}$ — алгебра, замкнутая относительно J-сопряжения, а v характеризует $\mathcal{H}$ столбцово, то $J^t\bar{v}$ характеризует $\mathcal{H}$ строчно.*

Для коммутативных алгебр выполнено также следующее элементарное равенство.

Предложение 6. *Пусть $\mathcal{H}$ есть n-мерная коммутативная матричная алгебра:*

• *если w характеризует $\mathcal{H}$ строчно, то $y^t \mathcal{H}_{(w)}(x) = x^t \mathcal{H}_{(w)}(y)$ $\forall x, y \in \mathbb{C}^n$;*

• *если v характеризует $\mathcal{H}$ столбцово, то $\mathcal{H}^{(v)}(y)x = \mathcal{H}^{(v)}(x)y$ $\forall x, y \in \mathbb{C}^n$.*

Теорема 2.5 в [10] утверждает, что если $\mathcal{H}$ порождена циклической матрицей H, то найдутся два вектора, которые характеризуют ее строчно и столбцово соответственно. Однако не каждая матричная алгебра размерности n порождается одним элементом, даже если существуют векторы, которые характеризуют ее. Далее приведен пример такой алгебры, причем каждый ее элемент не является циклическим.

Пример 2:

$$\mathcal{H} = \left\{ X = \begin{pmatrix} T & Q \\ 0 & T \end{pmatrix} \middle| T, Q - \text{верхние треугольные тёплицевы } \frac{n}{2} \times \frac{n}{2} \text{ матрицы} \right\}.$$

Для каждой матрицы $H \in \mathcal{H}$, $H - H_{11}I$ – матрица ранга не больше, чем $n-2$, так что собственное значение H_{11} имеет геометрическую кратность не меньше, чем 2, а значит, матрица H не может быть циклической. Если J – обменная матрица, то $\mathcal{H}$ замкнута относительно J-сопряжения, характеризуется строчно вектором e_1 и столбцово вектором e_n.

Если $\mathcal{U}$ – SDU-алгебра, то легко получить явное выражение для матриц $\mathcal{U}^{(v)}(z)$ и $\mathcal{U}_{(w)}(z)$.

Предложение 7. *Пусть $\mathcal{U}$ есть SDU-алгебра.*

- *Если v – такой вектор, что $(U^h v)_i \neq 0 \ \forall i$, то*

$$\mathcal{U}^{(v)}(z) = U \operatorname{diag}(U^h z) \operatorname{diag}(U^h v)^{-1} U^h.$$

- *Если w – такой вектор, что $(w^t U)_i \neq 0 \ \forall i$, то*

$$\mathcal{U}_{(w)}(z) = U \operatorname{diag}(U^t z) \operatorname{diag}(U^t w)^{-1} U^h.$$

Если $L = \operatorname{diag}(U^h v)^{-1} U^h$, то собственные значения матрицы $\mathcal{U}^{(v)}(z)$ совпадают с компонентами вектора Lz. Таким образом, эрмитовы элементы алгебры $\mathcal{U}$ образуют множество $\{\mathcal{U}^{(v)}(z) \,|\, Lz - \text{вещественный вектор}\}$. Этот результат можно обобщить на случай алгебр, замкнутых относительно J-сопряжения.

Предложение 8. *Пусть $\mathcal{H}$ есть n-мерная коммутативная алгебра, замкнутая относительно J-сопряжения, а v – вектор, характеризующий $\mathcal{H}$ столбцово. Тогда существует такая невырожденная матрица L, зависящая от v, что*

$$\{H \in \mathcal{H} \,|\, H - J\text{-эрмитова}\} = \{\mathcal{H}^{(v)}(x) \,|\, LJx - \text{вещественный вектор}\}.$$

Доказательство. Для начала рассмотрим J-эрмитову матрицу $H \in \mathcal{H}$. Если $Hv = x$, то $H = \mathcal{H}^{(v)}(x)$. Из предложения 1 вытекает, что число

$$v^h J \mathcal{H}^{(v)}(y) \mathcal{H}^{(v)}(x) v = v^h J \mathcal{H}^{(v)}(y) x = v^h (\mathcal{H}^{(v)}(y))^h J x = y^h J x$$

является вещественным для всех таких y, что матрица $\mathcal{H}^{(v)}(y)$ есть J-эрмитова.

Ввиду следствия 1, $\mathcal{H}$ допускает базис $\{\mathcal{H}^{(v)}(y_i)\}_{i=1}^n$, состоящий из J-эрмитовых матриц. Векторы $y_1, \ldots, y_n$, очевидно, линейно независимы. Определив матрицу L равенствами

$$e_i^t L = y_i^h,$$

получаем, что L невырождена, а LJx веществен.

Чтобы доказать обратное включение, рассмотрим матрицу $\mathcal{H}^{(v)}(x)$, не являющуюся J-эрмитовой. Пользуясь следствием 1, мы можем записать равенство

$$\mathcal{H}^{(v)}(x) = \mathcal{H}^{(v)}(x_1) + i\mathcal{H}^{(v)}(x_2),$$

где $\mathcal{H}^{(v)}(x_2)$ – ненулевая J-эрмитова матрица. Тогда $LJx = LJx_1 + iLJx_2$ не веществен, так как векторы LJx_2 и LJx_1 вещественны, а LJx_2 отличен от нуля.

В случае когда w характеризует алгебру строчно, можно доказать существование такой невырожденной матрицы M, зависящей от w, что $\mathcal{H}_{(w)}(x)$ есть J-эрмитова тогда и только тогда, когда $x^t JM$ веществен.

4. ТЕОРЕМЫ СМЕЩЕНИЯ ДЛЯ ЗАМКНУТЫХ ОТНОСИТЕЛЬНО J-СОПРЯЖЕНИЯ АЛГЕБР

В качестве приложения в этом разделе мы докажем обобщения теорем из [1], [2], использующихся в разложениях смещения замкнутых относительно J-сопряжения алгебр.

В [3] используются алгебры, порожденные циклической матрицей. Приведем основной результат.

Теорема 1 (см. 6 в [3]). *Пусть H – циклическая матрица, порождающая алгебру $\mathcal{H}$, вектор v характеризует $\mathcal{H}$ столбцово, а w характеризует $\mathcal{H}$ строчно. Рассмотрим алгебру $\mathcal{K}$, порожденную матрицей $K = H + vw^t$. Для любой матрицы A, если*

$$AH - HA = \sum_{i=1}^{k} x_i y_i^t,$$

то

$$A = \sum_{i=1}^{k} \mathcal{H}^{(v)}(x_i)\mathcal{K}_{(w)}(y_i) + \mathcal{H}^{(v)}(Av).$$

Ниже мы получим аналогичный результат для алгебр, замкнутых относительно J-сопряжения. Для алгебр, порожденных циклическим элементом, наш результат слабее, чем теорема 1. Однако наша теорема подходит для работы с алгебрами, которые не порождаются одной матрицей, как в примере 2 и в [9]. Более того, из нижеследующей теоремы можно вывести значительную часть известных формул смещения.

Предположим, что $\mathcal{H}$ и $\mathcal{K}$ – две n-мерные коммутативные матричные алгебры, замкнутые относительно J-сопряжения. Более того, пусть $H \in \mathcal{H}$, $K \in \mathcal{K}$ суть J-эрмитовы матрицы, $v \in \mathbb{C}^n$, причем $H + vv^hJ = K$. Наконец, пусть вектор v характеризует $\mathcal{H}$ столбцово, а вектор $w = (v^hJ)^t$ характеризует $\mathcal{K}$ строчно.

Замечание 1. Для SDU-алгебр, если $U[z] + Uxx^hU^h = \mathcal{V}[z']$, где $x_i \neq 0\ \forall i$, а Ux характеризует алгебру $\mathcal{U}$, то унитарная матрица $W = U^hV$ такова, что $U^hVD(z')V^hU = D(z) + xx^h$. Следовательно, столбцы $\{w_i\}$ матрицы W являются собственными векторами матрицы $D(z) + xx^h$:

$$\lambda_i w_i = (D(z) + xx^h)w_i,$$
$$w_i = (x^h w_i)(\lambda_i I - D(z))^{-1}x.$$

Следовательно,

$$W_{ij} = (x^h w_j)\frac{x_i}{(\lambda_j - z_i)};$$

а значит, W – унитарная матрица типа Коши. Таким образом, для рассмотрения SDU-алгебр, удовлетворяющих условиям теоремы 2, придется исследовать унитарные матрицы типа Коши.

Определение 4. J-разложением матрицы $A \in \mathbb{C}^{n\times n}$ мы будем называть разложение

$$A = \sum_{m=1}^{r} x_m y_m^t,$$

где векторы x_m, y_m таковы, что $\mathcal{H}^{(v)}(x_m)$ и $\mathcal{K}_{(w)}(y_m)$ являются J-эрмитовыми.

Из следствия 1 и предложения 8 следует, что существуют такие L, M (зависящие от алгебр $\mathcal{K}$, $\mathcal{H}$ и векторов v, w), что у матрицы A имеется J-разложение тогда и только тогда, когда $LJAJM$ – вещественная матрица. Следовательно, если у A существует J-разложение, то также существует "оптимальное" J-разложение, в котором число слагаемых совпадает с рангом матрицы A (достаточно рассмотреть вещественное скелетное разложение матрицы $LJAJM = \sum e x_m \tilde{y}_m^t$, а потом разложение матрицы $A = \sum JL^{-1}\tilde{x}_m\tilde{y}_m^tM^{-1}J$).

Лемма 1. *Если у A существует J-разложение, то у $AH - HA$ также существует J-разложение.*

Доказательство. Рассмотрим J-разложение $A = \sum_{i=1}^{l} x_i y_i^t$ и скелетное разложение коммутатора

$$AH - HA = \sum_{i=1}^{l} (x_i y_i^t H - H x_i y_i^t).$$

Чтобы показать, что мы получили J-разложение, достаточно доказать, что $\mathcal{H}^{(v)}(Hx_i)$ и $\mathcal{K}_{(w)}(H^t y_i)$ являются J-эрмитовыми. Из равенства

$$\mathcal{H}^{(v)}(Hx_i) = H\mathcal{H}^{(v)}(x_i)$$

следует, что матрица $\mathcal{H}^{(v)}(Hx_i)$ – J-эрмитова, поскольку она равна произведению коммутирующих J-эрмитовых матриц.

Теперь для матрицы $\mathcal{K}_{(w)}(H^t y_i)$ запишем равенство:

$$\begin{aligned}\mathcal{K}_{(w)}(H^t y_i) &= \mathcal{K}_{(w)}((y_i^t(K - vv^h J))^t) = \mathcal{K}_{(w)}((y_i^t K)^t) - \mathcal{K}_{(w)}((y_i^t vv^h J)^t) = \\ &= \mathcal{K}_{(w)}(y_i)K - (y_i^t v)\mathcal{K}_{(w)}(w) = \mathcal{K}_{(w)}(y_i)K - (y_i^t v)I.\end{aligned}$$

Матрица $\mathcal{K}_{(w)}(y_i)K$, очевидно, J-эрмитова. Для завершения доказательства осталось показать, что число $(y_i^t v)$ вещественно:

$$(y_i^t v) = w^t \mathcal{K}_{(w)}(y_i)v = w^t JJ\mathcal{K}_{(w)}(y_i)JJv = v^h(\mathcal{K}_{(w)}(y_i))^h \tilde{w} = \overline{(y_i^t v)}.$$

Значит, матрица $(y_i^t v)I = (y_i^t v)JJ$ также J-эрмитова.

Лемма 2 (см. [1], [11]). *Пусть $A \in \mathcal{C}^{n\times n}$ и $\mathfrak{C}_H(A) = AH - HA = \sum_{i=1}^{k} x_i y_i^t$. Тогда*

$$\sum_{i=1}^{k} x_i^t \tilde{H}^t y_i = 0 \quad \forall \tilde{H} \in \mathcal{H}.$$

Доказательство:

$$\begin{aligned}\sum_{i=1}^{k} x_i^t \tilde{H}^t y_i &= \sum_{i=1}^{k}\sum_{m,j=1}^{n} x_{i\,m}\tilde{H}^t_m y_{i\,j} = \sum_{m,j=1}^{n}\sum_{i=1}^{k} x_{i\,m}\tilde{H}^t_m y_{i\,j} = \sum_{m,j=1}^{n} \tilde{H}_{jm}\sum_{i=1}^{k}(x_i y_i^t)_{mj} = \sum_{m,j=1}^{n} \tilde{H}_{jm}(AH - HA)_{mj} = \\ &= \operatorname{tr}(\tilde{H}(AH - HA)) = \operatorname{tr}(\tilde{H}AH - H\tilde{H}A) = 0,\end{aligned}$$

где последнее равенство следует из совпадения характеристических многочленов матриц $\tilde{H}AH$ и $H\tilde{H}A$.

Теорема 2. *Пусть $\mathcal{H}$ и $\mathcal{K}$ суть n-мерные коммутативные матричные алгебры, замкнутые относительно J-сопряжения. Пусть $H \in \mathcal{H}$, $K \in \mathcal{K}$ суть J-эрмитовы матрицы, $\exists v \in \mathbb{C}^n$, причем $H + vv^h J = K$, вектор v характеризует $\mathcal{H}$ столбцово, а вектор $w = (v^h J)^t$ характеризует $\mathcal{K}$ строчно.*

1. *Для любой $A \in \mathbb{C}^{n\times n}$, если $\mathfrak{C}_H(A) = AH - HA = \sum_{i=1}^{k} x_i y_i^t$, то*

$$A = \sum_{i=1}^{k} \mathcal{H}^{(v)}(x_i)\mathcal{K}_{(w)}(y_i) + C,$$

где C — матрица, коммутирующая с H.

2. *Если H является циклической, то $C = \mathcal{H}^{(v)}(Av)$.*

Доказательство. Матрица C коммутирует с H тогда и только тогда, когда

$$\mathfrak{C}_H(A) = \mathfrak{C}_H\left(\sum_{i=1}^{k} \mathcal{H}^{(v)}(x_i)\mathcal{K}_{(w)}(y_i)\right).$$

Рассмотрим подробнее правую часть последнего равенства:

$$\sum_{i=1}^{k}\mathcal{H}^{(v)}(x_i)\mathcal{K}_{(w)}(y_i)H - H\sum_{i=1}^{k}\mathcal{H}^{(v)}(x_i)\mathcal{K}_{(w)}(y_i) =$$
$$= \sum_{i=1}^{k}\mathcal{H}^{(v)}(x_i)\mathcal{K}_{(w)}(y_i)(K - vv^hJ) - \sum_{i=1}^{k}\mathcal{H}^{(v)}(x_i)H\mathcal{K}_{(w)}(y_i) =$$
$$= \sum_{i=1}^{k}\mathcal{H}^{(v)}(x_i)(vv^hJ)\mathcal{K}_{(w)}(y_i) - \sum_{i=1}^{k}\mathcal{H}^{(v)}(x_i)\mathcal{K}_{(w)}(y_i)vv^hJ =$$
$$= \sum_{i=1}^{k}x_iy_i^t - \sum_{i=1}^{k}\mathcal{H}^{(v)}(x_i)\mathcal{K}_{(w)}(y_i)vv^hJ = \mathfrak{S}_H(A) - \sum_{i=1}^{k}\mathcal{H}^{(v)}(x_i)\mathcal{K}_{(w)}(y_i)vv^hJ.$$

Значит, для завершения доказательства первой части достаточно показать, что

$$\sum_{i=1}^{k}\mathcal{H}^{(v)}(x_i)\mathcal{K}_{(w)}(y_i)v = 0. \tag{2}$$

Из следствия 1 следует, что матрицы $\mathcal{H}^{(v)}(x_i)$, $\mathcal{K}_{(w)}(y_i)$ могут быть представлены в виде линейных комбинаций J-эрмитовых матриц:

$$\mathcal{H}^{(v)}(x_i) = \mathcal{H}^{(v)}(\varphi_i) + i\mathcal{H}^{(v)}(\psi_i),$$
$$\mathcal{K}_{(w)}(y_i) = \mathcal{K}_{(w)}(\xi_i) + i\mathcal{K}_{(w)}(\eta_i).$$

Теперь имеем

$$AH - HA = \sum_{j=1}^{k}(\varphi_j\xi_j^t - \psi_j\eta_j^t) + i\sum_{j=1}^{k}(\varphi_j\eta_j^t + \psi_j\xi_j^t).$$

Положим

$$C_1 = \sum_{j=1}^{k}(\varphi_j\xi_j^t - \psi_j\eta_j^t), \quad C_2 = \sum_{j=1}^{k}(\varphi_j\eta_j^t + \psi_j\xi_j^t).$$

Из предложения 8 и леммы 1 следует, что

$$C_1 = A_1H - HA_1, \quad C_2 = A_2H - HA_2, \tag{3}$$

где A_1 и A_2 определены в виде

$$A_1 = JL^{-1}\operatorname{Re}(LJAJM)M^{-1}J, \quad A_2 = JL^{-1}\operatorname{Im}(LJAJM)M^{-1}J.$$

Значит, для завершения доказательства первой части достаточно показать равенства:

$$J\sum_{j=1}^{k}(\mathcal{H}^{(v)}(\varphi_j)\mathcal{K}_{(w)}(\xi_j) - \mathcal{H}^{(v)}(\psi_j)\mathcal{K}_{(w)}(\eta_j))v = 0, \tag{4}$$

$$J\sum_{j=1}^{k}(\mathcal{H}^{(v)}(\varphi_j)\mathcal{K}_{(w)}(\eta_j) + \mathcal{H}^{(v)}(\psi_j)\mathcal{K}_{(w)}(\xi_j))v = 0. \tag{5}$$

Запишем подробнее m-й столбец в (4):

$$e_m^tJ\sum_{j=1}^{k}(\mathcal{H}^{(v)}(\varphi_j)\mathcal{K}_{(w)}(\xi_j) - \mathcal{H}^{(v)}(\psi_j)\mathcal{K}_{(w)}(\eta_j))v =$$
$$= e_m^tJ\sum_{j=1}^{k}(J(\mathcal{H}^{(v)}(\varphi_j))^hJ\mathcal{K}_{(w)}(\xi_j) - J(\mathcal{H}^{(v)}(\psi_j))^hJ\mathcal{K}_{(w)}(\eta_j))v =$$

$$= \sum_{j=1}^{k} (\varphi_j^h(\mathscr{H}^{(v)}(e_m))^h(\mathscr{K}_{(w)}(\xi_j))^h J - \psi_j^h(\mathscr{H}^{(v)}(e_m))^h(\mathscr{K}_{(w)}(\eta_j))^h J)v =$$

$$= \sum_{j=1}^{k} \varphi_j^h(\mathscr{H}^{(v)}(e_m))^h\overline{\xi}_j - \sum_{j=1}^{k} \psi_j^h(\mathscr{H}^{(v)}(e_m))^h\overline{\eta}_j = 0.$$

Для преобразований мы использовали предложение 6, а последнее равенство вытекает из (3) и леммы 2. Аналогичным образом преобразуется m-я строка в (5), а значит, первая часть теоремы доказана.

Если матрица H — циклическая, то коммутант множества $\{H\}$ совпадает с алгеброй $\mathscr{H}$, а значит, C принадлежит алгебре $\mathscr{H}$,

$$C = \mathscr{H}^{(v)}(\rho) = A - \sum_{i=1}^{k} \mathscr{H}^{(v)}(x_i)\mathscr{K}_{(w)}(y_i).$$

Значит, используя (2), получаем выражение для ρ:

$$\rho = \left(A - \sum_{i=1}^{k} \mathscr{H}^{(v)}(x_i)\mathscr{K}_{(w)}(y_i)\right)v = Av.$$

Теорема доказана.

Теперь мы хотим показать, что, с учетом примера 1, из теоремы 2 можно вывести некоторые известные формулы.

1. Формулы Гейдера в [7], использующие циркулянтные и треугольные тёплицевы матрицы:

$$H = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \cdot & \ddots & \ddots & \cdot & \vdots \\ \vdots & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \ddots & 1 \\ 0 & 0 & \cdots & \cdot & 0 \end{pmatrix}, \quad K = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \cdot & \ddots & \ddots & \cdot & \vdots \\ \vdots & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \ddots & 1 \\ 1 & 0 & \cdots & \cdot & 0 \end{pmatrix}, \quad K - H = e_n e_n^t J,$$

где J — обменная матрица.

2. Формулы Гохберга–Ольшевского в [4], использующие циркулянтные и антициркулянтные матрицы:

$$H = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \cdot & \ddots & \ddots & \cdot & \vdots \\ \vdots & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \ddots & 1 \\ -1 & 0 & \cdots & \cdot & 0 \end{pmatrix}, \quad K = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \cdot & \ddots & \ddots & \cdot & \vdots \\ \vdots & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \ddots & 1 \\ 1 & 0 & \cdots & \cdot & 0 \end{pmatrix}, \quad K - H = 2e_n e_n^t J.$$

3. Формулы Гохберга–Семенцула в [5], использующие верхние и нижние треугольные тёплицевы матрицы:

$$H = \begin{pmatrix} 0 & 0 & \cdot & \cdots & 0 \\ -1 & \ddots & \ddots & \cdot & \vdots \\ \vdots & \cdot & \cdot & \cdot & \cdot \\ -1 & \cdot & \cdot & \ddots & 0 \\ -1 & -1 & \cdots & -1 & 0 \end{pmatrix}, \quad K = \begin{pmatrix} 1 & 1 & \cdot & \cdots & 1 \\ 0 & \ddots & \ddots & \cdot & \vdots \\ \vdots & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \ddots & 1 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}, \quad K - H = ee^t.$$

4. Формулы Боццо–ди Фиоре в [2], использующие $\tau_{\varepsilon,\varphi}$-алгебры с $\varepsilon, \varphi = 0, 1, -1$:

$$H, K = T_{\varepsilon\varphi} = \begin{pmatrix} \varepsilon & 1 & 0 & \cdots & 0 \\ 1 & 0 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & 1 \\ 0 & \cdots & 0 & 1 & \varphi \end{pmatrix},$$

где $J = I$ – единичная матрица.

5. ЗАКЛЮЧЕНИЕ

Мы ввели понятия J-эрмитовости и замкнутости множества матриц относительно J-сопряжения. В качестве приложения мы доказали теорему о формулах разложения смещения, которая обобщает и структурирует известные результаты. Поскольку эта теорема применима к алгебрам, отличным от алгебр полиномов от фиксированной матрицы, она может привести к новым формулам низкой сложности, связанным с алгебрами, не порожденными одним элементом. Также есть потенциал для обобщения на алгебры с размерностью, большей чем размер матрицы, таких как в [9]. Эти темы могут быть исследованы в будущем.

Мы с радостью отмечаем доброжелательность и эффективность организационного комитета конференции MMMA2019, а также вспоминаем множество прекрасных и важных моментов, проведенных в Москве вместе с участниками конференции, в Институте вычислительной математики им. Г.И. Марчука РАН, в Сколковском институте науки и технологий и в МГУ им. М.В. Ломоносова.

СПИСОК ЛИТЕРАТУРЫ

1. *Di Fiore C., Zellini P.* Matrix decompositions using displacement rank and classes of commutative matrix algebras // Linear Algebra and its Appl. 1995. V. 229. P. 49–99.
2. *Bozzo E., Di Fiore C.* On the use of certain matrix algebras associated with discrete trigonometric transforms in matrix displacement decomposition // SIAM J. Matrix Anal. Appl. 1995. V. 16. P. 312–326.
3. *Bozzo E.* A note on matrix displacement representation // Integral Equations and Operator Theory. 1997. V. 29. P. 368–372.
4. *Gohberg I., Olshevsky V.* Circulants, displacements and decompositions of matrices // Integral Equations and Operator Theory. 1992. V. 15. P. 730–743.
5. *Gohberg I., Semencul A.* On the inversion of finite Toeplitz matrices and their continuous analogs // Mat. Issled. 1972. V. 7. P. 201–233.
6. *Kailath T., Kung S., Morf M.* Displacement ranks of matrices and linear equations // J. Math. Anal. Appl. 1979. V. 68. P. 395–407.
7. *Gader P.D.* Displacement operator based decompositions of matrices using circulants or other group matrices // Linear Algebra Appl. 1990. V. 139. P. 111–131.
8. *Bini D., Pan V.* Polynomial and Matrix Computations, Fundamental Algorithms. Boston: Birkhäuser, 1994.
9. *Bozzo E.* Algebras of higher dimension for diplacement decompositions and computations with Toeplitz plus Hankel matrices // Linear Algebra and its Appl. 1995. V. 230. P. 127–150.
10. *Di Fiore C., Zellini P.* Matrix algebras in optimal preconditioning // Linear Algebra and its Appl. 2001. V. 335. P. 1–54.
11. *Di Fiore C.* Matrix algebras and displacement decompositions // SIAM J. Matrix Anal. Appl. 2000. V. 21. P. 646–667.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 706–722

ОБЩИЕ ЧИСЛЕННЫЕ МЕТОДЫ

УДК 519.6

ТОЧНЫЙ ПЕРЕЗАПУСК МЕТОДА ПОДПРОСТРАНСТВА КРЫЛОВА “СДВИГ–ОБРАЩЕНИЕ” ДЛЯ ВЫЧИСЛЕНИЯ ДЕЙСТВИЯ ЭКСПОНЕНТЫ НЕСИММЕТРИЧНЫХ МАТРИЦ[1)]

© 2021 г. М. А. Бочев[1,2]

[1] *125047 Москва, Миусская пл., 4, ИПМ РАН им. М.В. Келдыша, Россия*

[2] *119333 Москва, ул. Губкина, 8, ИВМ РАН им. Г.И. Марчука, Россия*

e-mail: botchev@ya.ru

Поступила в редакцию 24.12.2020 г.
Переработанный вариант 24.12.2020 г.
Принята к публикации 14.01.2021 г.

Предложен алгоритм перезапуска метода подпространства Крылова “сдвиг–обращение” для вычисления действия матричной экспоненты несимметричных матриц. Представленный метод является развитием недавно предложенного невязочно-временного перезапуска и разработан, чтобы предотвратить потерю точности, возможную в неувязочно-временном перезапуске. Наиболее затратная по вычислениям часть метода подпространства Крылова “сдвиг–обращение” – решение линейных систем со сдвинутой матрицей. Поскольку наш алгоритм перезапуска подразумевает изменение величины сдвига, мы показываем, что можно реализовать перезапуск так, чтобы единственного построения предобусловливателя (или LU разложения) было достаточно. Вычислительные эксперименты демонстрируют улучшенную точность и эффективность подхода. Библ. 44. Фиг. 6. Табл. 2.

Ключевые слова: метод подпространства Крылова “сдвиг–обращение”, экспоненциальное интегрирование по времени, процесс Арнольди, перезапуск методов подпространства Крылова.

DOI: 10.31857/S0044466921050033

1. ВВЕДЕНИЕ

Вычисление действия матричной экспоненты на заданный вектор – задача, часто возникающая в различных приложениях, таких как интегрирование по времени (см. [1]), анализ сетей (см. [2]) или редукция моделей (см. [3]). Для больших матриц методы подпространства Крылова являются важной группой методов, хорошо подходящих для этой задачи (см., например, [4]). Среди прочих методов для вычисления действия матричной экспоненты больших матриц можно выделить, например, методы, основанные на полиномах Чебышёва (см. [5]), метод масштабирования и квадратирования на основе ряда Тейлора (см. [6]) и другие. Методы подпространства Крылова для вычисления действия матричной экспоненты и других матричных функций представляют собой область активных исследований, среди недавних результатов и направлений которых отметим методы на основе рациональных подпространств Крылова (см. [7]–[11]), методы перезапуска (см. [12]–[17]) и эффективное решение задач большой размерности в разнообразных приложениях (см. [9], [18]–[21]).

Методы перезапуска позволяют ограничить число базисных векторов (размерность) подпространства Крылова, сохраняя при этом сходимость метода. Недавно предложенный невязочно-временной (НВ) перезапуск для вычисления матричной экспоненты представляется привлекательным инструментом (см. [22]) для этой цели. Одним из преимуществ этого метода перезапуска является то, что полиномиальные методы подпространства Крылова с НВ перезапуском гарантированно сходятся с требуемой точностью для любой длины перезапуска (т.е. для любой размерности подпространства) (см. [22]).

Другим свойством НВ перезапуска является то, что размерность малой спроецированной задачи, возникающей в ходе итераций, не растет с числом перезапусков, как в некоторых других

[1)]Работа выполнена при финансовой поддержке РНФ, проект № 19-11-00338.

методах перезапуска (см., например, [23, гл. 3]). Это означает, что, например, при длине перезапуска 10 размерность подпространства и размер малой спроецированной задачи не превосходят 10. Кроме того, спроецированная задача в методе подпространства Крылова с HB перезапуском представляет собой вычисление действия матричной экспоненты спроецированной матрицы. Это относительно простая задача, которая может быть эффективно решена стандартными методами линейной алгебры (см. [24]–[26]). Напротив, спроецированная задача в так называемых невязочных перезапусках (см. [12], [27]) – это система неавтономных дифференциальных уравнений (см. [12, соотношение (3)]). Хотя такая система, как правило, имеет небольшой размер, ее решение обычно более затратно по вычислениям и требует внимания, в частности, правильного подбора решателя систем дифференциальных уравнений и его параметров.

Важным классом рациональных методов подпространства Крылова (см. [11]) являются методы типа “сдвиг–обращение” (СО) (см. [7], [8]). Эти методы часто оказываются эффективными в разных приложениях, в частности, потому что они требуют решения линейных систем с одной и той же сдвинутой матрицей – в методах используется единственный сдвиг. Один из недостатков HB перезапуска, предложенного в [22], состоит в том, что точность метода подпространства Крылова СО с HB перезапуском не всегда может быть гарантирована. В данной статье предлагается перезапуск для метода подпространства Крылова СО, который является развитием HB перезапуска и позволяет получать требуемую точность вычислений. Предложенный подход работает для несимметричных матриц. Кроме того, в данной статье показано, как реализовать предложенный алгоритм эффективно, так чтобы достаточно было выполнить LU разложение или построение предобусловливателя только один раз.

Статья организована следующим образом. Предлагаемый алгоритм перезапуска, который мы называем THB (точный невязочно-временной) перезапуск, описан в разд. 2. Здесь сначала приводятся основные необходимые для изложения факты по методам подпространства Крылова (п. 2.1), затем описывается и обсуждается THB перезапуск (п. 2.2) и рассматривается, как организовать решение сдвинутых линейных систем в методе с THB перезапуском эффективно (п. 2.3). В разд. 3 представлены вычислительные эксперименты для двух тестовых задач. Заключительные выводы представлены в разд. 4.

2. ТОЧНЫЙ НЕВЯЗОЧНО-ВРЕМЕННОЙ ПЕРЕЗАПУСК

Условимся, что, если не оговорено иначе, $\|\cdot\|$ обозначает стандартную евклидову норму $\|\cdot\|_2$. По всей статье предполагаем также, что для матрицы $A \in \mathbb{R}^{n\times n}$ выполняется

$$\mathrm{Re}(x^*Ax) \geq 0 \quad \forall x \in \mathbb{C}^n, \tag{1}$$

где $\mathrm{Re}(z)$ обозначает вещественную часть $z \in \mathbb{C}$.

2.1. Методы подпространства Крылова и HB перезапуск

Предположим, что по данным $A \in \mathbb{R}^{n\times n}$, $v \in \mathbb{R}^n$ и $t > 0$ нужно вычислить действие матричной экспоненты матрицы $-tA$ на вектор v, т.е.

$$\text{вычислить} \quad y := \exp(-tA)v. \tag{2}$$

Эта задача эквивалентна решению задачи Коши

$$y'(t) = -Ay(t), \quad y(0) = v, \tag{3}$$

где, слегка пренебрегая точностью обозначений, мы используем t и как независимую переменную в (3), и для обозначения длины интервала в (2). Метод подпространства Крылова для вычисления действия матричной экспоненты можно рассматривать как галеркинскую проекцию задачи Коши (3) на подпространство Крылова

$$\mathcal{K}_k(A, v) = \mathrm{span}(v, Av, A^2v, \ldots, A^{k-1}v). \tag{4}$$

Сначала ортонормальный базис подпространства $\mathcal{K}_k(A, v)$ вычисляется обычным процессом Арнольди (или, если $A = A^{\mathrm{T}}$, процессом Ланцоша) (см. [24], [28]–[30]) и сохраняется в виде

столбцов $v_1, \ldots, v_k$ матрицы $V_k = [v_1 \ \ldots \ v_k] \in \mathbb{R}^{n\times k}$, так что выполняется так называемое разложение Арнольди:

$$AV_k = V_{k+1}\underline{H}_k = V_k H_k + h_{k+1,k}v_{k+1}e_k^{\mathrm{T}}, \tag{5}$$

где $e_k = (0, \ldots, 0, 1)^{\mathrm{T}} \in \mathbb{R}^k$, $\underline{H}_k \in \mathbb{R}^{(k+1)\times k}$ – верхняя хессенбергова матрица, а матрица $H_k \in R^{k\times k}$ состоит из первых k строк матрицы $\underline{H}_k$. После этого крыловская аппроксимация $y_k(t) \approx \exp(-tA)v$ определяется как (см. [31]–[33])

$$y_k(t) = V_k u(t), \tag{6}$$

где $u(t) : \mathbb{R} \to \mathbb{R}^k$ решает задачу Коши с проецированной матрицей $H_k = V_k^{\mathrm{T}} A V_k$:

$$u'(t) = -H_k u(t), \quad u(0) = \beta e_1. \tag{7}$$

Здесь $\beta = \|v\|$ и $e_1 = (1, 0, \ldots, 0)^{\mathrm{T}} \in \mathbb{R}^k$. Заметим, что задача Коши (7) – это галеркинская проекция задачи Коши (3) на подпространство Крылова и что $u(t)$ может быть вычислена как

$$u(t) = \exp(-tH_k)\beta e_1. \tag{8}$$

Если k не слишком велико, то (8) – предпочтительный способ вычисления u, который может быть реализован многими стандартными методами линейной алгебры (см. [24]–[26]). Вычисление $\exp(-tH_k)$ обычно является более простой операцией, чем решение системы дифференциальных уравнений в (7), что требует выбора подходящего решателя (в частности, жесткого или нежесткого) и его параметров. Метод подпространства Крылова (5)–(8) иногда называют полиномиальным методом подпространства Крылова, чтобы подчеркнуть тот факт, что вектора подпространства (4) – многочлены от A.

Естественным способом контроля (неизвестной) ошибки крыловского приближения (6) является отслеживание невязки $r_k(t)$ этого приближения $y_k(t)$ по отношению к системе дифференциальных уравнений $y' = -Ay$, а именно (см. [12], [27], [34]),

$$r_k(t) = -Ay_k(t) - y_k'(t). \tag{9}$$

Невязка $r_k(t)$ доступна в ходе процесса Арнольди и может быть вычислена по формуле (см. [12], [27], [34])

$$r_k(t) = -h_{k+1,k}(e_k^{\mathrm{T}}u(t))v_{k+1}. \tag{10}$$

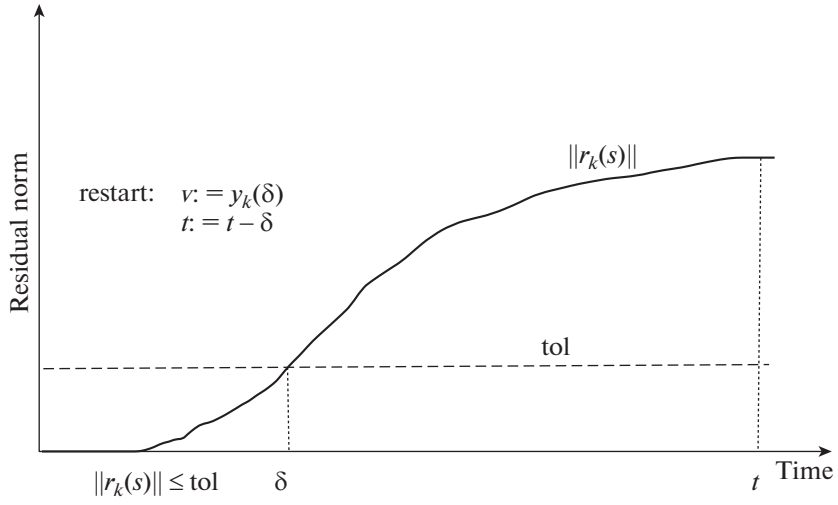
Как видим, $r_k(t)$ – скалярная функция, помноженная на v_{k+1}. Следовательно, $V_k^{\mathrm{T}} r_k(t) = 0$ для всех $t > 0$, а (6) – действительно галеркинская проекция на $\mathcal{K}_k(A, v)$. Некоторые результаты по сходимости невязки и ее связи с ошибкой могут быть найдены, например, в [22], [27].

Метод подпространства Крылова CO (“сдвиг–обращение”) (см. [7], [8]) для вычисления (2) отличается от стандартного полиномиального метода подпространства Крылова, описанного выше, тем, что подпространство Крылова строится для сдвинутой и обращенной матрицы $(I + \gamma A)^{-1}$, а не для A, где параметр $\gamma > 0$ фиксирован. Это делается для ускорения сходимости: процесс Арнольди имеет тенденцию лучше приближать наибольшие по модулю собственные числа, а для матрицы $(I + \gamma A)^{-1}$ они соответствуют малым собственным значениям матрицы A. Именно эти собственные числа важны для матричной экспоненты (компоненты, соответствующие большим собственным числам, не столь важны, они угасают экспоненциально быстро). Цена этого ускорения в сходимости – необходимость решать линейные системы с матрицей $I + \gamma A$ на каждой крыловской итерации. Разложение Арнольди (5) для сдвинутой и обращенной (CO) матрицы $(I + \gamma A)^{-1}$ принимает вид

$$(I + \gamma A)^{-1}V_k = V_{k+1}\underline{\tilde{H}}_k = V_k\tilde{H}_k + \tilde{h}_{k+1,k}v_{k+1}e_k^{\mathrm{T}}.$$

Это соотношение удобнее использовать после преобразования

$$AV_k = V_k H_k - \frac{\tilde{h}_{k+1,k}}{\gamma}(I + \gamma A)v_{k+1}e_k^{\mathrm{T}}\tilde{H}_k^{-1}, \tag{11}$$



Фиг. 1. Схема НВ перезапуска, взятая из работы [22].

где обозначение $\tilde{}$ указывает, что проекция построена для СО матрицы $(I + \gamma A)^{-1}$, а матрица H_k определяется как преобразование, обратное к сдвигу и обращению:

$$H_k = \frac{1}{\gamma}(\tilde{H}_k^{-1} - I). \tag{12}$$

Подчеркнем, что матрицы V_{k+1} и H_k здесь отличаются от матриц V_{k+1} и H_k из соотношения (5). Подробный анализ метода подпространства Крылова СО и родственных методов можно найти в (см. [7], [8], [10]).

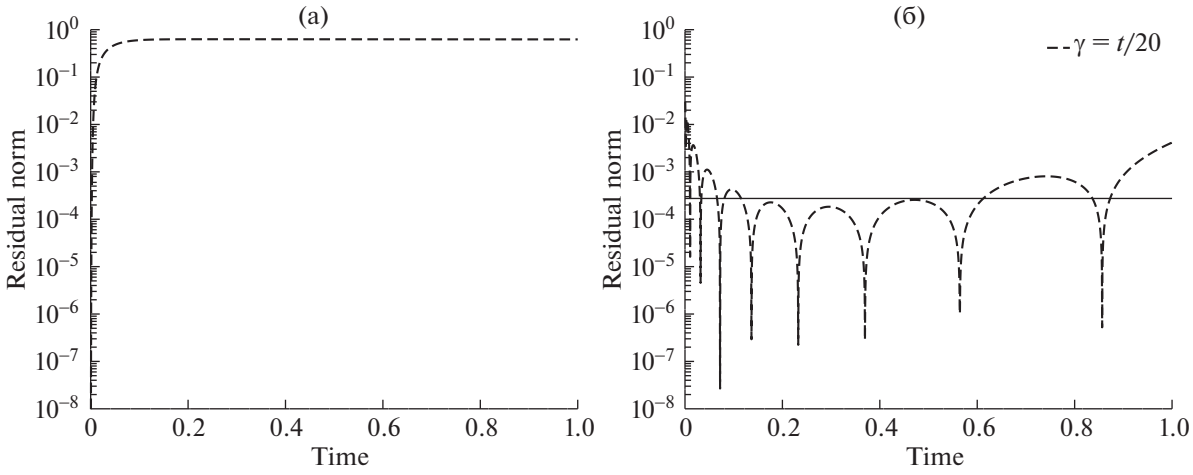
В методе подпространства Крылова СО невязка легко может быть вычислена следующим образом (см. [27]):

$$r_k(t) = \frac{\tilde{h}_{k+1,k}}{\gamma}(e_k^{\mathrm{T}}\tilde{H}_k^{-1}u(t))(I + \gamma A)v_{k+1}. \tag{13}$$

Здесь $u(t)$ – решение спроецированной задачи Коши (7), где H_k – матрица, получающаяся обратным преобразованием (12).

Величина сдвига γ обычно выбирается в соответствии с длиной t временного интервала $[0, t]$ (см. [8]), при этом часто используется величина $\gamma = t/10$. Таким образом, изменение параметра γ означает изменение t. Стандартный полиномиальный метод подпространства Крылова (5)–(8) имеет привлекательное свойство инвариантности относительно t: коль скоро V_{k+1} и $\underline{H}_k$ вычислены, они могут быть использованы для любых времен t (хотя качество аппроксимации $y_k(t) \approx y(t)$, вообще говоря, ухудшается с ростом t). К сожалению, это свойство полностью не распространяется на метод подпространства Крылова СО: в этом методе матрицы V_{k+1} и $\underline{\tilde{H}}_k$ зависят от величины γ, которая, в свою очередь, зависит от t. Тем не менее на практике можно использовать вычисленные матрицы Арнольди V_{k+1} и $\underline{\tilde{H}}_k$ для определенного диапазона t, не вычисляя их заново.

Недавно предложенный НВ перезапуск основан на том факте, что невязка как функция от t обычно является для обычного метода подпространства Крылова неубывающей функцией. Следовательно, когда выполнено максимально допустимое число $k_{\max}$ крыловских итераций (так что хранение большего числа базисных векторов Крылова и работа с ними слишком затратны), мы можем найти подынтервал $[0, \delta]$, $\delta < t$, на котором невязка уже достаточно мала по норме. Тогда мы можем перезапустить метод, полагая $v := y_{k_{\max}}(\delta)$, уменьшая временной интервал $t := t - \delta$ и выполняя следующие $k_{\max}$ крыловских итераций для задачи (2) с обновленными v и t. Схема НВ перезапуска представлена на фиг. 1.



Фиг. 2. Норма невязки $\|r_k(s)\|$ как функция времени $s \in [0,t]$, $t = 1$, для обычного полиномиального (а) и СО (б) методов подпространства Крылова после $k = 10$ крыловских итераций. Величина сдвига $\gamma = t/20$. Матрица A – дискретизированный оператор конвекции-диффузии для числа Пекле $\mathrm{Pe} = 1000$ на сетке 402×402 (см. п. 3.2). Для обоих графиков норма невязки вычислена в 2000 равноотстоящих точках интервала [0, 1]. Горизонтальная сплошная линия на графике (б) показывает геометрическое среднее $\overline{r_k} = 2.75\mathrm{e}{-04}$ значений нормы невязки $\|r_k(s_i)\|$ в 2000 точках.

2.2. THB перезапуск: идеи и алгоритм

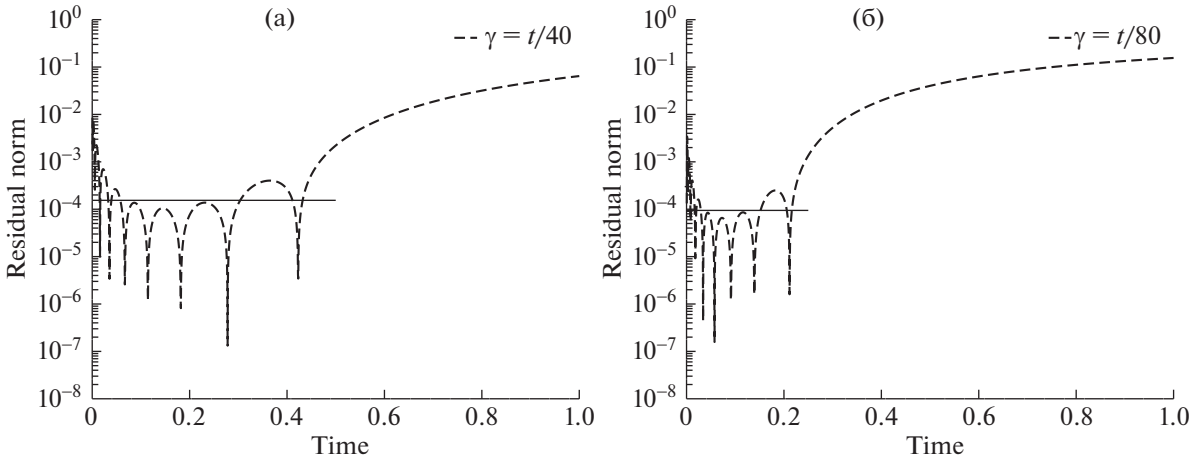
В методе подпространства Крылова СО (5)–(8) невязка как функция от t проявляет гораздо более нерегулярное поведение, чем в обычном полиномиальном методе подпространства Крылова (фиг. 2). Если для метода подпространства Крылова СО применяется НВ перезапуск, то может случиться, что такое δ, что $\|r_k(s)\|$ не превышает заданной точности для $s \in [0,\delta]$, не может быть найдено или слишком мало для практически эффективного перезапуска. Разумеется, можно устроить перезапуск, положив δ равной любой точке s, где $\|r_k(s)\|$ достаточно мала (фиг. 2). Однако гарантии, что $\min_{s\in[0,t]} \|r_k(s)\|$ не превосходит заданной точности, нет, и в таком случае перезапуск с любым $\delta \in [0,t]$ неизбежно приводит к потере точности.

В данной работе предлагается подход, позволяющий устранить этот недостаток НВ перезапуска для метода подпространства Крылова СО. Подход наш основан на следующих двух наблюдениях.

1. Поскольку γ обычно выбирается пропорционально t, меньшая величина сдвига γ означает более короткий временной интервал $[0,t]$. Для несимметричных матриц A невязка $r_k(s)$ в методе подпространства Крылова СО становится меньше по норме с уменьшением γ на некотором подынтервале $s \in [0,\delta]$, $0 < \delta < t$ (фиг. 2, 3).

2. Как уже обсуждалось выше, чтобы изменить γ в подпространстве Крылова СО, необходимо заново выполнить весь процесс Арнольди. Однако если линейная система с матрицей $I + \gamma A$ решена для определенного значения сдвига γ, то часть уже проделанной вычислительной работы может быть использована для решения линейных систем с меньшим значением сдвига $\tilde{\gamma} \le \gamma$. В частности, если для некоторого сдвига γ вычислено (разреженное) LU разложение, то оно может быть успешно использовано как предобусловливатель для решения линейных систем с новым значением сдвига, с матрицами $I + \tilde{\gamma}A$, $\tilde{\gamma} \le \gamma$ (см. утверждение 2).

На основе этих наблюдений для метода подпространства Крылова СО предлагается организовать НВ перезапуск без потери точности следующим образом. Предположим, что может быть выполнено не более $k_{\max}$ шагов процесса Арнольди или Ланцоша, так как хранение и использование более $k_{\max}$ крыловских векторов слишком затратны. Тогда выполняются итерации $k = 1, 2, \ldots, k_{\max}$ с контролем на каждой итерации нормы невязки $\|r_k(t)\|$ (см. (13)). Если норма невязки меньше заданной точности, процесс успешно оканчивается. Иначе после выполнения шага $k = k_{\max}$, значения функции $\|r_k(s)\|$ анализируются на отрезке $s \in [0,t]$. Если не может быть най-



Фиг. 3. Норма невязки $\|r_k(s)\|$ как функция времени $s \in [0, t]$, $t = 1$, для методов подпространства Крылова СО со сдвигом $\gamma = t/40$ (а) и $\gamma = t/80$ (б) после $k = 10$ крыловских итераций. Матрица A – дискретизированный оператор конвекции-диффузии для числа Пекле Pe = 1000 на сетке 402×402 (см. п. 3.2). Для обоих графиков норма невязки вычислена в 2000 равноотстоящих точках интервала [0, 1]. Горизонтальные сплошные линии показывают геометрические средние $\overline{r}_k$ значений нормы невязки, вычисленные для $s \in [0, 0.5]$ с $\overline{r}_k$ = 1.54e-04 (а) и $s \in [0, 0.25]$ с $\overline{r}_k$ = 9.44e-05 (б).

дено точки s такой, что норма $\|r_k(s)\|$ достаточно мала, то мы уменьшаем γ в два раза и повторяем шаги метода $k = 1, 2, \ldots, k_{\max}$. Затем процедура перезапуска (подробно описанная на фиг. 4) повторяется до тех пор, пока норма невязки не будет достаточно малой в конечной точке заданного временного интервала.

Если обычное или разреженное LU разложение слишком затратно, то для решения линейных систем СО может быть использован какой-либо предобусловленный итерационный метод. В этом случае в алгоритме перезапуска, представленном на фиг. 4, мы заменяем вычисление LU разложения построением предобусловливателя. Построенный предобусловливатель может быть использован для всех величин сдвига, т.е. достаточно построить предобусловливатель один раз.

Отметим, что для симметричных матриц A описанное выше поведение невязки в методе подпространства Крылова СО с уменьшением γ не наблюдается. Это подтверждается довольно точной оценкой леммы 3.1 из [8] (где полагаем $\mu = 0$ и выбираем γ пропорционально $t = \tau$).

Следующие лемма и утверждение показывают, что норма невязки $r_k(s)$ метода подпространства Крылова СО является функцией, ограниченной по времени. Оценка, дающая ограничение, зависит от γ.

Лемма 1. *Пусть для* $A \in \mathbb{R}^{n\times n}$ *выполняется соотношение* (1) *и пусть* H_k *– матрица, полученная в методе подпространства Крылова СО* (*см.* (11), (12)). *Тогда существует такая константа* $\omega_k \geq 0$, *что*

$$\|\exp(-tH_k)\| \leq e^{-t\omega_k}. \tag{14}$$

Доказательство. Пусть $\omega = \min_{x\in\mathbb{C}^n, \|x\|=1} \mathrm{Re}(x^*Ax)$. Известно (см., например, [35, Теорема 2.4]), что

$$\mathrm{Re}(x^*Ax) \geq \omega \quad \forall x \in \mathbb{C}^n \Leftrightarrow \|\exp(-tA)\| \leq e^{-t\omega}.$$

В силу (1) эти два эквивалентные соотношения выполняются для некоторой константы $\omega \geq 0$. Кроме того, эти соотношения эквивалентны неравенству (см. [35, теорема 2.13])

$$\left\|(I + \gamma A)^{-1}\right\| \leq \frac{1}{1 + \gamma\omega},$$

```
% Даны: A ∈ ℝ^{n×n}, v ∈ ℝ^n, t > 0, k_max и tol > 0
convergence := false
γ_changed := false
вычислить LU разложение LU := I + γA
while (not(convergence) and t > 0)
    β := ||v||, v_1 := v/β
    for k = 1,..., k_max
        if γ_changed
            решить (I + γA)w = v_k итерационно,
                с предобусловливателем LU
        else
            решить (I + γA)w = v_k LU разложением
        end
        for i = 1,..., k
            h̃_{i,k} := w^T v_i, w := w − h̃_{i,k} v_i
        end
        h_{k+1,k} := ||w||
        H_k := (1/γ)(H̃_k^{-1} − I)
        вычислить u(s_j), ||r_k(s_j)||, s_j = jt/3, j = 1, 2, 3
        resnorm := max_j ||r_k(s_j)||
        if resnorm ≤ tol and k > 1
            convergence := true
            прервать цикл for k = ...
        elseif k = k_max
            % -- перезапуск на шаге k_max
            вычислить ||r_k(s_j)||, s_j = jt/500, j = 1,..., 500
            r_min := min_j ||r_k(s_j)||
            if r_min > tol
                δ := 0
                γ := γ/2
                γ_changed := true
            else
                δ := max{s_j | ||r_k(s_j)|| ≤ tol}
            end
            u := exp(−δH_k)e_1, v := V_k(βu)
            t := t − δ
        end
        v_k+1 := w/h_{k+1,j}
    end
end
y_k := V_k(βu(s_3))
```

Фиг. 4. Алгоритм THB перезапуска метода подпространства Крылова CO. Вычисляется приближение $y_k(t) \approx \exp(-tA)v$, для невязки $r_k(t)$ которого $\|r_k(s)\| \le \text{tol}$ для $s = t/3$, $s = 2t/3$ и $s = t$.

которое выполняется для всех $\gamma > 0$ и всех $\omega \in \mathbb{R}$ при условии, что $1 + \gamma\omega > 0$. Пусть $\gamma > 0$ так же, как и в методе подпространства Крылова CO. Определим

$$\omega_k := \frac{1}{\gamma}\left(\left\|\tilde{H}_k\right\|^{-1} - 1\right)$$

так, чтобы $\left\|\tilde{H}_k\right\| = 1/(1 + \gamma\omega_k)$. Получаем

$$\frac{1}{1+\gamma\omega_k} = \left\|\tilde{H}_k\right\| = \left\|V_k^{\mathrm{T}}(I+\gamma A)^{-1}V_k\right\| \le \left\|(I+\gamma A)^{-1}\right\| \le \frac{1}{1+\gamma\omega},$$

откуда следует, что $0 \le \omega \le \omega_k$. Поскольку $\tilde{H}_k = (I + \gamma H_k)^{-1}$ (см. (12)), видим, снова используя (см. [35, теорема 2.13]), что

$$\left\|(I+\gamma H_k)\right\|^{-1} = \frac{1}{1+\gamma\omega_k} \le \frac{1}{1+\gamma\omega} \Leftrightarrow \mathrm{Re}(x^* H_k x) \ge \omega_k \quad \forall x \in \mathbb{C}^k. \tag{15}$$

Это равносильно искомому неравенству (14). Доказательство завершено.

Определим функцию $\varphi(z)$ (см., например, [1]):

$$\varphi(z) = (e^z - 1)/z. \tag{16}$$

Утверждение 1. *Пусть для* $A \in \mathbb{R}^{n\times n}$ *выполняется соотношение* (1) *и пусть* $r_k(t)$ *— невязка метода подпространства Крылова СО* (13) *при решении задачи* (2). *Тогда для всех* $t \geq 0$ *справедливо*

$$\begin{gathered} r_k(t) = \beta_k(t) w_{k+1}, \quad \beta_k(t) = \frac{\tilde{h}_{k+1,k}}{\gamma} e_k^{\mathrm{T}}(I + \gamma H_k)u(t), \quad w_{k+1} = (I + \gamma A)v_{k+1}, \\ \|r_k(t)\| = |\beta_k(t)|\|w_{k+1}\|, \end{gathered} \tag{17}$$

$$|\beta_k(t)| \leq \beta\tilde{h}_{k+1,k}\left(\frac{1}{\gamma}\min\left\{t\|(I + \gamma H_k)H_k\|\varphi(-t\omega_k), \|I + \gamma H_k\|(1 + e^{-t\omega_k})\right\} + |h_{k,1}|\right), \tag{18}$$

где функция $u(t)$ *определена в* (8), (12), $\omega_k \geq 0$ *— константа из* (14), *а* $\tilde{h}_{k+1,k}$ *и* $h_{k,1}$ *— элементы матриц* $\underline{\tilde{H}}_k \in \mathbb{R}^{(k+1)\times k}$ *и* $H_k \in \mathbb{R}^{k\times k}$ *соответственно* (см. (11), (12)). *Минимум в соотношении* (18) *берется по двум элементам множества, обозначенного* $\{\ldots\}$. *Подчеркнем, что* $\underline{\tilde{H}}_k$ *в оценке выше зависит от* γ (*а следовательно, также и* H_k, $u(t)$, ω_k).

Доказательство. Соотношение (17) идентично (13) (см. (12)), а доказательство (13) можно найти в [27]. В силу (7), (12) имеем

$$|\beta_k(0)| = \frac{\tilde{h}_{k+1,k}}{\gamma}\left|e_k^{\mathrm{T}}\tilde{H}_k^{-1}u(0)\right| = \frac{\tilde{h}_{k+1,k}}{\gamma}\left|e_k^{\mathrm{T}}(I + \gamma H_k)\beta e_1\right| = \tilde{h}_{k+1,k}\left|e_k^{\mathrm{T}}H_k\beta e_1\right| = \beta\tilde{h}_{k+1,k}|h_{k,1}|.$$

Далее нетрудно проверить, что (см. (16))

$$u(t) - u(0) = (\exp(-tH_k) - I)u(0) = -tH_k\varphi(-tH_k)u(0).$$

Поэтому, учитывая $u(0) = \beta e_1$ и $\tilde{H}_k^{-1} = I + \gamma H_k$, можно оценить

$$\begin{gathered} \|(I + \gamma H_k)(u(t) - u(0)\| = t\|(I + \gamma H_k)H_k\varphi(-tH_k)u(0)\| \leq \\ \leq t\|(I + \gamma H_k)H_k\|\|\varphi(-tH_k)\|\|u(0)\| \leq \beta t\|(I + \gamma H_k)H_k\|\varphi(-t\omega_k). \end{gathered} \tag{19}$$

Здесь используется неравенство

$$\|\varphi(-tH_k)\| \leq \varphi(-t\omega_k),$$

справедливое в силу (14), (15) (см., например, [1, доказательство леммы 2.4]). Оценка (19) особенно полезна для малых t. Получим теперь альтернативную оценку, которая может быть точнее для больших t:

$$\begin{gathered} \|(I + \gamma H_k)(u(t) - u(0)\| = \|(I + \gamma H_k)(\exp(-tH_k) - I)u(0)\| \leq \\ \leq \|I + \gamma H_k\|\|\exp(-tH_k) - I\|\|u(0)\| \leq \beta\|I + \gamma H_k\|(1 + \|\exp(-tH_k)\|) \leq \\ \leq \beta\|I + \gamma H_k\|(1 + e^{-t\omega_k}), \end{gathered} \tag{20}$$

где используется неравенство (14). Из (19), (20) следует, что

$$\|(I + \gamma H_k)(u(t) - u(0)\| \leq \beta\min\left\{t\|(I + \gamma H_k)H_k\|\varphi(-t\omega_k), \|I + \gamma H_k\|(1 + e^{-t\omega_k})\right\}.$$

Это позволяет оценить

$$|\beta_k(t)| \leq |\beta_k(t) - \beta_k(0)| + |\beta_k(0)| = \frac{\tilde{h}_{k+1,k}}{\gamma}\left|e_k^{\mathrm{T}}(I + \gamma H_k)(u(t) - u(0))\right| + \beta\tilde{h}_{k+1,k}|h_{k,1}| \leq$$

$$\leq \frac{\tilde{h}_{k+1,k}}{\gamma}\|(I + \gamma H_k)(u(t) - u(0)\| + \beta\tilde{h}_{k+1,k}|h_{k,1}| \leq$$

$$\leq \frac{\tilde{h}_{k+1,k}}{\gamma}\beta\min\left\{t\|(I + \gamma H_k)H_k\|\varphi(-t\omega_k), \|I + \gamma H_k\|(1 + e^{-t\omega_k})\right\} + \beta\tilde{h}_{k+1,k}|h_{k,1}| =$$

$$= \beta\tilde{h}_{k+1,k}\left(\frac{1}{\gamma}\min\left\{t\left\|(I+\gamma H_k)H_k\right\|\varphi(-t\omega_k), \left\|I+\gamma H_k\right\|(1+e^{-t\omega_k})\right\} + \left|h_{k,1}\right|\right),$$

что и дает (18). Доказательство завершено.

Подчеркнем, что оценка (18), к сожалению, не настолько точна, чтобы отразить зависимость $\|r_k(t)\|$ от γ (см. фиг. 2 и 3). Однако следует сделать следующее

Замечание 1. Численные эксперименты показывают, что величина $|h_{k,1}|$ (вспомним, что $|\beta_k(0)| = \beta\tilde{h}_{k+1,k}\,|h_{k,1}|$), появляющаяся в (18), обычно мала, много порядков меньше, чем другой член $\frac{1}{\gamma}\min\{\ldots\}$, участвующий в правой части (18). Если $|h_{k,1}| = 0$, то $\beta_k(0) = 0$. Таким образом, оценка (18) показывает, что для любого k и для любой точности $\varepsilon > 0$ можно найти такой временной интервал $[0, \delta]$, что $\|r_k(s)\| \le \varepsilon$ для $s \in [0, \delta]$. В этом случае временной интервал можно сократить (фиг. 1), поэтому как HB, так и THB перезапуски гарантируют сходимость метода подпространства Крылова CO для любой длины перезапуска. Разумеется, указанное δ может быть слишком мало, чтобы использовать его на практике. Поэтому подстройка параметра γ, как это делается в THB перезапуске, может быть необходима для успешной работы.

2.3. Решение сдвинутых линейных систем

Покажем теперь, что LU разложение, вычисленное для матрицы $I + \gamma A$, может быть успешно использовано как предобусловливатель для сдвинутой матрицы $I + \tilde{\gamma}A$ с уменьшенным значением сдвига $\tilde{\gamma}$, $0 < \tilde{\gamma} \le \gamma$. Говоря точнее, для решений сдвинутой линейной системы

$$\mathscr{A}x = b, \quad \mathscr{A} = I + \tilde{\gamma}A,$$

будем применять предобусловливатель

$$\mathscr{M}^{-1}\mathscr{A}x = \mathscr{M}^{-1}b, \quad \mathscr{M} = I + \gamma A. \tag{21}$$

Тогда нетрудно показать (см. утверждение 2 ниже), что даже метод простых итераций с таким предобусловливанием, точнее

$$x_{m+1} = \tilde{G}x_m + \mathscr{M}^{-1}b, \quad \tilde{G} = I - \mathscr{M}^{-1}\mathscr{A}, \tag{22}$$

сходится безусловно, т.е. для спектрального радиуса $\rho(\tilde{G})$ матрицы перехода $\tilde{G}$ выполняется $\rho(\tilde{G}) < 1$. Следовательно, собственные числа предобусловленной матрицы $\mathscr{M}^{-1}\mathscr{A}$ расположены на комплексной плоскости внутри круга единичного радиуса с центром в точке $1 + 0i$, $i^2 = -1$. Это означает, что современные итерационные методы подпространства Крылова, такие как GMRES, BiCGSTAB, QMR или аналогичные им (см. [29], [30], [36]), будут успешно решать предобусловленную линейную систему (21).

Однако чем меньше значение сдвига $\tilde{\gamma}$, тем лучше обусловлена сдвинутая матрица $I + \tilde{\gamma}A$. Следовательно, для малого $\tilde{\gamma}$ может оказаться, что непредобусловленный итерационный метод сходится достаточно быстро. Поэтому в утверждении 2 нами дается условие, достаточное для того, чтобы предобусловленный метод простых итераций сходился быстрее, чем непредобусловленный метод.

Утверждение 2. *Пусть для $A \in \mathbb{R}^{n\times n}$ выполняется соотношение* (1)*, $0 < \tilde{\gamma} \le \gamma$ и пусть линейная система с матрицей $I + \tilde{\gamma}A$ решается итерационно. Тогда предобусловленный метод простых итераций* (22) *с матрицей предобусловливателя $\mathscr{M} = I + \gamma A$ сходится.*

Кроме того, пусть метод простых итераций без предобусловливания также сходится. Тогда предобусловленный метод простых итераций (22) *с матрицей предобусловливателя $\mathscr{M} = I + \gamma A$ сходится быстрее, чем метод простых итераций без предобусловливания при условии, что*

$$\frac{1}{1+\gamma\rho(A)} < \frac{\tilde{\gamma}}{\gamma}, \tag{23}$$

где $\rho(A)$ – спектральный радиус матрицы A.

Доказательство. Пусть λ – некоторое собственное число матрицы A. Тогда собственные числа предобусловленной матрицы $(I + \gamma A)^{-1}(I + \tilde{\gamma}A)$ имеют вид

$$\frac{1 + \tilde{\gamma}\lambda}{1 + \gamma\lambda} = 1 - \left(1 - \frac{\tilde{\gamma}}{\gamma}\right)\frac{\gamma\lambda}{1 + \gamma\lambda}.$$

Предобусловленный метод простых итераций сходится тогда и только тогда, когда собственные числа матрицы перехода $\tilde{G} = I - (I + \gamma A)^{-1}(I + \tilde{\gamma}A)$ по модулю меньше единицы, т.е. если

$$\left|\left(1 - \frac{\tilde{\gamma}}{\gamma}\right)\frac{\gamma\lambda}{1 + \gamma\lambda}\right| < 1.$$

Левую часть этого неравенства можно оценить сверху:

$$\left|\left(1 - \frac{\tilde{\gamma}}{\gamma}\right)\frac{\gamma\lambda}{1 + \gamma\lambda}\right| \le \frac{|\gamma\lambda|}{|1 + \gamma\lambda|} < 1,$$

где второе неравенство выполняется потому, что собственные числа матрицы A имеют неотрицательную вещественную часть (см. (1)). Следовательно, предобусловленный метод простых итераций сходится.

Далее заметим, что матрица перехода G метода простых итераций без предобусловливания – это матрица $G = I - (I + \tilde{\gamma}A) = -\tilde{\gamma}A$. Предобусловленный метод сходится быстрее, чем непредобусловленный, при условии $\rho(\tilde{G}) < \rho(G)$, т.е. если

$$\left(1 - \frac{\tilde{\gamma}}{\gamma}\right)\max_{\lambda}\left|\frac{\gamma\lambda}{1 + \gamma\lambda}\right| < \tilde{\gamma}\max_{\lambda}|\lambda| = \tilde{\gamma}\rho(A).$$

Левую часть здесь можно оценить величиной $1 - \dfrac{\tilde{\gamma}}{\gamma}$, так что неравенство выполняется, если

$$\left(1 - \frac{\tilde{\gamma}}{\gamma}\right) < \tilde{\gamma}\rho(A).$$

Легко проверить, что это неравенство равносильно (23). Доказательство завершено.

3. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

3.1. Подробности экспериментов

Мы реализовали ТНВ перезапуск так, как показано на фиг. 4, со следующими модификациями.

1. Находимое алгоритмом приближенное наименьшее значение нормы невязки зависит от числа точек, в которых вычисляется норма невязки. Поэтому это число точек, обозначенное в описании алгоритма (фиг. 4) n_{steps}, задается в соответствии с требуемой точностью `tol`:

tol	1e–06	1e–07	≤1e–08
n_{steps}	500	1000	2000

2. Если величина δ (т.е. такая величина, что $\|r_k(\delta)\| \le \text{tol}$, см. фиг. 4) остается равной нулю после двух последовательных уменьшений γ, то γ получает свое изначальное значение 4γ, помноженное на 0.8, т.е. $\gamma := 0.8 \times 4 \times \gamma$, а число тестовых точек n_{steps} удваивается. После этого обычная работа алгоритма продолжается. Это означает, что если изначально $\gamma = 1$, γ в алгоритме последовательно принимает значения 1, 0.5, 0.25, 0.8, 0.4, 0.2, 0.64, 0.32, ... Как только найдено положительное δ, т.е. перезапуск успешен, число тестовых точек n_{steps} уменьшается до 500.

3. Как только в ходе выполнения алгоритма γ меняется, пропорционально меняется и длина временного интервала, на котором по n_{steps} тестовым точкам ищется приближенное минимальное значение нормы невязки. Например, если значение γ уменьшается вдвое, то временной интервал поиска меняется от $[0, t]$ к $[0, t/2]$. Это делается потому, что $\|r_k(s)\|$ навряд ли будет мала для

$s > t/2$ (фиг. 3). Как только перезапуск оказался успешным, т.е. $\delta > 0$ и временной интервал уменьшен (строка алгоритма $t := t - \delta$), мы увеличиваем интервал поиска до $[0, t]$.

4. Последняя модификация состоит в том, что в нашей реализации итерационный метод GMRES(10) (см. [37]) может быть использован вместо LU разложения не только тогда, когда сдвиг γ уменьшен. В качестве предобусловливателя с GMRES(10) может быть использовано неполное LU разложение ILUT(ε) (см. [38, гл. 10]. Оно вычисляется один раз и используется для всех значений γ. Мы используем реализацию GMRES, доступную на сайте www.netlib.org/templates/, из [36]. Предобусловливатель применяется справа, а в качестве остановочного критерия итераций GMRES(10) при решении линейных систем "сдвиг–обращение" $(I + \gamma A)x = b$ используется такое условие на невязку res_i линейной системы:

$$\|\mathrm{res}_i\| \le \mathrm{tol}_{\mathrm{gmres}}\|b\|, \quad \text{with} \quad \mathrm{tol}_{\mathrm{gmres}} = \min\{1\mathrm{e}{-}08,\ \mathrm{tol}/10\},$$

где tol – требуемая заданная точность вычисления действия матричной экспоненты. При малых значениях γ может быть разумным отключить предобусловливатель (см. утверждение 2).

Начальное значение γ может быть задано как необязательный параметр нашей процедуры THB перезапуска. По умолчанию, если начальное значение γ не задано пользователем, оно устанавливается равным $t/20$. Заметим, что значение $\gamma = t/10$ рекомендуется в [8] для умеренных величин допустимой точности $\mathrm{tol} \approx 10^{-6}$ для симметричных матриц. Выбор начального значения $\gamma := t/20$ в THB перезапуске представляется разумным выбором, поскольку, согласно нашему опыту, оптимальное значение γ для несимметричных матриц обычно меньше, чем $t/10$.

В экспериментах, представленных ниже, в рамках метода подпространства Крылова СО мы сравниваем THB перезапуск с HB перезапуском. В работе [22] HB перезапуск был всесторонне протестирован и сравнен с другими тремя методами перезапуска: перезапуском пакета EXPOKIT (см. [39]), перезапуском Нихоффа–Хохбрук (см. [23, гл. 3]) и невязочным перезапуском (см. [12], [27]). Значения ошибок, представленные в этом разделе, получены для численного решения $y_k(t)$, как

$$\frac{\|y_k(t) - y_{\mathrm{ref}}(t)\|}{\|y_{\mathrm{ref}}(t)\|},$$

где $y_{\mathrm{ref}}(t)$ – референтное решение, вычисленное с высокой точностью функцией phiv пакета EXPOKIT (см. [39]). Численные тесты были проведены в Матлабе на линукс-компьютере с 8 процессорами Intel Xeon E5504 2.00GHz.

3.2. Задача конвекции-диффузии

В этой задаче матрица A получается стандартной пятиточечной конечно-разностной аппроксимацией оператора конвекции-диффузии, определенного для функций $u(x, y)$ с $(x, y) \in \Omega = [0,1]\times[0,1]$ и $u|_{\partial\Omega} = 0$. Оператор имеет вид

$$L[u] = -(D_1 u_x)_x - (D_2 u_y)_y + \mathrm{Pe}\left(\frac{1}{2}(v_1 u_x + v_2 u_y) + \frac{1}{2}((v_1 u)_x + (v_2 u_y)_y)\right),$$

$$D_1(x,y) = \begin{cases} 10^3 & (x,y) \in [0.25,\ 0.75]^2, \\ 1 & \text{иначе}, \end{cases} \qquad D_2(x,y) = \frac{1}{2} D_1(x,y),$$

$$v_1(x,y) = x + y, \quad v_2(x,y) = x - y,$$

где Pe – число Пекле. Здесь конвективные члены (первые производные) записаны в специальном виде так, чтобы их вклады в матрицу A представляли собой кососимметричную матрицу (см. [40]). В экспериментах использовалась равномерная сетка 802×802 и значения числа Пекле $\mathrm{Pe} = 200$ и $\mathrm{Pe} = 1000$. Размер задачи для этой сетки – $n = 800^2 = 640\,000$. Для обоих значений числа Пекле $\left\|\frac{1}{2}\left(A + A^{\mathrm{T}}\right)\right\|_2 \approx 6000$, в то время как $\left\|\frac{1}{2}\left(A - A^{\mathrm{T}}\right)\right\|_2 \approx 0.5$ для $\mathrm{Pe} = 200$ и $\left\|\frac{1}{2}\left(A - A^{\mathrm{T}}\right)\right\|_2 \approx 2.5$ для $\mathrm{Pe} = 1000$. Следовательно, в обоих случаях матрицы можно считать слабо несимметричными. Значения функции $\sin(\pi x)\sin(\pi y)$ на конечно-разностной сетке присваивались начальному вектору v, который затем нормализовывался $v := v/\|v\|$. Задавалось конечное время $t = 1$.

Таблица 1. Результаты для тестовой задачи конвекции-диффузии, сетка 802×802, конечное время $t = 1$

Метод	Точности заданная, полученная	Процессорное время, с	Число итераций (итераций GMRES(10))
Pe = 200, длина перезапуска 10			
HB, разреж. LU	1e–06, 2.50e–07	46.2	20 (–)
HB, разреж. LU	1e–08, 2.51e–07	48.0	27 (–)
THB, разреж. LU, GMRES(10)	1e–08, 1.60e–08	484	73 (962)
THB, разреж. LU, GMRES(10), найденное γ	1e–08, 1.65e–08	56.2	53 (–)
HB, GMRES(10)/ILUT	1e–08, 2.54e–07	90.5	36 (440)
THB, GMRES(10)/ILUT	1e–08, 1.85e–08	191	77 (1258)
THB, GMRES(10)/ILUT, найденное γ	1e–08, 1.51e–08	68.2	57 (342)
Pe = 1000, длина перезапуска 10			
HB, GMRES(10)/ILUT	1e–06, 3.36e–07	49.2	14 (154)
HB, GMRES(10)/ILUT	1e–08, 3.52e–07	72.9	27 (325)
THB, GMRES(10)/ILUT	1e–06, 2.43e–07	44.5	17 (136)
THB, GMRES(10)/ILUT	1e–08, 7.55e–08	110	55 (630)
THB, GMRES(10)/ILUT, найденное γ	1e–08, 7.41e–08	52.8	25 (205)

В экспериментах начальное значение сдвига γ в THB перезапуске не задавалось и было по умолчанию $t/20$, а в HB перезапуске использовалось обычное значение $\gamma = t/10$. Это не обязательно дает преимущество THB перезапуску, потому что оптимальные значения γ, находимые в THB перезапуске, все равно были меньше, чем $t/20$.

Результаты для этой тестовой задачи представлены в табл. 1. Как можно увидеть в первых двух строках таблицы, HB перезапуск не в состоянии дать меньшую ошибку при уменьшенном значении допустимой точности, несмотря на возросший объем вычислений (27 вместо 20 итераций). В то же время THB перезапуск, где используется тот же самый линейный решатель – разреженное LU разложение – справляется с поставленной задачей, хотя процессорное время и увеличивается в 10 раз (см. строку 3 табл. 1). Отметим, что процессорное время, измеренное средствами Матлаба, не всегда правильно отражает вычислительную работу. В данном случае оно не соответствует числу крыловских шагов метода (73 шага с THB перезапуском вместо 27 шагов с HB перезапуском). Причина здесь в том, что прямые методы решения линейных систем (LU разложение и оператор действия обратной матрицы "обратная дробная черта" \) реализованы в среде Матлаб весьма эффективно, чего нельзя сказать об итерационных решателях. Из-за этого процессорное время вычисления действий предобусловливателя внутри GMRES(10) оказывается значительным. Однако, как только алгоритм определил подходящее значение γ, это значение успешно может быть использовано для повторных вычислений действия матричной экспоненты: в этом случае, как видим в строке 4 табл. 1, мы получаем требуемую высокую точность при небольшом увеличении расчетного времени.

Мы также тестируем наш подход на этой задаче с итерационным линейным решателем – итерационным методом GMRES(10) с предобусловливателем ILUT($\varepsilon = 10^{-3}$). В строке 5 табл. 1 показано, что HB перезапуск в комбинации с предобусловленным GMRES требует 36 шагов Арнольди (вместо 27 шагов для HB перезапуска в комбинации с прямым методом решения, см. табл. 1, строка 2), потому что невязки в этих двух реализациях метода (с LU разложением и с методом GMRES(10)) слегка отличаются. THB перезапуск с тем же самым предобусловленным итерационным методом требует в два раза больше процессорного времени, чем HB перезапуск, но дает меньшую ошибку (см. табл. 1, строка 6). Наконец, в последней строке таблицы мы видим, что коль скоро правильное значение сдвига определено THB алгоритмом, THB перезапуск позволяет получать лучшую точность при сравнимых вычислительных затратах.

В нижней части табл. 1 представлены результаты для большего числа Пекле. При требуемой точности `tol = 1e-06` перезапуск HB дает результат с точностью `3.36e-07`, что вполне удовлетворительно. Однако при требуемой точности `tol = 1e-08` метод оказывается не в состоя-

нии получить меньшую ошибку, хотя вычислительные затраты выросли с 14 до 27 шагов. В следующих двух строках таблицы показаны результаты для THB перезапуска. THB перезапуск позволяет получить более точный результат при возросшем процессорном времени. Из последней строки табл. 1 следует, что как только оптимальное значение γ найдено, аналогичная точность может быть получена за примерно то же процессорное время.

3.3. Уравнения Максвелла в среде без потерь

Рассмотрим уравнения Максвелла в трехмерной области, заполненной непроводящей средой без источников электромагнитного поля:

$$\frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\mu}\nabla \times \mathbf{E}, \qquad \frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\varepsilon}\nabla \times \mathbf{H}. \tag{24}$$

Здесь ε и μ – относительные диэлектрическая и магнитная проницаемости соответственно, являющиеся скалярными функциями переменных (x, y, z), а магнитное поле $\mathbf{H}$ и электрическое поле $\mathbf{E}$ – неизвестные вектор-функции переменных (x, y, z, t). Краевые условия задают на границе области нулевые тангенциальные компоненты электрического поля, что физически означает идеально проводящую границу области или так называемые краевые условия “большого бака” (см. [41], [42]). Данная модельная задача взята из [42]: в пространственной области $[-6.05, 6.05]\times[-6.05, 6.05]\times[-6.05, 6.05]$, наполненной воздухом (относительная диэлектрическая проницаемость $\varepsilon_r = 1$), помещен образец из материала с относительной диэлектрической проницаемостью $\varepsilon_r = 5.0$, занимающий подобласть $[-4.55, 4.55]\times[-4.55, 4.55]\times[-4.55, 4.55]$. В образце имеются 27 сферических отверстий ($\varepsilon_r = 1$) радиуса 1.4, центры отверстий расположены в точках $(x_i, y_j, z_k) = (3.03i, 3.03j, 3.03k)$, $i, j, k = -1, 0, 1$. Задаются нулевые начальные значения для всех компонент обоих полей $\mathbf{H}$ и $\mathbf{E}$, кроме компонент x и y поля $\mathbf{E}$. Последние не равны нулю в центре области и представляют собой световой импульс.

Дискретизация по пространству центральными, разнесенными по сетке, конечными разностями (схема Йи (Yee)) приводит в системе дифференциальных уравнений вида (3). Пространственная сетка в этом тесте состоит из $40\times40\times40$ или $80\times80\times80$ ячеек Йи, так что размер задачи $n = 413\,526$ или $n = 3\,188\,646$ соответственно. После дискретизации вектор начальных значений $v \in \mathbb{R}^n$ нормализуется $v := v/\|v\|$. Сравнение результатов, полученных на этих двух пространственных сетках, показывает, что разрешение сетки достаточно для этого теста. Конечное время $t = 1$.

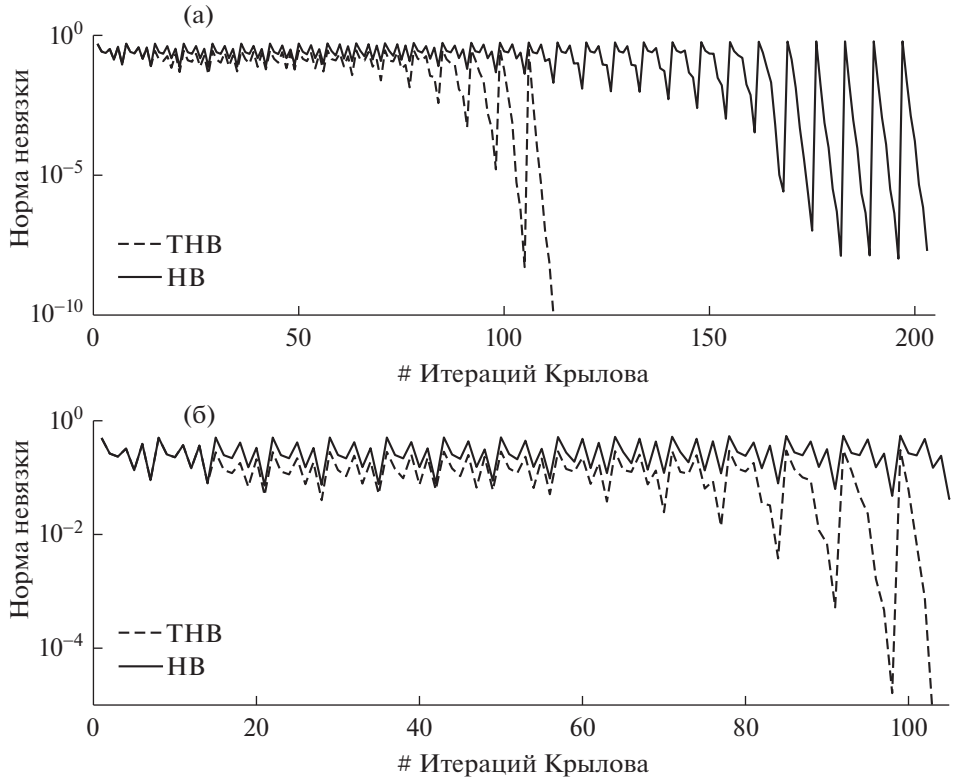
Этот тест является сложным для метода подпространства Крылова CO, потому что матрица A сильно несимметрична (можно выбрать такую диагональную матрицу D, что $D^{-1}AD$ – кососимметричная). Для сильно несимметричных задач, таких как дискретизированные уравнения Максвелла в непроводящей среде, методы подпространства Крылова CO зачастую не являются эффективными (см. [43]). Действительно, другие методы подпространства Крылова с перезапуском оказываются более эффективными в этой тестовой задаче (см. [22]). Кроме того, это – трехмерная векторная задача, где 6 переменных (компоненты x, y и z обоих полей) соответствуют каждой ячейке вычислительной сетки. Поэтому в зависимости от конкретных значений параметров задачи решать линейные системы со сдвинутой матрицей $I + \gamma A$ может быть весьма непросто. Тем не менее для данной конкретной задачи оказывается, что величина $\gamma\|A\|$ достаточно мала, так что условие (23) не выполняется и даже непредобусловленный метод простых итераций может быть успешно использован для решения сдвинутых линейных систем. В представленных здесь экспериментах для этой цели был использован итерационный метод GMRES(10). Таким образом, мы рассматриваем эту тестовую задачу, чтобы показать возможности предложенного THB перезапуска.

Опыт показывает, что для успешной работы с сильно несимметричными матрицами A методы подпространства Крылова CO должны использовать гораздо меньшие величины сдвига γ, чем обычно используемое значение $t/10$ (см. [44]). Поэтому мы задаем для γ значение $t/80 = 1/80$ в обоих методах перезапуска (при этом THB перезапуск при необходимости может уменьшить это значение). Результаты представлены в табл. 2 и на фиг. 5 и 6. THB перезапуск очевидно пре-

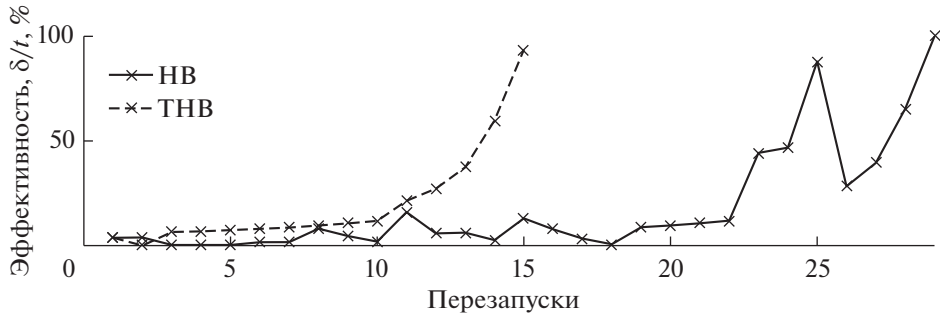
Таблица 2. Результаты для уравнений Максвелла в непроводящей среде

Метод	Точности заданная, полученная	Процессорное время, с	Число итераций (итераций GMRES(10))
Сетка $40 \times 40 \times 40$, длина перезапуска 7			
HB, GMRES(10)	1e–09, 2.96e–07	113.7	203 (1827)
THB, GMRES(10)	1e–09, 9.51e–08	48.8	112 (812)
THB, GMRES(10)	1e–10, 1.22e–08	78.1	161 (1302)
Сетка $80 \times 80 \times 80$, длина перезапуска 8			
HB, GMRES(10)	1e–09, 5.33e–07	2067	352 (4263)
THB, GMRES(10)	1e–09, 3.90e–08	785	191 (1410)

восходит HB перезапуск как по вычислительным затратам, так и по полученной точности. Потеря точности в HB перезапуске происходит на первых перезапусках, когда норма невязки оказывается больше требуемой точности по всему временному интервалу. Во избежание потери точности THB перезапуск уменьшает γ, что не только восстанавливает точность, но и приводит к более быстрому решению сдвинутых линейных систем (напомним, что при меньших значениях γ непредобусловленный GMRES сходится быстрее). Более того, уменьшенные значения γ при этом приводят к дальнейшему выигрышу эффективности в THB перезапуске. Как видно на фиг. 6, этот выигрыш получается из-за больших временных интервалов $[0, \delta]$, на которых норма невязки оказывается меньше заданной точности (напомним, что временной интервал на каждом перезапуске сокращается с $[0, t]$ до $[0, t - \delta]$).



Фиг. 5. (а) – Сходимость метода подпространства Крылова CO с HB перезапуском (сплошная линия) и THB перезапуском (штриховая линия) для уравнений Максвелла в непроводящей среде, сетка $40 \times 40 \times 40$, длина перезапуска – 7. (б) – Увеличение верхнего графика. Каждый зигзаг соответствует перезапуску.



Фиг. 6. Эффективность перезапусков, представленная как отношение сокращаемой части временного интервала δ к оставшемуся временному интервалу t. Тестовая задача: уравнения Максвелла в непроводящей среде, сетка $40 \times 40 \times 40$, длина перезапуска – 7. Эффективность 0% на втором ТНВ перезапуске означает уменьшение параметра γ.

4. ВЫВОДЫ

Предлагаемый ТНВ (точный невязочно-временной) перезапуск представляется полезным подходом для повышения эффективности методов подпространства Крылова СО ("сдвиг–обращение") при вычислении действий матричной экспоненты несимметричных матриц. Данный подход обладает всеми свойствами обычного НВ (невязочно-временного) перезапуска, а также позволяет предотвратить потерю точности при сохранении эффективности НВ перезапуска.

Можно обозначить несколько направлений дальнейших исследований. Во-первых, поиск минимума нормы невязки сейчас выполняется на равномерной сетке точек временного интервала. Этот поиск может быть организован более эффективно на неравномерной сетке, сгущающейся в областях, где норма невязки имеет локальные минимумы. Может быть разработана адаптивная процедура для построения такой сетки. Кроме того, следует изучить вопрос о том, как обобщить данный подход на симметричные матрицы. Мы надеемся заняться этими вопросами в будущем.

СПИСОК ЛИТЕРАТУРЫ

1. *Hochbruck M., Ostermann A.* Exponential integrators // Acta Numer. 2010. V. 19. P. 209–286.
2. *De la Cruz Cabrera O., Matar M., Reichel L.* Analysis of directed networks via the matrix exponential // J. of Comput. and Appl. Math. 2019. V. 355. P. 182–192. Access mode: https://doi.org/10.1016/j.cam.2019.01.015
3. *Kürschner P.* Balanced truncation model order reduction in limited time intervals for large systems // Adv. Comput. Math. 2018. V. 44. https://doi.org/10.1007/s10444-018-9608-6
4. *Frommer A., Simoncini V.* Matrix functions // Model Order Reduction: Theory, Research Aspects and Applications / Ed. by Wil H. A. Schilders, Henk A. van der Vorst, Joost Rommes. Springer, 2008. P. 275–304.
5. *Bergamaschi L., Vianello M.* Efficient computation of the exponential operator for large, sparse, symmetric matrices // Numer. Linear Algebra with Appl. 2000. V. 7. № 1. P. 27–45.
6. *Al-Mohy A.H., Higham N.J.* Computing the action of the matrix exponential, with an application to exponential integrators // SIAM J. Sci. Comput. 2011. V. 33. № 2. P. 488–511. https://doi.org/10.1137/100788860
7. *Moret I., Novati P.* RD rational approximations of the matrix exponential // BIT. 2004. V. 44. P. 595–615.
8. *van den Eshof J., Hochbruck M.* Preconditioning Lanczos approximations to the matrix exponential // SIAM J. Sci. Comput. 2006. V. 27. № 4. P. 1438–1457.
9. *Druskin V., Knizhnerman L., Zaslavsky M.* Solution of large scale evolutionary problems using rational Krylov subspaces with optimized shifts // SIAM J. on Sci. Comput. 2009. V. 31. № 5. P. 3760–3780.
10. *Güttel S.* Rational Krylov Methods for Operator Functions: Ph. D. thesis / Stefan Güttel; Technischen Universität Bergakademie Freiberg. 2010. March. www.guettel.com.
11. *Güttel S.* Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection // GAMM Mitteilungen. 2013. V. 36. № 1. P. 8–31. www.guettel.com.

12. *Celledoni E., Moret I.* A Krylov projection method for systems of ODEs // Appl. Numer. Math. 1997. V. 24. № 2–3. P. 365–378.

13. *Tal-Ezer H.* On restart and error estimation for Krylov approximation of $w = f(A)v$ // SIAM J. Sci. Comput. 2007. V. 29. № 6. P. 2426–2441. Access mode: https://doi.org/10.1137/ 040617868

14. *Afanasjew M., Eiermann M., Ernst O.G., Güttel S.* Implementation of a restarted Krylov subspace method for the evaluation of matrix functions // Linear Algebra Appl. 2008. V. 429. P. 2293–2314.

15. *Eiermann M., Ernst O.G., Güttel S.* Deflated restarting for matrix functions // SIAM J. Matrix Anal. Appl. 2011. V. 32. № 2. P. 621–641.

16. *Güttel S., Frommer A., Schweitzer M.* Efficient and stable Arnoldi restarts for matrix functions based on quadrature // SIAM J. Matrix Anal. Appl. 2014. V. 35. № 2. P. 661–683.

17. *Jawecki T., Auzinger W., Koch O.* Computable strict upper bounds for Krylov approximations to a class of matrix exponentials and ϕ-functions // arXiv preprint arXiv:1809.03369. 2018. https://arxiv.org/pdf/1809.03369.

18. *Hochbruck M., Lubich C.* Exponential integrators for quantum-classical molecular dynamics // BIT. 1999. V. 39. № 4. P. 620–645.

19. *Hochbruck M., Pažur T., Schulz A. et al.* Efficient time integration for discontinuous Galerkin approximations of linear wave equations // ZAMM. 2015. V. 95. № 3. P. 237–259. Access mode: https://doi.org/10.1002/zamm.201300306

20. *Börner R.-U., Ernst O.G., Güttel S.* Three-dimensional transient electromagnetic modeling using rational Krylov methods // Geophys. J. Internat. 2015. V. 202. № 3. P. 2025–2043.

21. *Botchev M.A., Hanse A.M., Uppu R.* Exponential Krylov time integration for modeling multifrequency optical response with monochromatic sources // J. Comput. Appl. Math. 2018. V. 340. P. 474–485. https://doi.org/10.1016/j.cam.2017.12.014

22. *Botchev M.A., Knizhnerman L.A.* ART: Adaptive residual-time restarting for Krylov subspace matrix exponential evaluations // J. Comput. Appl. Math. 2020. V. 364. № 112311. https://doi.org/10.1016/j.cam.2019.06.027

23. *Niehoff J.* Projektionsverfahren zur Approximation von Matrixfunktionen mit Anwendungen auf die Implementierung exponentieller Integratoren: Ph.D. thesis / Jörg Niehoff; Mathematisch-Naturwissenschaftlichen Fakultät der Heinrich-Heine-Universität Düusseldorf. 2006. December.

24. *Golub G.H., Van Loan C.F.* Matrix Computations. Third edition. Baltimore and London: The Johns Hopkins Univ. Press, 1996. P. 694.

25. *Moler C.B., Van Loan C.F.* Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later // SIAM Rev. 2003. V. 45. № 1. P. 3–49.

26. *Higham N.J.* Functions of Matrices: Theory and Computation. Philadelphia, PA, USA: Soc. for Industrial and Appl. Math., 2008.

27. *Botchev M.A., Grimm V., Hochbruck M.* Residual, restarting and Richardson iteration for the matrix exponential // SIAM J. Sci. Comput. 2013. V. 35. № 3. P. A1376–A1397. https://doi.org/10.1137/110820191

28. *Parlett B.N.* The Symmetric Eigenvalue Problem. SIAM, 1998.

29. *van der Vorst H.A.* Iterative Krylov methods for large linear systems. Cambridge Univ. Press, 2003.

30. *Saad Y.* Iterative Methods for Sparse Linear Systems. 2d ed. SIAM, 2003. Available from http://www-users.cs.umn.edu/~saad/books.html.

31. *Druskin V.L., Knizhnerman L.A.* Two polynomial methods of calculating functions of symmetric matrices // U.S.S.R. Comput. Maths. Math. Phys. 1989. V. 29. № 6. P. 112–121.

32. *Knizhnerman L.A.* Calculation of functions of unsymmetric matrices using Arnoldi's method // U.S.S.R. Comput. Math. Math. Phys. 1991. V. 31. № 1. P. 1–9.

33. *Hochbruck M., Lubich C.* On Krylov subspace approximations to the matrix exponential operator // SIAM J. Numer. Anal. 1997. V. 34. № 5. P. 1911–1925.

34. *Druskin V.L., Greenbaum A., Knizhnerman L.A.* Using nonorthogonal Lanczos vectors in the computation of matrix functions // SIAM J. Sci. Comput. 1998. V. 19. № 1. P. 38–54.

35. *Hundsdorfer W., Verwer J.G.* Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations. Springer Verlag, 2003.

36. *Barrett R., Berry M., Chan T.F. et al.* Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods / Philadelphia, PA: SIAM, 1994. Available at www.netlib.org/templates/.

37. *Saad Y., Schultz M.H.* GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems // SIAM J. Sci. Stat. Comput. 1986. V. 7. № 3. P. 856–869.

38. *Saad Y.* Iterative Methods for Sparse Linear Systems. Book out of print, 2000. www-users.cs.umn.edu/~saad/books.html.

39. *Sidje R.B.* Expokit. A software package for computing matrix exponentials // ACM Trans. Math. Softw. 1998. V. 24. № 1. P. 130–156. www.maths.uq.edu.au/expokit/.

40. *Krukier L.A.* Implicit difference schemes and an iterative method for solving them for a certain class of systems of quasi-linear equations // Sov. Math. 1979. V. 23. № 7. P. 43–55. Translation from Izv. Vyssh. Uchebn. Zaved., Mat. 1979. № 7(206). P. 41–52.

41. *Taflove A., Hagness S.C.* Computational electrodynamics: the finite-difference time-domain method. 3d ed. Boston, MA: Artech House Inc., 2005.

42. *Kole J.S., Figge M.T., De Raedt H.* Unconditionally stable algorithms to solve the timedependent Maxwell equations // Phys. Rev. E. 2001. V. 64. P. 066705.

43. *Verwer J.G., Botchev M.A.* Unconditionally stable integration of Maxwell's equations // Linear Algebra and its Applications. 2009. V. 431. № 3–4. P. 300–317.

44. *Botchev M.A.* Krylov subspace exponential time domain solution of Maxwell's equations in photonic crystal modeling // J. Comput. Appl. Math. 2016. V. 293. P. 24–30. https://doi.org/10.1016/j.cam.2015.04.022

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 723–743

ОБЩИЕ ЧИСЛЕННЫЕ МЕТОДЫ

УДК 519.61

МЕТОДЫ ЭКСТРАПОЛЯЦИИ ШЭНКСА И ИХ ПРИЛОЖЕНИЯ[1)]

© 2021 г. К. Брезински[1],*, М. Редиво-Дзалья[2],**

[1] *Université de Lille, CNRS, UMR 8524 – Laboratoire Paul Painlevé, F-59000 Lille, France*

[2] *Università degli Studi di Padova, Dipartimento di Matematica "Tullio Levi-Civita", Via Trieste 63, 35121-Padova, Italy*

**e-mail: Claude.Brezinski@univ-lille.fr*

***e-mail: Michela.RedivoZaglia@unipd.it*

Поступила в редакцию 24.11.2020 г.
Переработанный вариант 24.11.2020 г.
Принята к публикации 14.01.2021 г.

Когда последовательность или серия скаляров, векторов, матриц, тензоров медленно сходится к своему пределу, она может быть преобразована путем преобразования последовательности в новую последовательность или набор новых последовательностей, которые при некоторых предположениях сходятся быстрее к тому же пределу. Такое преобразование можно применять также к расходящимся последовательностям или рядам, обеспечивая тем самым их аналитическое продолжение. Преобразование Шэнкса — хорошо известное преобразование последовательностей для ускорения сходимости в случае скаляров. В этом обзоре мы объясняем его разработку, различные расширения и реализацию. Несколько приложений иллюстрируют его эффективность. Библ. 51. Фиг. 11. Табл. 2.

Ключевые слова: методы ускорения, преобразования последовательностей, преобразование Шэнкса, аппроксимация Паде, тензор.

DOI: 10.31857/S0044466921050069

1. ВВЕДЕНИЕ

В настоящее время экстраполяционные методы подробно исследованы [1]–[7]. Они направлены на ускорение сходимости последовательностей или различных итерационных процессов. Однако области применения методов экстраполяции не ограничены только данной целью. Методы экстраполяции можно использовать в различных аппроксимационных задачах, при решении систем линейных и нелинейных уравнений, при вычислении матричных функций, и это лишь немногие из их потенциальных приложений. Однако эти методы не очень широко известны в сообществе прикладных математиков. В данной обзорной работе мы представляем один из методов экстраполяции, а именно – преобразование Шэнкса [8], так как согласно [9]: "Преобразование Шэнкса, пожалуй, является лучшим универсальным методом ускорения сходимости последовательностей". Мы предложим подробное описание данного метода, его различных расширений и продемонстрируем, что он может быть очень полезен во многих ситуациях.

Когда последовательность чисел, векторов, матриц или тензоров (S_n) медленно сходится к своему пределу S, при стремящемся к бесконечности n, и когда человек не имеет доступа к процессу, генерирующему значения исходной последовательности, этот процесс может быть преобразован путем преобразования последовательности T в новую последовательность (T_n) или набор новых последовательностей $\{(T_k^{(n)})\}$, которые, согласно некоторым предположениям, будут сходиться быстрее к тому же пределу, т.е. $\lim\limits_{n\to\infty} \|T_n - S\| / \|S_n - S\| = 0$.

Существует много таких преобразований последовательности, но общая идея, лежащая в их основе, – это интерполяция с последующей экстраполяцией. Пусть $\mathcal{H}(T)$ – множество последовательностей, члены которых зависят от $k+1$ произвольных неизвестных параметров. Предпо-

[1)]Работа К. Брезински выполнена при финансовой поддержке Labex CEMPI (ANR-11-LABX-0007-01). Работа М. Редиво-Дзальи выполнена при частичной финансовой поддержке проекта университета Падуи, Project 2019 no. DOR1903575/19. Микела Редиво-Дзалья – член группы INdAM Research group GNCS.

ложим, что для каждой последовательности $(t_n) \in \mathcal{K}(T)$, ее предел t может быть вычислен с использованием $k+1$ ее членов. Множество $\mathcal{K}(T)$ назовем *ядром* преобразования T.

Пусть теперь (S_n) – последовательность, подлежащая преобразованию. Предполагается, что она не принадлежит ядру преобразования. Зафиксируем целое число n. Проинтерполируем члены $S_n, \ldots, S_{n+k}$ последовательностью $(t_n) \in \mathcal{H}(T)$ такой, что $S_{n+i} = t_{n+i}$ для $i = 0, \ldots, k$. По построению данные условия для интерполянта позволяют вычислить предел t последовательности (t_n). Такие шаги порождают экстраполяционный процесс, описанный, например, в книге Марчука и Шайдурова, см. [3]. Заметим, что число t – предел интерполирующей последовательности $(t_n) \in \mathcal{K}(T)$, но оно не обязано быть пределом исходной последовательности (S_n), если она не принадлежит множеству $\mathcal{H}(T)$. Таким образом, с изменением n будет изменяться и предел t интерполяционной последовательности ядра, и мы обозначим его через T_n. В результате исходная последовательность (S_n) через ядро преобразования T порождает новую последовательность (T_n).

2. МЕТОД ЭКСТРАПОЛЯЦИИ ШЭНКСА

Среди преобразований для ускорения сходимости скалярных последовательностей преобразование Даниэля Шэнкса, пожалуй, является одним из лучших. Впервые оно было предложено в 1949 г. (см. [10]), но опубликовано лишь в 1955 г. (см. [8]). Его ядро – это множество последовательностей, удовлетворяющих для всех n однородному линейному разностному уравнению порядка k:

$$a_0(t_n - t) + a_1(t_{n+1} - t) + \ldots + a_k(t_{n+k} - t) = 0 \tag{1}$$

при условии $a_0 \times a_k \neq 0$. Без потери общности можно предположить, что выполнено условие нормировки $a_0 + a_1 + \ldots + a_k = 1$, и, таким образом, мы имеем

$$t = a_0 t_n + \cdots + a_k t_{n+k}, \quad n = 0, 1, \ldots . \tag{2}$$

2.1. Преобразование Шэнкса

Запись уравнения (2) для индексов $n, \ldots, n+k$ позволяет определить коэффициенты a_i как решение системы линейных уравнений, а затем вычислить t при помощи (2). Конечно, если последовательность (S_n) не принадлежит ядру, все еще возможно записать и решить систему уравнений, определяющую коэффициенты a_i и далее из уравнений (2) определить так называемое преобразование Шэнкса, обычно обозначаемое $e_k(S_n)$, так как результат будет зависеть и от n, и от k. В результате получим

$$e_k(S_n) = \begin{vmatrix} S_n & S_{n+1} & \cdots & S_{n+k} \\ \Delta S_n & \Delta S_{n+1} & \cdots & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \Delta S_{n+k} & \cdots & \Delta S_{n+2k-1} \end{vmatrix} \Bigg/ \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta S_n & \Delta S_{n+1} & \cdots & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta S_{n+k-1} & \Delta S_{n+k} & \cdots & \Delta S_{n+2k-1} \end{vmatrix}, \tag{3}$$

где Δ – разностный оператор дифференцирования “справа”, определяемый $\Delta S_i = S_{i+1} - S_i$ для всех i.

Решая разностные уравнения (1) через запись характеристического многочлена и поиск всех его корней, мы получаем следующий результат [11] (см. также [12] с более детальными комментариями).

Теорема 1. *Необходимым и достаточным условием для $e_k(S_n) = S$ при всех n является условие, что последовательность (S_n) удовлетворяет*

$$S_n - S = \sum_{i=1}^{p} A_i(n) r_i^n + \sum_{i=p+1}^{q} [B_i(n)\cos(b_i n) + C_i(n)\sin(b_i n)] e^{\omega_i n} + \sum_{i=0}^{m} c_i \delta_{in},$$

где $r_i \neq 0$ *для* $i = 1, \ldots, p$, A_i, B_i *и* C_i *– многочлены порядка* n *такие, что если* d_i *равняется степени* A_i *плюс* 1 *для всех* $i = 1, \ldots, p$ *и максимуму из степеней* B_i *и* C_i *плюс* 1 *для* $i = p = 1, \ldots, q$, *то*

$$m + \sum_{i=1}^{p} d_i + 2\sum_{i=p+1}^{q} d_i = k - 1,$$

с обязательным условием, что $m = -1$, *если нет члена* δ_{in} (*символ Кронекера*).

Коэффициенты полиномов A_i, B_i и C_i определяются начальными условиями в уравнении (1). Из этого результата видно, что многие итерационные методы, используемые в численном анализе и прикладной математике, порождают последовательности такого вида или близкие к нему, что объясняет важность метода экстраполяции Шэнкса.

2.2. ε*-Алгоритм*

Несколько преобразований последовательности могут быть выражены как отношение двух определителей. Таким образом, для нахождения преобразованных членов исходной последовательности должны быть получены алгоритмы, избегающие численного вычисления этих определителей. Рекурсивная процедура, ε-алгоритм, для реализации преобразования Шэнкса была предложена в 1956 г. Питером Винном [13]. Организация этой процедуры очень проста:

$$\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + (\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)})^{-1}, \quad k, n = 0, 1, \ldots, \tag{4}$$

при $\varepsilon_{-1}^{(n)} = 0$ и $\varepsilon_0^{(n)} = S_n$ для $n = 0, 1, \ldots$, и это выполнено для всех k и n,

$$\varepsilon_{2k}^{(n)} = e_k(S_n) \quad \text{и} \quad \varepsilon_{2k+1}^{(n)} = 1/e_k(\Delta S_n).$$

Таким образом, значения $\varepsilon_{2k+1}^{(n)}$ – это промежуточные результаты и не представляют интереса для наших целей.

Значения $\varepsilon_k^{(n)}$ обычно изображаются в виде двумерного ε-массива, где нижний индекс k сохраняется вдоль столбцов, а верхний индекс n сохраняется вдоль нисходящей диагонали:

$$\begin{array}{ccccccc}
\varepsilon_{-1}^{(0)} = 0 & & & & & & \\
 & \varepsilon_0^{(0)} = S_0 & & & & & \\
\varepsilon_{-1}^{(1)}=0 & & \varepsilon_1^{(0)} & & & & \\
 & \varepsilon_0^{(1)}=S_1 & & \varepsilon_2^{(0)} & & & \\
\varepsilon_{-1}^{(2)}=0 & & \varepsilon_1^{(1)} & & \varepsilon_3^{(0)} & & \\
 & \varepsilon_0^{(2)}=S_2 & & \varepsilon_2^{(1)} & & \ddots & \\
\varepsilon_{-1}^{(3)} = 0 & & \varepsilon_1^{(2)} & & \varepsilon_3^{(1)} & & \\
\vdots & \varepsilon_0^{(3)} = S_3 & & \varepsilon_2^{(2)} & & \ddots & \\
\vdots & \vdots & \varepsilon_1^{(3)} & & \varepsilon_3^{(2)} & & \\
\vdots & \vdots & \vdots & \varepsilon_2^{(3)} & & \ddots & \\
\vdots & \vdots & \vdots & \vdots & \varepsilon_3^{(3)} & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots &
\end{array}$$

В процедуре (4) ε-алгоритма обрабатываются значения, расположенные в четырех вершинах следующего ромба:

$$\begin{array}{ccc}
 & \varepsilon_k^{(n)} & \\
\nearrow & & \searrow \\
\varepsilon_{k-1}^{(n+1)} & & \varepsilon_{k+1}^{(n)}. \\
\searrow & & \nearrow \\
 & \varepsilon_k^{(n+1)} &
\end{array}$$

В случае, когда в знаменателе (4) два значения оказались равными, вычислительная процедура должна быть остановлена (данное явление принято называть *обвалом*). Но если значения в знаменателе почти равны, возникает неустойчивость при вычислении очередного члена (называемая *почти–обвалом*). Для разрешения этих недостатков алгоритма Винн, см. [14], также предложил конкретные правила для скалярного ε-алгоритма, которые позволяют повысить устойчивость вычислений.

2.3. Аппроксимации Паде

ε-Алгоритм, описанный в п. 2.2, связан с известными аппроксимациями Паде, используемыми, в частности, для суммирования сходящихся и расходящихся рядов [6], [15].

Пусть $f(z) = \sum_{i=0}^{\infty} c_i z^i$ – формальный степенной ряд. Аппроксимация Паде для f – это рациональная функция, разложение в степенной ряд которой по возрастающим степеням z является близким к ряду f настолько, насколько это возможно, т.е. с точностью до суммы $p+q$ включительно, где p – степень его числителя, а q – степень q его знаменателя.

Обозначим такую аппроксимацию через

$$[p/q]_f(z) = (a_0 + a_1 z + \ldots + a_p z^p) \Big/ (b_0 + b_1 z + \ldots + b_q z^q),$$

где коэффициенты можно получить из следующих выражений:

$$\begin{aligned}
&a_0 = c_0 b_0,\\
&a_1 = c_1 b_0 + c_0 b_1,\\
&\vdots \quad \vdots\\
&a_p = c_p b_0 + c_{p-1} b_1 + \ldots + c_{p-q} b_q,\\
&0 = c_{p+1} b_0 + c_p b_1 + \ldots + c_{p-q+1} b_q,\\
&\vdots \quad \vdots\\
&0 = c_{p+q} b_0 + c_{p+q-1} b_1 + \ldots + c_p b_q
\end{aligned}$$

при условии, что $c_i = 0$ для $i < 0$.

Линейная система, порожденная последними q уравнениями, содержит $q+1$ неизвестную $b_0, \ldots, b_q$ и, следовательно, существует ее нетривиальное решение. Положив $b_0 = 1$, мы можем получить значения $b_1, \ldots, b_q$, решая линейную систему, предполагая ее невырожденность. Зная значения b_i, из первых $p+1$ уравнений мы можем вычислить a_i.

Выполнено

$$[n+k/k]_f(z) = \begin{vmatrix} z^k S_n(z) & z^{k-1} S_{n+1}(z) & \cdots & S_{n+k}(z) \\ c_{n+1} & c_{n+2} & \cdots & c_{n+k+1} \\ \vdots & \vdots & & \vdots \\ c_{n+k} & c_{n+k+1} & \cdots & c_{n+2k} \end{vmatrix} \Bigg/ \begin{vmatrix} z^k & z^{k-1} & \cdots & 1 \\ c_{n+1} & c_{n+2} & \cdots & c_{n+k+1} \\ \vdots & \vdots & & \vdots \\ c_{n+k} & c_{n+k+1} & \cdots & c_{n+2k} \end{vmatrix}$$

при $S_n(z) = \sum_{i=0}^{n} c_i z^i$.

Применяя ε-алгоритм к частичным суммам f при $\varepsilon_0^{(n)} = S_n(z)$, можно заметить, сопоставив предыдущую формулу с (3), что

$$e_k(S_n) = [n+k/k]_f(z).$$

Аппроксиманты со степенью знаменателя большей, чем степень числителя, могут быть получены через применение ε-алгоритма к частичным суммам обратного ряда g функции f, формально определяемого через $f(z)g(z) = 1$ и через свойство $[p/q]_f(z)[q/p]_g(z) = 1$. Ряд g существует только тогда, когда $c_0 \neq 0$.

3. НЕСКАЛЯРНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ

Преобразование Шэнкса и ε-алгоритм можно обобщить на случай нескалярных последовательностей, как мы увидим ниже.

3.1. Случай векторных и матричных последовательностей

В работе [16] Винн предложил расширение своего ε-алгоритма для случая векторных и матричных последовательностей. В данном случае правила (4) остаются прежними, а степень -1 соответствует получению обратной матрицы. Ядро преобразования снова можно записать в форме (1). Позже А. Салам (см. [17]) предложил доказательство, что данный результат все еще верен, если коэффициенты a_i являются квадратными матрицами, и произведение возникает либо слева, либо справа. Алгоритм может быть обобщен и для случая прямоугольных матриц через замену операции обращения матриц вычислением псевдообратных, но результат, касающийся ядра, не остается верным. Подробнее случай ε-алгоритма Винна для матричных последовательностей рассмотрен в [18].

В случае векторных последовательностей Винн определил обращение вектора u через $u^{-1} = u/(u,u)$. Векторы $\varepsilon_{2k}^{(n)}$ тоже могут быть выражены, как и в скалярном случае, через отношение определителей, но размерности $2k+1$ вместо исходной $k+1$, как было доказано П.Р. Грейвс-Моррисом и К. Дженкинсом [9], [19]. Применив алгебру Клиффорда, Дж.Б. МакЛеод доказал, что ядро векторного ε-алгоритма также содержит векторы формы (1) [20], где коэффициенты a_i являются вещественными числами. Позже А. Салам доказал, что ядром векторного ε-алгоритма является непосредственно (1), но с коэффициентами a_i, принадлежащими алгебре Клиффорда, ассоциированной с $\mathbb{R}^p$ [21], [22], где p – размерность векторов последовательности S_n. Аналогичная (3) формула по-прежнему имеет место, но определители необходимо заменить на конструкторы (designants), которые обобщают их в некоммутативной алгебре [17], [23].

3.2. Топологическое преобразование Шэнкса

Столкнувшись с трудностями, возникающими в алгебраических теориях для векторных и матричных ε-алгоритмов, К. Брежинский решил заново заняться этой задачей с начальных этапов [24]. Пусть (S_n) – последовательность элементов топологического векторного пространства E над $\mathbb{C}$, или просто над $\mathbb{R}$ (топология необходима, чтобы уметь рассматривать вопросы сходимости). Задачей было построить преобразование последовательности $(S_n) \mapsto \{(e_k(S_n)\}$ такое, что для всех последовательностей (S_n), удовлетворяющих (1) с $t = S$, при фиксированном k и с коэффициентами $a_i \in \mathbb{C}$, $e_k(S_n) = a_0S_n + a_1S_{n+1} + \ldots + a_kS_{n+k} = S \in E$ для всех n. Записав (1) для индексов n и $n+1$ и вычислив разность, получим

$$a_0\Delta S_n + a_1\Delta S_{n+1} + \ldots + a_k\Delta S_{n+k} = 0 \in E.$$

Следующей задачей стало вычислить коэффициенты a_i. Для этого полученное выражение было преобразовано в выражение над $\mathbb{C}$. Пусть y является элементом алгебраического сопряженного пространства E^* для E, являющегося пространством линейных функционалов над E, и обозначим через $\langle\cdot,\cdot\rangle$ соотношение двойственности между E^* и E. Применение этого соотношения к выражениям, описанным выше, позволяет получить

$$a_0\langle y, \Delta S_n\rangle + a_1\langle y, \Delta S_{n+1}\rangle + \ldots + a_k\langle y, \Delta S_{n+k}\rangle = 0 \in \mathbb{C}. \tag{5}$$

Записав это выражение для индексов от n до $n+k-1$, добавив условие нормировки и решив соответствующую линейную систему для a_i, мы получим (первое) *топологическое преобразование Шэнкса*, определяемое через:

$$\hat{e}_k(S_n) = \frac{\begin{vmatrix} S_n & S_{n+1} & \cdots & S_{n+k} \\ \langle y, \Delta S_n\rangle & \langle y, \Delta S_{n+1}\rangle & \cdots & \langle y, \Delta S_{n+k}\rangle \\ \vdots & \vdots & & \vdots \\ \langle y, \Delta S_{n+k-1}\rangle & \langle y, \Delta S_{n+k}\rangle & \cdots & \langle y, \Delta S_{n+2k-1}\rangle \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \langle y, \Delta S_n\rangle & \langle y, \Delta S_{n+1}\rangle & \cdots & \langle y, \Delta S_{n+k}\rangle \\ \vdots & \vdots & & \vdots \\ \langle y, \Delta S_{n+k-1}\rangle & \langle y, \Delta S_{n+k}\rangle & \cdots & \langle y, \Delta S_{n+2k-1}\rangle \end{vmatrix}}. \tag{6}$$

Второе топологическое преобразование Шэнкса, обозначаемое через $\tilde{e}_k(S_n)$, можно определить, сделав замену первой строки числителя на $S_{n+k}, \ldots, S_{n+2k}$. Если $E = \mathbb{C}$, два данных преобразования редуцируются до скалярного преобразования Шэнкса (3).

3.3. Топологические ε*-алгоритмы*

Следующей задачей стало получение рекурсивного алгоритма для реализации полученного преобразования. Этот алгоритм также должен был быть сокращен до (4) в скалярном случае. Ключевым моментом стало определение обращения, которое ограничивается обычным преобразованием для скаляров и векторным в форме, использованной Винном в его векторном ε-алгоритме. Решением оказалось определение обращения пары $(y,u) \in E^* \times E$ через $y^{-1} = u/\langle y,u\rangle \in E$ и $u^{-1} = y/\langle y,u\rangle \in E^*$. Такое обращение удовлетворяет всем свойствам, которым должна удовлетворять данная операция: обратный элемент может быть вычислен, а отношение двойственности элемента со своим обратным равно единице. Это определение обращения привело к тому, что правило этого, так называемого, *топологического* ε-алгоритма пришлось разделить на два правила: одно для членов с еще четным нижним индексом, принадлежащих E, и второе для элементов с нечетным нижним индексом, которые находятся в E^* и являются промежуточными результатами, как в скалярном случае. Эти наблюдения приводят к (первому) *топологическому* ε-алгоритму (TEA1). Второй алгоритм (TEA2) позволяет реализовать рекурсивно второе преобразование. Доказано, что $\hat{\varepsilon}_{2k}^{(n)} = \hat{\varepsilon}_k(S_n)$, как определено уравнением (6), а для второго преобразования и алгоритма, что $\tilde{\varepsilon}_{2k}^{(n)} = \tilde{e}_k(S_n)$ [24].

Правила алгоритмов TEA были довольно сложными (два разных правила, они требуют хранения элементов E и E^*, отношение двойственности было рекурсивно использовано в правилах). Таким образом, было трудно реализовать их для линейных функционалов общего вида, и в прошлом эти алгоритмы использовались только в простом случае, когда E совпадает с $\mathbb{R}^p$ (или $\mathbb{C}^p$), y был вектором, и отношение двойственности сводится к внутреннему произведению между действительными или комплексными векторами. В 2014 г. авторы этой работы осознали, спустя 40 лет после публикации [24], что соотношение, записанное ранее, позволяет существенно упростить правила двух топологических ε-алгоритмов [25], что привело к формулировке *упрощенных топологических* ε-алгоритмов (STEA). Каждый из двух этих упрощенных алгоритмов может быть записан в 4 эквивалентных формах. Приведем только правило второго (STEA2), которое еще более эффективно и включает работу только с величинами с четными нижними индексами:

$$\tilde{\varepsilon}_{2k+2}^{(n)} = \tilde{\varepsilon}_{2k}^{(n+1)} + \frac{\varepsilon_{2k+2}^{(n)} - \varepsilon_{2k}^{(n+1)}}{\varepsilon_{2k}^{(n+2)} - \varepsilon_{2k}^{(n+1)}}(\tilde{\varepsilon}_{2k}^{(n+2)} - \tilde{\varepsilon}_{2k}^{(n+1)})$$

при $\tilde{\varepsilon}_0^{(n)} = S_n$, и где значения $\varepsilon_k^{(n)}$ вычисляются с использованием скалярного ε-алгоритма Винна, примененного к последовательности $(\varepsilon_0^{(n)} = \langle y, S_n\rangle)$.

По сравнению с оригинальными топологическими ε-алгоритмами упрощенные используют только одно треугольное правило вместо двух более сложных, также они позволяют использовать только элементы E с нижними четными индексами, отсутствует необходимость вычислять или хранить элементы E^*, соотношение двойственности используется только на этапе инициализации скалярного ε-алгоритма, требования по количеству используемых ячеек памяти были существенно снижены (что важно в случае работы с векторными и матричными последовательностями), а устойчивость алгоритмов была повышена с использованием конкретных правил Винна для скалярного ε-алгоритма [14]. Больше деталей и информацию о программной реализации можно найти в работе [26].

3.4. Другие преобразования Шэнкса

Как мы видим теперь, существуют различные способы использования уравнения (5) для вычисления коэффициентов a_i в преобразовании. Всегда предполагается, что сумма коэффициентов a_i равна 1.

Вместо использования одного элемента $y \in E^*$ при меняющемся n мы можем зафиксировать n и выбрать k линейно-независимых функционалов $y_i \in E^*$. Эта процедура приводит к *методу минимальной полиномиальной экстраполяции* (MMPE) [24], который записывается через:

$$e_k(S_n) = \frac{\begin{vmatrix} S_n & S_{n+1} & \cdots & S_{n+k} \\ \langle y_1, \Delta S_n \rangle & \langle y_1, \Delta S_{n+1} \rangle & \cdots & \langle y_1, \Delta S_{n+k} \rangle \\ \vdots & \vdots & & \vdots \\ \langle y_K, \Delta S_n \rangle & \langle y_k, \Delta S_{n+1} \rangle & \cdots & \langle y_k, \Delta S_{n+k} \rangle \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \langle y_1, \Delta S_n \rangle & \langle y_1, \Delta S_{n+1} \rangle & \cdots & \langle y_1, \Delta S_{n+k} \rangle \\ \vdots & \vdots & & \vdots \\ \langle y_k, \Delta S_n \rangle & \langle y_k, \Delta S_{n+1} \rangle & \cdots & \langle y_k, \Delta S_{n+k} \rangle \end{vmatrix}}.$$

Для этого преобразования мы используем те же обозначения $e_k(S_n)$, что и для топологического преобразования Шэнкса, даже если оно отличается от него. Это преобразование может быть реализовано рекурсивно через $S\beta$-алгоритм согласно Л. Джейбилоу [27] (см. также [28]).

Метод полиномиальной экстраполяции (MPE) [29] и метод *экстраполяции с редуцированным рангом* (RRE) [30], [31] можно напрямую получить из выражения MMPE, подставив $y_i = \Delta S_{n+i-1}$ для первого и $y_i = \Delta^2 S_{n+i-1}$ для второго. Однако оба этих метода применимы лишь в случае векторных последовательностей.

Существует и несколько других обобщений, упомянем, например, [18], [32] и книги [1], [4], [5].

4. СТРАТЕГИИ И РЕАЛИЗАЦИИ

Обсудим два способа использования преобразований Шэнкса: методы *ускорения* и *рестарта*, непосредственно посвященные решению задач о неподвижной точке. Реализация обсуждаемых стратегий будет детально обсуждаться ниже.

4.1. Метод ускорения (AM)

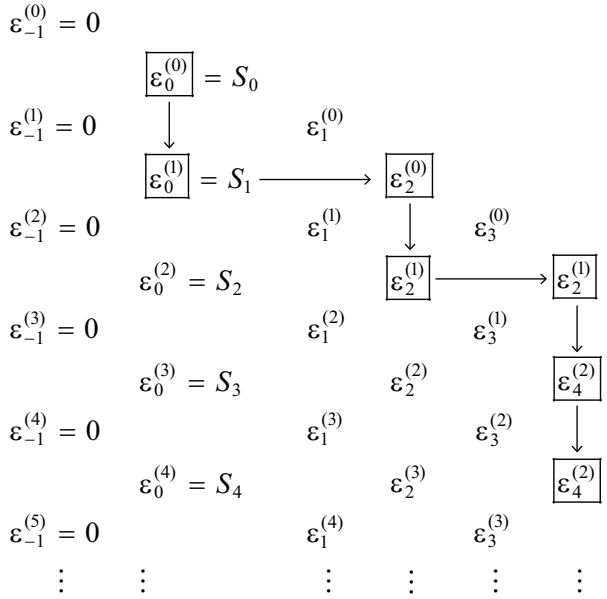
Пусть (S_n) – последовательность скаляров, векторов, матриц или тензоров, сходимость которой необходимо ускорить с помощью одного из ε-алгоритмов. Предположим, что для фиксированного значения k мы хотим вычислить последовательность $(\varepsilon_{2k}^{(n)})$. Члены последовательности (S_n) определяются один за другим в соответствующей программной реализации, и четные столбцы ε-массива вычислены настолько далеко, насколько это возможно. Начиная с S_0 и S_1, ни один член не может быть получен. Тогда, вводя S_2, можно вычислить $\varepsilon_2^{(0)}$. С использованием дополнительно еще S_3 мы можем получить $\varepsilon_2^{(1)}$. Далее с S_4, можно получить $\varepsilon_4^{(0)}$. Таким образом, нисходящая "лестница" вдоль главной диагонали ε-массива получается до тех пор, пока, введя член S_{2k}, мы не сможем вычислить $\varepsilon_{2k}^{(0)}$. Далее каждый новый член $S_{2k+1}, S_{2k+2}, \ldots$ позволяет вычислить следующий член в столбце $2k$, т.е. $\varepsilon_{2k}^{(1)}, \varepsilon_{2k}^{(2)}, \ldots$. Если вместо вычисления при фиксированном значении k последовательности $(\varepsilon_{2k}^{(n)})$ требуется вычислить элементы главной нисходящей диагонали или "лестницу" вдоль нее, достаточно дать большое значение k и прекратить вводить новые члены последовательности для преобразования. Данная стратегия называется *методом ускорения* (AM). Фигура 1 демонстрирует сопутствующий алгоритм, а фиг. 2 приводит пример данной стратегии.

Поясним далее, как построить таблицу, см. фиг. 2, по последовательности (S_n). Простейший способ построения ε-массива – это вычислить столбцы один за другим, начиная с первого столбца, используя далее для всех столбцов (кроме первого) значения из двух предыдущих. С точки зрения требований по количеству используемых ячеек памяти и количества действий данная процедура может быть слишком затратной в случае, если членами последовательности являются векторы или матрицы. Поэтому обычно при реализации алгоритмов используется

Алгоритм

Выбрать $2k$ и S_0
For $n = 1, 2, 3, \ldots,$
Вычислить S_n
Использовать метод и вычислить на каждом цикле новый экстраполированный член последовательности
$\varepsilon_0^{(0)} = S_0, \varepsilon_0^{(1)}, \varepsilon_2^{(0)}, \varepsilon_2^{(1)}, \ldots, \varepsilon_{2k}^{(0)}, \varepsilon_{2k}^{(1)}, \varepsilon_{2k}^{(2)}, \ldots$
end for n

Фиг. 1. Метод ускорения (AM).



Фиг. 2. Пример ε-массива при $2k = 4$.

прием *бегущего окна*, который известен благодаря Винну [14], использовавшему данную технику в своем скалярном ε-алгоритме.

Конечно, эта техника может использоваться и в других алгоритмах, в частности, в топологическом ε-алгоритме или в его упрощенных версиях. Идея состоит в том, чтобы вести работу от возрастающих диагоналей, т.е. мы вычисляем каждую новую возрастающую диагональ массива, используя предыдущую диагональ (или две предшествующие в случае первого топологического ε-алгоритма). Большее число деталей сообщено в работе [26] и в пользовательской инструкции соответствующего программного обеспечения [33].

4.2. Метод рестарта (RM)

В частном случае вычисления скалярной, векторной, матричной или тензорной неподвижной точки основная проблема состоит в том, чтобы найти x, удовлетворяющий уравнению $x = F(x)$. Очевидно, можно организовать итерационный процесс в форме

$$S_{n+1} = F(S_n), \quad n = 0, 1, \ldots,$$

и затем ускорить его сходимость с помощью AM. Однако можно использовать и другую стратегию стеффенсенского типа, известную под названием *метода рестарта* (RM). Начиная с заданного x_0, мы вводим $u_0 = x_0$ и проводим некоторое количество итераций Пикара, *базовых итераций*, $u_{i+1} = F(u_i)$ для $i = 0, \ldots, 2k-1$ для фиксированного значения k. Далее применение ε-алгоритма к этим $2k+1$ элементам генерирует $x_1 = \varepsilon_{2k}^{(0)}$. Базовые итерации после этого можно начать

Алгоритм

Выбрать $2k$ и x_0.

For $n = 0, 1, \ldots$ (внешние итерации)

Присвоить $u_0 = x_n$

Вычислить $u_{i+1} = F(u_i)$, для $i = 0, \ldots, 2k-1$ (внутренние итерации)

Применить метод экстраполяции для $u_0, \ldots, u_{2k}$ и вычислить $\varepsilon_{2k}^{(0)}$

Присвоить $x_{n+1} = \varepsilon_{2k}^{(0)}$

end for n

Фиг. 3. Метод рестарта (RM).

уже с $u_0 = x_1$ и продолжать далее, что порождает последовательность (x_n), которая при некоторых предположениях сходится к неподвижной точке x. Фигура 3 демонстрирует соответствующий алгоритм. В векторном случае $F : \mathbb{R}^p \mapsto \mathbb{R}^p$, Х. Ле Ферран [34] доказал, что при выборе $k = p$ использование любого из топологических ε-алгоритмов в RM генерирует последовательность, которая при некоторых предположениях сходится квадратично к неподвижной точке $x \in \mathbb{R}^p$ для отображения F. Данная процедура называется *обобщенным методом Стеффенсена* (GSM), так как при $p = 1$ процедура оказывается непосредственно методом Стеффенсена [35]. Для метода RRE подробный анализ представлен в работе Сиди [36].

4.3. Программные реализации

Несмотря на то что во многих публикациях в литературе утверждается, что преобразования последовательностей и соответствующие алгоритмы могут быть полезными методами для получения или улучшения численных решений широкого круга задач численного анализа и прикладной математики, потенциальных пользователей часто отговаривают использовать их из-за сложности их реализации простым и оптимизированным способом. По этой причине в 1991 г. в дополнение к [1] на языке FORTRAN 77 была опубликована библиотека (процедуры и демонстрационные программы). Недавно библиотека EPSfun [33] на языке MATLAB была внедрена в библиотеку с открытым программным кодом Netlib. Она содержит необходимые функции по реализации ε-алгоритмов Винна, оригинальных топологических ε-алгоритмов и их упрощенных версий. Напомним, что оригинальные и упрощенные топологические ε-алгоритмы требуют определения элемента $y \in E^*$. В оригинальных алгоритмах y непосредственно используется в одном из правил, в то время как в упрощенных данный элемент играет роль только на этапе инициализации скалярного ε-алгоритма, который применяется к последовательности $(\langle y, S_n \rangle)$. Демонстрационные программы, шаблоны проектов (как для методов AM, так и для RM), а также пользовательские инструкции являются частью данного программного обеспечения.

5. ПРИМЕРЫ ПРИЛОЖЕНИЙ

Все эти алгоритмы получили множество применений: в ускорении сходимости скалярных, векторных и матричных последовательностей (в частности, для рекурсивного решения систем линейных и нелинейных уравнений), при работе с осцилляциями Гиббса для рядов Фурье, при вычислении матричных функций, при поиске решений интегральных уравнений, в задачах, связанных с тензорами и многих других. В данном разделе мы постараемся дать обзор некоторых из таких задач и проиллюстрировать конкретные улучшения, достигаемые в результате использования методов экстраполяции.

5.1. Скалярный ε-алгоритм

Начнем с демонстрации двух случаев применения скалярного ε-алгоритма. В первом случае рассматривается задача об ускорении сходимости конкретной последовательности. Во втором случае алгоритм применяется к рядам Фурье.

Таблица 1. Паде аппроксимации для логарифма

k	$f_{2k}(1)$	$[k/k]_f(1)$	k	$f_{2k}(2)$	$[k/k]_f(2)$
1	0.830	0.7	1	0.260×10^{1}	1.14
2	0.783	0.6933	2	0.506×10^{1}	1.101
3	0.759	0.693152	3	0.126×10^{2}	1.0988
4	0.745	0.69314733	4	0.375×10^{2}	1.098625
5	0.736	0.6931471849	5	0.121×10^{3}	1.0986132
6	0.730	0.69314718068	6	0.410×10^{3}	1.09861235
7	0.725	0.693147180563	7	0.142×10^{4}	1.098612293
8	0.721	0.69314718056000	8	0.504×10^{4}	1.0986122890
9	0.718	0.6931471805599485	9	0.181×10^{5}	1.098612288692
10	0.716	0.6931471805599454	10	0.655×10^{5}	1.0986122886698

5.1.1. Скалярные последовательности. Применим скалярный ε-алгоритм к частичным суммам f_{2k} ряда

$$f(z) = \ln(1+z) = z - z^2/2 + z^3/3 - z^4/4 + \ldots,$$

который сходится для $|z| \le 1$, кроме $z = -1$.

Для $z = 1$ имеем $\ln 2 = 0.6931471805599453\ldots$. Для $z = 2$ ряд расходится, и мы имеем $\ln 3 = 1.098612288668110\ldots$. Аппроксимации Паде приводят к результатам, содержащимся в табл. 1 (напомним, что $\varepsilon_{2k}^{(0)} = [k/k]_f(z)$).

Мы можем видеть, что ε-алгоритм позволяет получить суммы для расходящихся последовательностей (рядов, в данном конкретном случае). Это следует из факта, что значения $\varepsilon_{2k}^{(n)}$ связаны с аппроксимациями Паде и что эти аппроксимации позволяют получить аналитическое продолжение для некоторых рядов вне области сходимости, что и демонстрирует приведенный пример.

5.1.2. Эффект Гиббса. Далее мы приведем численные примеры, показывающие, что ε-алгоритм позволяет локализовать разрывы в рядах Фурье, ускорить их сходимость и уменьшить выбросы, возникающие из-за эффекта Гиббса.

Рассмотрим ряд Фурье общего вида (без ограничений общности, предположим, что ряд является периодическим на отрезке $[0, 2\pi]$ и обладает нулевым постоянным членом):

$$S(t) = \sum_{k=1}^{\infty} a_k \cos kt + \sum_{k=1}^{\infty} b_k \sin kt.$$

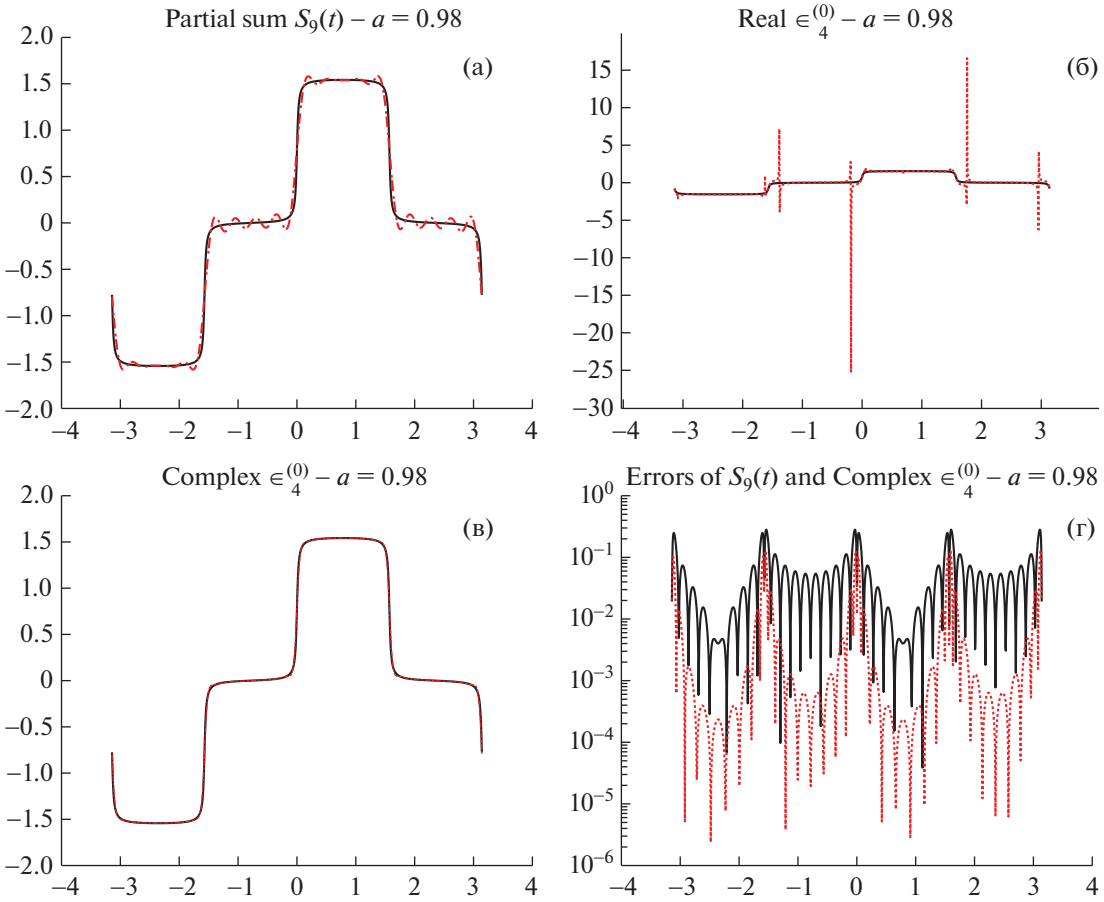
Складывая этот ряд с сопряженным:

$$\widetilde{S}(t) = \sum_{k=1}^{\infty} a_k \sin kt - \sum_{k=1}^{\infty} b_k \cos kt$$

как с мнимой частью, мы получаем комплексный ряд Фурье:

$$F(t) = S(t) + i\widetilde{S}(t) = \sum_{k=1}^{\infty} (a_k - ib_k)e^{ikt}.$$

Применим теперь ε-алгоритм к частичным суммам $F(t)$ и далее возьмем вещественную часть значений $\varepsilon_{2k}^{(n)}$. Данная процедура известна под названием *комплексного* ε-алгоритма, а получаемые значения обозначаются ${}_C\varepsilon_{2k}^{(n)}$. В случае, если скалярный ε-алгоритм применяется к частичным суммам $S(t)$ (как в п. 2.3), мы считаем, что речь идет о *вещественном* ε-алгоритме и обозначаем получаемые значения ${}_R\varepsilon_{2k}^{(n)}$.



Фиг. 4. Эффект Гиббса и скалярный ε-алгоритм.

В качестве примера рассмотрим ряд Фурье на отрезке $[0, \pi]$:

$$S(t) = \frac{1}{2}\left(\arctan\frac{2a\cos t}{1-a^2} + \arctan\frac{2a\sin t}{1-a^2}\right) =$$
$$= \sum_{k=1}^{\infty}(-1)^{k+1}\frac{a^{2k-1}}{2k-1}\cos(2k-1)t + \sum_{k=1}^{\infty}\frac{a^{2k-1}}{2k-1}\sin(2k-1)t.$$

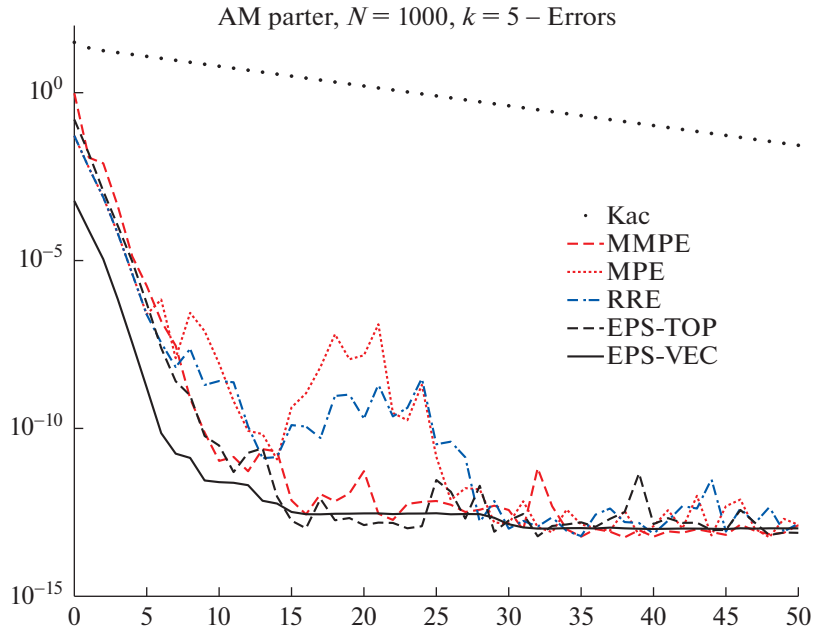
На фиг. 4а–в сплошная линия соответствует точному результату, пунктирная представляет результаты использования ε-алгоритма. На фиг. 4г мы демонстрируем уровень ошибки $S_9(t)$ (что соответствует частичной сумме из первых 9 членов $S(t)$) относительно точных значений при значении $a = 0.98$. График ${}_R\varepsilon_4^{(0)}$ демонстрирует, что алгоритм позволяет выявить почти все разрывы.

График значений ошибки демонстрирует улучшения, полученные в результате использования комплексного ε-алгоритма. Эта идея была предложена Винном [37], затем приведена в работе [38], где ε-алгоритм использовался для локализации разрывов, и окончательно оформлена в [39].

5.2. Работа с системами уравнений

Применим теперь ранее описанные алгоритмы к задачам о решении систем линейных и нелинейных уравнений.

5.2.1. Метод Качмарца. Метод Качмарца – итерационный метод решения систем линейных уравнений [40]. Хотя история этого алгоритма началась в 1937 г., в последнее время к нему возрождается интерес из-за его хорошей структуры для параллельной реализации. Другим достоин-



Фиг. 5. Ускорение сходимости метода Качмарца для `parter` матрицы, $N = 1000$, $k = 5$.

ством этого подхода является его сходимость для всех случаев, как было доказано [41]. Тем не менее сходимость обычно оказывается медленной, и несколько процедур для ее ускорения можно найти в литературе.

Одна итерация *циклического* метода Качмарца для решения линейной системы $Ax = b$ размерности N состоит в следующих действиях:

$$p_0 = x_n,$$
$$p_i = p_{i-1} + \frac{(b - Ap_{i-1}, e_i)}{(A^{\mathrm{T}}e_i, A^{\mathrm{T}}e_i)} A^{\mathrm{T}}e_i, \quad i = 1, \ldots, N,$$
$$x_{n+1} = p_N.$$

Обозначим через $a_i = A^{\mathrm{T}}e_i$ вектор-столбец, сформированный из i-го столбца A, а через b_i – компоненту с номером i вектора правой части b. Тогда вычисление каждого из векторов p_i в шаге итерации не требует вычисления умножения матрицы на вектор, что приводит к:

$$p_i = p_{i-1} + \frac{b_i - (p_{i-1}, a_i)}{\|a_i\|^2} a_i.$$

Данное замечание позволяет легко получить параллельную реализацию алгоритма.

Обозначим далее $P_i = I - A\alpha_i e_i^{\mathrm{T}}$ и $Q_i = A^{-1}P_iA$ при $\alpha_i = A^{\mathrm{T}}e_i \Big/ \left\|A^{\mathrm{T}}e_i\right\|^2$. Выполнено $x_{n+1} - x = Q(x_n - x)$ при $Q = Q_N \cdots Q_1$, что есть $x_n - x = Q^n(x_0 - x)$. Данное выражение демонстрирует, что итерационная последовательность (x_n) принадлежит к ядру топологического преобразования Шэнкса. Так, в теории его применение может привести к получению точного решения системы. Однако, так как N обычно (очень) велико, преобразование нельзя использовать на практике, хотя его можно использовать в целях ускорения сходимости, как это делается в работе [42].

Мы рассматриваем `parter` матрицу A, $N = 1000$, $\varkappa(A) \simeq 4.2306$, – тёплицеву матрицу с сингулярными числами, близкими к π. Фигура 5 демонстрирует сравнение метода Качмарца с MMPE, MPE, RRE, топологическим ε-алгоритмом и с векторным ε-алгоритмом при $k = 5$.

Мы видим, что все методы достигают хорошей точности с преимуществом векторного ε-алгоритма. Более того, его сходимость оказывается более гладкой. В приведенном примере стар-

шее собственное значение матрицы A равно 0.8732178, а следующее за ним -0.3170877. Таким образом, согласно теоретическим результатам, общим для всех этих методов, хорошее ускорение может наблюдаться даже при $k = 1$.

5.2.2. Интегральные уравнения. Рассмотрим следующее нелинейное *интегральное уравнение Фредгольма* II *рода* с ядром K:

$$u(t) = \int_a^b K(t, x, u(x))dx + f(t), \quad t \in [a, b].$$

Будем считать, что выполнены все обычные условия, гарантирующие существование единственного решения.

Стандартным методом решения этого уравнения является приближение интегрального оператора с помощью квадратурных формул:

$$\int_a^b K(t, x, u(x))dx \simeq \sum_{j=0}^{p} w_j^{(p)} K(t, x_j^{(p)}, u(x_j^{(p)})),$$

где $x_0^{(p)}, \ldots, x_p^{(p)}$ – это $p+1$ точка на отрезке $[a, b]$, а верхний индекс p соответствует зависимости от номера выбранной точки. Напомним, что веса $w_j^{(p)}$ строго положительны, а их сумма равна $b - a$. Таким образом, уравнение аппроксимируется через

$$u_p(t) = \sum_{j=0}^{p} w_j^{(p)} K(t, x_j^{(p)}, u_p(x_j^{(p)})) + f(t).$$

Далее мы приближаем решение u_p методом коллокаций для точек $t_i^{(p)} = x_i^{(p)}$ при $i = 0, \ldots, p$. При фиксированном значении p обозначим для удобства $t_i = x_i = t_i^{(p)} = x_i^{(p)}$, $f_i = f(t_i)$, $w_i^{(p)} = w_i$ и определим далее аппроксимации u_i для $u_p(t_i)$, $i = 0, \ldots, p$, как решение системы из $p+1$ нелинейных уравнений:

$$u_i = \sum_{j=0}^{p} w_j K(t_i, t_j, u_j) + f_i, \quad i = 0, \ldots, p.$$

Для решения этой системы для начала мы используем итерационный метод Пикара:

$$u_i^{(n+1)} = \sum_{j=0}^{p} w_j K(t_i, t_j, u_j^{(n)}) + f_i, \quad i = 0, \ldots, p,$$

или, в более общем случае схему релаксации:

$$u_i^{(n+1)} = u_i^{(n)} - \alpha\left\{u_i^{(n)} - \sum_{j=0}^{p} w_j K(t_i, t_j, u_j^{(n)}) - f_i\right\}, \quad i = 0, \ldots, p,$$

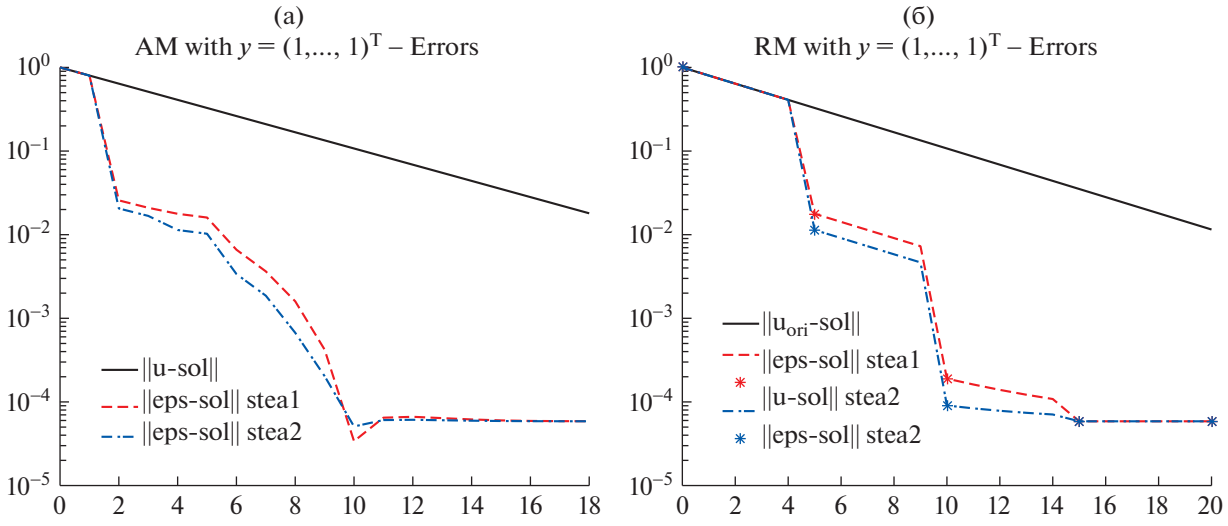
где α – параметр, который необходимо найти, а $u_i^{(0)}$, $i = 0, \ldots, p$ – начальное приближение решения в точках t_i.

Затем эти итерации вводятся в один из методов, полученных из преобразования Шэнкса, таких как MPE, MMPE, RRE, векторный или упрощенный топологический ε-алгоритмы. Полный математический анализ этой методологии и алгоритм описаны в работе [43]. Программы на языке MATLAB с несколькими примерами находятся в свободном доступе на сайте Matlab File Exchange.

В качестве примера мы рассмотрим интегральное уравнение

$$u(t) = t^2 \int_0^1 \frac{x^2}{1 + u^2(x)} dx + (1/2 - \ln 2)t^2 + \sqrt{t},$$

чьим решением является $u(x) = \sqrt{x}$. Из преобразований Шэнкса здесь используются первый и второй упрощенные топологические ε-алгоритмы, каждый с использованием AM и RM стратегий.



Фиг. 6. Стратегии AM (а) и RM (б) при $y = (1, \ldots, 1)^T$.

Результаты использования стратегии AM при параметрах $\alpha = 0.2$, $p = 31$, $y = (1, \ldots, 1)^T$ и $2k = 10$ представлены на фиг. 6а, результаты при $2k = 4$ для стратегии RM представлены на фиг. 6б.

Нашей целью было не конкурировать со сложными методами, которые можно найти в литературе, а показать, что простая процедура, сопровождаемая методом ускорения, может быть весьма эффективной.

5.2.3. Нелинейные алгебраические уравнения. Для ускорения сходимости итерационного метода $x_{n+1} = x_n + \alpha f(x_n)$ при решении системы линейных или нелинейных уравнений $f(x) = 0$ мы можем воспользоваться *методом ускорения* (AM) или *методом рестарта* (RM).

Рассмотрим далее нелинейную систему:

$$x_1 = x_1 x_2^3/2 - 1/2 + \sin x_3,$$
$$x_2 = (\exp(1 + x_1 x_2) + 1)/2,$$
$$x_3 = 1 - \cos x_3 + x_1^4 - x_2$$

с решением $(-1, 1, 0)^T$.

Начиная с точки $x_0 = 0 \in \mathbb{R}^3$, мы наблюдаем результаты, представленные на фиг. 7а для AM при $\alpha = 0.2$ и $\varepsilon_4^{(n)}$. При $\alpha = 0.1$ наблюдаются аналогичные результаты. Для GSM ($2k = 6$) при $\alpha = 0.1$ использование STEA2 позволяет получить результаты лучше, чем STEA1, что показано на фиг. 7б. Ступенчатая структура графика на фиг. 7б наглядно демонстрирует квадратичную сходимость.

Квадратичная сходимость GSM также наблюдается для MMPE, MPE и RRE, что доказано в работе [44].

5.2.4. Матричные уравнения. Рассмотрим симметричное матричное уравнение Стейна, известное также как уравнение Ляпунова с дискретным временем:

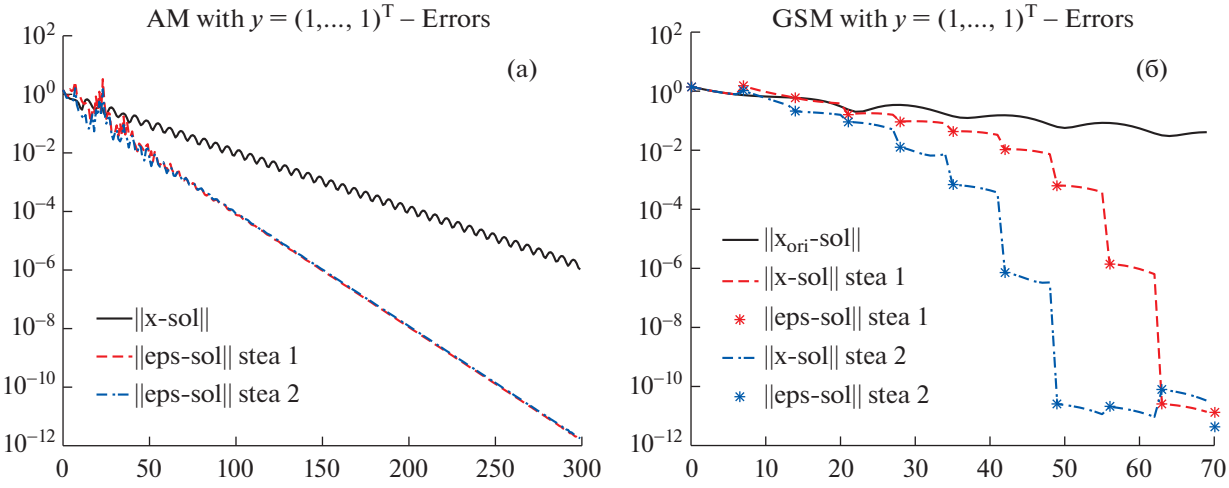
$$S - ASA^T = FF^T,$$

$F \in \mathbb{R}^{m \times s}$, $s \ll m$, при собственных значениях A, находящихся внутри единичного круга.

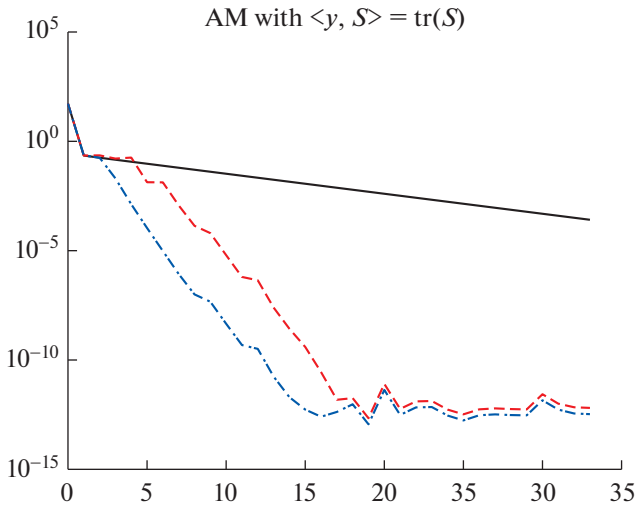
Известен итерационный метод, позволяющий получить численное решение (см. [25]):

$$S_{n+1} = FF^T + AS_nA^T,$$

$n = 0, 1, \ldots$, при $S_0 = 0$.



Фиг. 7. Нелинейная система: (а) – для AM, (б) – для GSM.



Фиг. 8. Уравнение Стейна.

Результаты на фиг. 8 соответствуют матрице `moler` размерности 500 для матрицы A, нормированной на скалярное значение такое, что ее спектральный радиус стал равен 0.9. Матрица `moler` – симметричная положительно-определенная матрица с одним малым собственным значением. Ее элементы определяются как $a_{ij} = \min(i, j) - 2$ и $a_{ii} = i$. Матрица F – `parter` матрица размерности 500×30. Данные две матрицы принадлежат набору примеров MATLAB© `gallery`. При $2k = 6$ для упрощенных топологических ε-алгоритмов, где дуальное произведение с функционалом y соответствует следу матрицы, мы применяем стратегию AM.

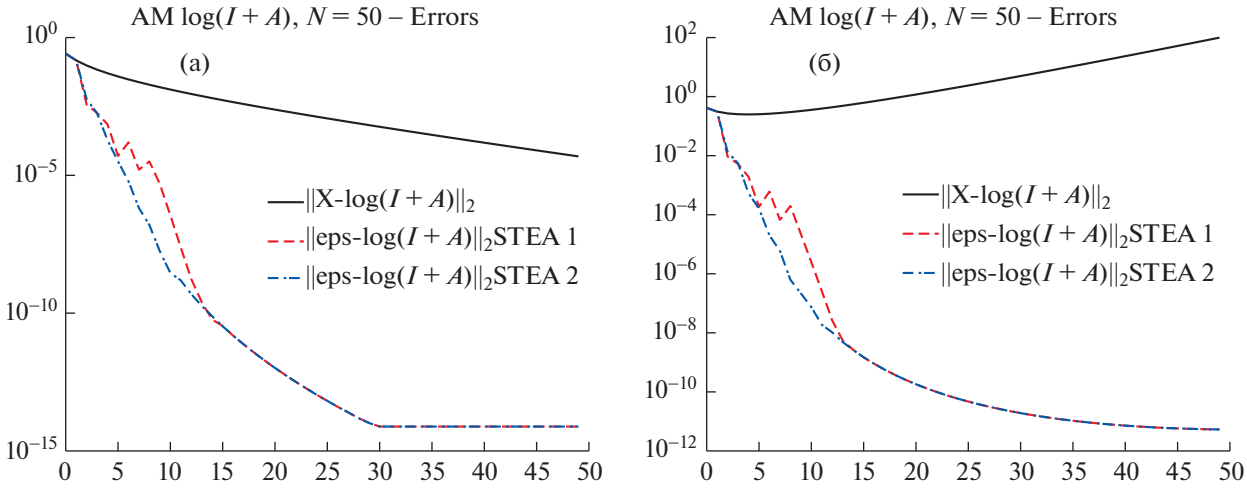
Сплошная линия на фиг. 8 соответствует итерационному методу, штриховая – алгоритму STEA1, а штрихпунктирная – методу STEA2.

5.3. Матричные функции

Рассмотрим задачу вычисления:

$$\log(I + A) = A - \frac{A^2}{2} + \frac{A^3}{3} - \frac{A^4}{4} + \ldots + (-1)^{n+1}\frac{A^n}{n} + \ldots,$$

где A – вещественная матрица.



Фиг. 9. Вычисление $\log(I + A)$: (а) – для $\rho(A) = 0.9$, (б) – для $\rho(A) = 1.2$.

Данный ряд сходится, если спектральный радиус матрицы A строго меньше единицы.

В данном примере мы покажем, что упрощенные топологические ε-алгоритмы позволяют ускорить сходимость частичных сумм данного ряда, а также получить сходимость к приближенному значению в случае расходимости исходного ряда.

Выберем квадратную случайную матрицу B размерности 50, значение r и определим

$$A = r \times B/\rho(B).$$

Таким образом, спектральный радиус A равен r.

Используя упрощенные топологические ε-алгоритмы при $\langle y, \cdot \rangle = \mathrm{tr}(\cdot)$, при $\varepsilon_8^{(n)}$ мы получаем результаты для частичных сумм ряда, представленные на фиг. 9, соответствующие $r = 0.9$ (случай сходимости, на фиг. 9а) и $r = 1.2$ (случай расходимости, на фиг. 9б). На графиках изображены значения евклидовой нормы погрешности с использованием функции `logm` для языка программирования MATLAB (которая позволяет вычислить основной матричный логарифм).

6. ТЕНЗОРЫ

Далее продемонстрируем два примера применения топологического ε-алгоритма и его упрощенной формы к задачам, связанным с тензорами (известных также как гиперматрицы). В нашей работе под тензором понимается многомерный массив.

Пусть $\mathbf{T} = (t_{i_1,\ldots,i_d}) \in \mathbb{R}^{n\times\cdots\times n}$ – вещественный кубический тензор с d измерениями размерности n. Его можно кратко обозначить через $\mathbf{T} \in \mathbb{R}^{[d,n]}$. Конечно, в случае $d = 1$ тензор является вектором, а при $d = 2$ – матрицей.

Для любого кубического тензора $\mathbf{T}$ и для любого вектора $\mathbf{x}$ размерности n можно записать вектор $\mathbf{y}$, вводя понятие умножение тензора $\mathbf{T}$ на вектор $\mathbf{x}$. Сделать это можно с помощью следующего отображения:

$$T : \mathbb{R}^n \to \mathbb{R}^n, \quad \mathbf{x} \mapsto T(\mathbf{x})_{i_1} = \sum_{i_2,\ldots,i_d} t_{i_1,i_2,\ldots,i_d} x_{i_2} \cdots x_{i_d},$$

для $i_1 = 1, \ldots, n$.

6.1. Тензорные ℓ^p-собственные пары

Так как T переводит $\mathbb{R}^n$ в себя, мы можем ввести концепцию собственных пар для $\mathbf{T}$ через отображение T. Вещественное число $\lambda \in \mathbb{R}$ будем считать ℓ^p-собственным значением $\mathbf{T}$ [45] (при $p > 1$), соответствующим ℓ^p-собственному вектору $\mathbf{x} \in \mathbb{R}^n$, если

$$T(\mathbf{x}) = \lambda \Phi_p(\mathbf{x}), \quad \|\mathbf{x}\|_p = 1,$$

где $\|\mathbf{x}\|_p$ соответствует обычно ℓ^p-норме:

$$\|\mathbf{x}\|_p = \left(|x_1|^p + \ldots + |x_n|^p\right)^{1/p}$$

и отображение $\Phi_p : \mathbb{R}^n \to \mathbb{R}^n$ определяется поэлементно $\Phi_p(\mathbf{x})_i = |x_i|^{p-2} x_i = \operatorname{sign}(x_i)|x_i|^{p-1}$ при $i = 1, \ldots, n$. Если $p \neq d$, то ℓ^p-собственные векторы нельзя определить с точностью до скалярного множителя. По этой причине мы добавляем условие нормировки $\|\mathbf{x}\|_p = 1$.

Тензор $\mathbf{T} = (t_{i_1,\ldots,i_d})$ будем называть неотрицательным и писать $\mathbf{T} \geq 0$, если $t_{i_1,\ldots,i_d} 0$ для всех $i_j = 1, \ldots, n$ и $j = 1, \ldots, d$. Далее можно ввести несколько обобщений понятия матричного спектрального радиуса для тензоров (см., например, [46]). В этой работе мы используем следующее определение:

$$r_p(\mathbf{T}) = \sup\{|\lambda| \lambda\text{: является } \ell^p\text{-собственным значением } \mathbf{T}\}.$$

Доказано, что для кубических неотрицательных тензоров $\mathbf{T}$ таких, что $T(\mathbb{1})$ поэлементно положителен, где $\mathbb{1}$ – вектор из единиц, при $p > d$ выполнено $r_p(\mathbf{T}) > 0$, и существует единственный поэлементно положительный $\mathbf{u} \in \mathbb{R}^n$ такой, что $\|\mathbf{u}\|_p = 1$ и $T(\mathbf{u}) = r_p(\mathbf{T})\Phi_p(\mathbf{u})$.

Определим дополнительно:

$$\mathbf{x} \mapsto f_p(\mathbf{x}) = \frac{\mathbf{x}^{\mathrm{T}} T(\mathbf{x})}{\|\mathbf{x}\|_p^d}, \quad \mathbf{x}^{\mathrm{T}} T(\mathbf{x}) = \sum_{i_1,\ldots,i_d} t_{i_1,i_2,\ldots,i_d} x_{i_1} \cdots x_{i_d}.$$

Для вычисления максимальной ℓ^p-собственной пары тензора $\mathbf{T}$ степенной метод, возможно, лучший из известных подходов. Начнем с $\mathbf{x}_0 \in C_+(\mathbf{T})$, где $C_+(\mathbf{T})$ – конус неотрицательных векторов, имеющих такой же шаблон нулей, как $T(\mathbb{1})$. Выберем $p > d$, точность $\varepsilon > 0$ и $q = p/(p-1)$ (сопряженный показатель). Степенной метод, записанный ниже, позволяет получить последовательность (λ_k), сходящуюся к требуемому собственному значению, а последовательность $(\mathbf{x}_k)$ сходится к соответствующему собственному вектору при любом выборе начального условия $\mathbf{x}_0 \in C_+(\mathbf{T})$:

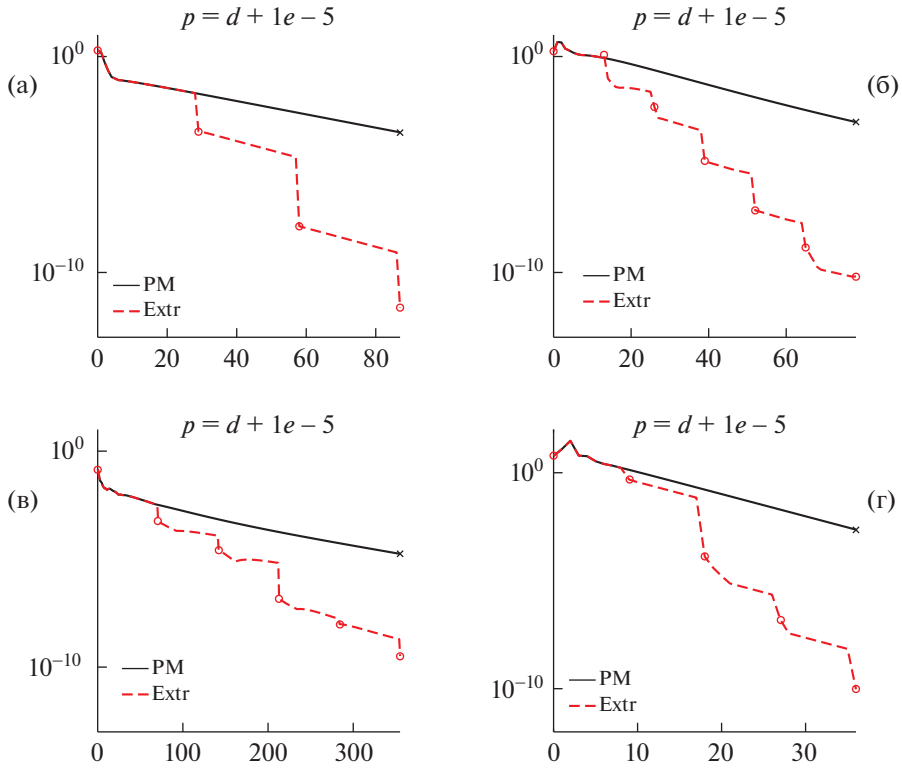
For $k = 0, 1, 2, 3, \ldots$ **repeat**
$\mathbf{y}_{k+1} = \Phi_q(T(\mathbf{x}_k))$
$\mathbf{x}_{k+1} = \mathbf{y}_{k+1} / \|\mathbf{y}_{k+1}\|_p$
$\lambda_{k+1} = f_p(\mathbf{x}_{k+1})$
until $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_p < \varepsilon$

Данный алгоритм может работать достаточно медленно, что делает его неприменимым для решения реальных прикладных задач, например, для работы с тензорами, соответствующими сетевым данным. Таким образом, для ускорения сходимости последовательности $(\mathbf{x}_k)$ мы используем алгоритм STEA2 с техникой рестарта.

На графиках ниже (фиг. 10) мы отображаем результаты вплоть до 30-й итерации для поэлементной невязки собственного вектора

$$\|T(\mathbf{x}_k) - \lambda_k \Phi_p(\mathbf{x}_k)\|_\infty,$$

как для исходной последовательности, так и для экстраполированной, и мы “подчеркиваем” крупными точками каждый рестарт внешнего цикла, например, невязку, сгенерированную ите-



Фиг. 10. Тензор ℓ^p-собственных векторов. Первая строка: `dolphins` ($n = 62$) и `yeast` ($n = 2361$). Вторая строка: `gre1107` ($n = 1107$) и `wb - cs - stanford` ($n = 9914$).

рациями экстраполяции. Мы фиксируем $p = d + 10^{-5}$ и устанавливаем $\mathbf{x}_0$ случайным вектором. С выбранным p мы можем вычислить приближение H-собственных пар (соответствующих $p = d$), так как в данном случае сходимость не гарантируется.

Линейный функционал $\mathbf{y}$ обновляется в конце каждого внешнего цикла как $\mathbf{y} = \tilde{\mathbf{e}}_h(\mathbf{x}_0)$ (для первого шага экстраполяции мы выбираем $\mathbf{y} = \mathbf{x}_0$).

В табл. 2 показаны четыре примера реальных сложных сетей, описываемых с помощью неотрицательных тензоров $\mathbf{T}$ третьего порядка, порожденных данными из работы [47].

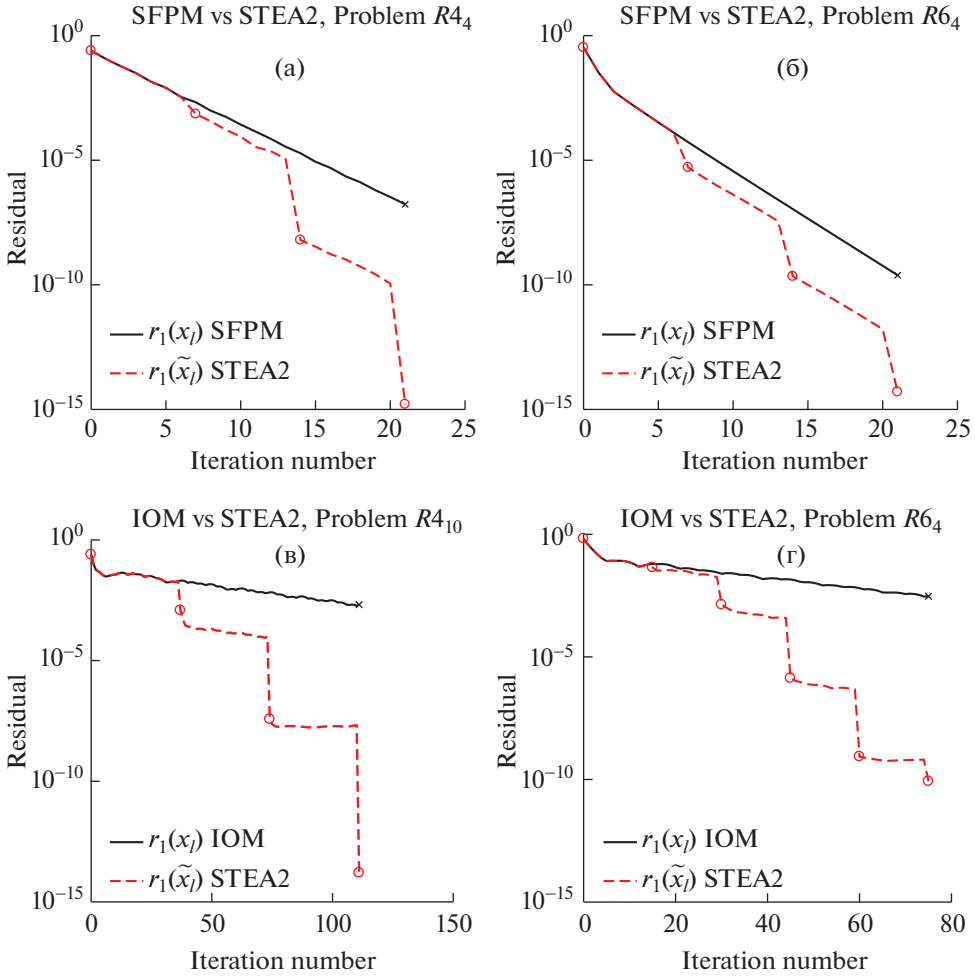
Больше подробностей, информацию о других алгоритмах и примерах для данного типа задач можно найти в [48].

6.2. Вычисление полилинейного PageRank

Недавно идея PageRank была обобщена для случая многомерных марковских цепей [49]. Исходное многомерное распределение PageRank аппроксимируется так называемым полилиней-

Таблица 2. Характеристики сетей, описываемых с помощью неотрицательных тензоров $\mathbf{T}$ третьего порядка, порожденных данными из работы [47]

Название задачи	Размер
`dolphins` (undirected)	62
`yeast` (undirected)	2361
`gre1107` (directed)	1107
`wb - cs - stanford` (directed)	9914



Фиг. 11. Полилинейный PageRank. Первая строка – *метод простой итерации со сдвигами* SFPM, $\alpha = 0.499$, $2k = 6$. Вторая строка – *метод внутренних и внешних итераций* IOM, $\alpha = 0.99$, $2k = 36$ (в) и $2k = 14$ (г).

ным PageRank, а задача поиска истинного многомерного стационарного распределения заменяется задачей, связанной с вычислением симметричного тензора ранга один.

В рамках предложенной модели требуется решить следующую задачу о неподвижной точке:

$$\alpha \mathbf{T}\mathbf{x}^{d-1} + (1-\alpha)\mathbf{v} = \mathbf{x}, \quad \|\mathbf{x}\|_1 = 1,$$

где $\mathbf{T}$ – стохастический тензор и $\mathbf{v}$ – поэлементно положительный стохастический вектор, и где по определению из предыдущего раздела $\mathbf{T}\mathbf{x}^{d-1} = T(\mathbf{x})$.

Для решения данной задачи в работе [50] рассмотрены два метода решения задачи о неподвижной точке: *метод простой итерации со сдвигами* (кратко SFPM) [49] в формулировке, адаптированной для конкретного случая вычисления полилинейного PageRank, а также *многомерный степенной метод со сдвигами* [51] и дополнительно *метод внутренних и внешних итераций* из работы [49]. Скорость сходимости может быть очень низкой. Таким образом, мы применили топологический ε-алгоритм с техникой рестарта для последовательностей $\mathbf{x}_\ell$, порожденных этими методами. Многочисленные численные эксперименты на синтетических и реальных данных демонстрируют улучшение работы перечисленных методов из-за применения методов экстраполяции. Мы демонстрируем результаты для нескольких задач из работы [49] при $d = 3$, $n = 4$ или $n = 6$, и где $y \in \mathbb{R}^n$ в STEA2 – последний экстраполированный член с предыдущего цикла. На фиг. 11 демонстрируются графики изменения относительной первой нормы невязки.

7. ЗАКЛЮЧЕНИЕ

Целью данной работы было показать, что методы экстраполяции могут быть весьма полезными инструментами при решении различных задач численного анализа и прикладной математики. Мы сосредоточились на изучении преобразований Шэнкса и ε-алгоритмах для их реализации. Дополнительно мы демонстрируем ряд конкретных примеров, иллюстрирующих выгоду использования методов экстраполяции. Мы надеемся, что эта статья будет стимулировать исследователей использовать эти алгоритмы в будущем. Соответствующие программные реализации алгоритмов на языке программирования MATLAB доступны в открытом доступе.

СПИСОК ЛИТЕРАТУРЫ

1. *Brezinski C., Redivo-Zaglia M.* Extrapolation Methods: Theory and Practice. Amsterdam: North–Holland, 1991.
2. *Delahaye J.P.* Sequence Transformations. Berlin: Springer–Verlag, 1988.
3. *Marchuk G.I., Shaidurov V.V.* Difference Methods and Their Extrapolations. New York: Springer–Verlag, 1983.
4. *Sidi A.* Practical Extrapolation Methods: Theory and Applications. Cambridge: Cambridge University Press, 2003.
5. *Sidi A.* Vector Extrapolation Methods with Applications. Philadelphia: Society for Industrial and Applied Mathematics, 2017.
6. *Weniger E.J.* Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series // Comput. Phys. Rep. 1989. V. 10. № 5. P. 189–371.
7. *Wimp J.* Sequence Transformations and Their Applications. New York: Academic Press, 1981.
8. *Shanks D.* Nonlinear Transformations of Divergent and Slowly Convergent Sequences // J. of Math. and Phys. 1955. V. 34. P. 1–42.
9. *Graves-Morris P.R., Jenkins C.D.* Vector-valued, rational interpolants III // Constructive Approximation. 1986. V. 2. P. 263–289.
10. *Shanks D.* An analogy between transient and mathematical sequences and some nonlinear sequence-to-sequence transforms suggested by it. Part I. White Oak: Naval Ordnance Laboratory, 1949.
11. *Brezinski C., Crouzeix M.* Remarques sur le procédé Δ^2 d'Aitken // C. R. Acad. Sci. Paris. 1970. P. 896–898.
12. *Brezinski C., Redivo-Zaglia M.* The genesis and early developments of Aitken's process, Shanks' transformation, the ε-algorithm, and related fixed point methods // Numerical Algorithms. 2019. V. 80. P. 11–133.
13. *Wynn P.* On a Device for Computing the $e_m(S_n)$ Transformation // Math. Tables and Other Aids to Comput. 1956. V. 10. № 54. P. 91–96.
14. *Wynn P.* Singular rules for certain non-linear algorithms // BIT Numerical Math. 1963. V. 3. P. 175–195.
15. *Baker G.A., Jr., Graves-Morris P.R.* Padé Approximants, 2nd Edition. Cambridge: Cambridge University Press, 1996.
16. *Wynn P.* Acceleration Techniques for Iterated Vector and Matrix Problems // Math. of Comput. 1962. V. 16. № 79. P. 301–322.
17. *Salam A.* Non-commutative extrapolation algorithms // Numerical Algorithms. 1994. V. 7. P. 225–251.
18. *Brezinski C., Redivo-Zaglia M.* Matrix Shanks Transformations // Electronic Journal of Linear Algebra. 2019. V. 35. P. 248–265.
19. *Graves-Morris P.R., Jenkins C.D.* Generalised inverse vector valued rational interpolation // Padé Approximat. and its Appl. Bad Honnef 1983. Lecture Notes in Math. 1984. P. 144–156.
20. *McLeod J.B.* A note on the ε-algorithm // Comput. 1971. V. 7. P. 17–24.
21. *Artin E.* Geometric Algebra. New York: Interscience, 1957.
22. *Porteous I.R* Topological Geometry, Second Edition. New York: Cambridge University Press, 1981.
23. *Salam A.* An algebraic approach to the vector ε-algorithm // Numerical Algorithms. 1996. V. 11. P. 327–337.
24. *Brezinski C.* Généralisation de la transformation de Shanks, de la table de Padé et de l'ε-algorithme // Calcolo. 1975. V. 12. P. 317–360.
25. *Brezinski C., Redivo-Zaglia M.* The simplified topological ε-algorithms for accelerating sequences in a vector space // SIAM Journal on Sci. Comput. 2014. V. 36. P. A2227–A2247.
26. *Brezinski C., Redivo-Zaglia M.* The simplified topological ε-algorithms: software and applications // Numerical Algorithms. 2017. V. 74. P. 1237–1260.
27. *Jbilou K.* Méthodes d'Extrapolation et de Projection. Applications aux Suites de Vecteurs. Thèse de 3ème cycle. Université des Sciences et Techniques de Lille, 1988.
28. *Jbilou K., Sadok H.* Some results about vector extrapolation methods and related fixed point iteration // J. Comp. Appl. Math. 1991. V. 36. P. 385–398.

29. *Cabay S., Jackson L.W.* A polynomial extrapolation method for finding limits and antilimits of vector sequences // SIAM J. Numer. Anal. 1976. V. 13. P. 734–752.
30. *Kaniel S., Stein J.* Least-square acceleration of iterative methods for linear equations // J. Optim. Theory Appl. 1974. V. 14. P. 431–437.
31. *Mešina M.* Convergence acceleration for the iterative solution of $x = Ax + f$ // Comput. Meth. in Appl. Mech. and Eng. 1977. V. 10. P. 165–173.
32. *Brezinski C., Redivo-Zaglia M., Saad Y.* Shanks Sequence Transformations and Anderson Acceleration // SIAM Review. 2018. V. 60. P. 646–669.
33. *Brezinski C., Redivo-Zaglia M.* EPSfun: a Matlab toolbox for the simplified topological ε-algorithm. Netlib (http://www.netlib.org/numeralgo/), 2017, na44 package.
34. *Le Ferrand H.* The quadratic convergence of the topological epsilon algorithm for systems of nonlinear equations // Numerical Algorithms. 1992. V. 3. P. 273–283.
35. *Steffensen J.F.* Remarks on iteration // Scandinavian Actuarial Journal. 1933. V. 1933. P. 64–72.
36. *Sidi A.* A convergence study for reduced rank extrapolation on nonlinear systems // Numerical Algorithms. 2020. V. 84. P. 957–982.
37. *Wynn P.* Transformations to accelerate the convergence of Fourier series // Gertrude Blanch Anniversary Volume. 1967. P. 339–379.
38. *Guilpin C., Gacougnolle J., Simon Y.* The ε-algorithm allows to detect Dirac delta functions // Appl. Numerical Math. 2004. V. 48. P. 27–40.
39. *Brezinski C.* Extrapolation algorithms for filtering series of functions, and treating the Gibbs phenomenon // Numerical Algorithms. 2004. V. 36. P. 309–329.
40. *Kaczmarz S.* Angenäherte Auflösung von Systemen linearer Gleichungen // Bull. Acad. Polon. Sci. 1937. V. 35. P. 355–357.
41. *Gastinel N.* Linear Numerical Analysis. New York: Acad. Press, 1970.
42. *Brezinski C., Redivo-Zaglia M.* Convergence acceleration of Kaczmarz's method // J. of Eng. Math. 2013. V. 93. P. 3–19.
43. *Brezinski C., Redivo-Zaglia M.* Extrapolation methods for the numerical solution of nonlinear Fredholm integral equations // J. Integral Equat. Appl. 2019. V. 31. № 1. P. 29–57.
44. *Jbilou K., Sadok H.* Vector extrapolation methods. Applications and numerical comparison // J. of Comput. and Appl. Math. 2000. V. 122. P. 149–165.
45. *Lim L.-H.* Singular values and eigenvalues of tensors: a variational approach // 1st IEEE Internat. Workshop on Comput. Advances in Multi-Sensor Adaptive Proc. 2005. P. 129–132.
46. *Gautier A., Tudisco F., Hein M.* The Perron-Frobenius theorem for multi-homogeneous mappings // SIAM J. Matrix Anal. Appl. 2019. V. 40. P. 1179–1205.
47. *Davis T.A., Yifan H.* The University of Florida Sparse Matrix Collection // ACM Trans. Math. Softw. 2011. V. 38. № 1. P. 1–25.
48. *Cipolla S., Redivo-Zaglia M., Tudisco F.* Shifted and extrapolated power methods for tensor ℓ^p-eigenpairs // Electron. Trans. Numer. Anal. 2020. V. 53. P. 1–27.
49. *Gleich D.F., Lim L.H., Yu Y.* Multilinear pagerank // SIAM J. Matrix Anal. Appl. 2015. V. 36. № 4. P. 1507–1541.
50. *Cipolla S., Redivo-Zaglia M., Tudisco F.* Extrapolation methods for fixed-point multilinear PageRank computations // Numer. Linear. Algebra. Appl. 2020. V. 27. P. e2280.
51. *Kolda T.G., Mayo J.R.* Shifted power method for computing tensor eigenpairs // SIAM J. Matrix Anal. Appl. 2011. V. 32. № 4. P. 1095–1124.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 744–758

ОБЩИЕ ЧИСЛЕННЫЕ МЕТОДЫ

УДК 519.61

ИНДУКТИВНОЕ ВОССТАНОВЛЕНИЕ МАТРИЦ С ОТБОРОМ ПРИЗНАКОВ[1)]

© 2021 г. М. Буркина[3], И. Назаров[1,*], М. Панов[1,], Г. Федонин[2,3,4], Б. Широких[1,2,3]**

[1] 121205 Москва, Большой бульвар, 30, стр. 1, Сколтех, Россия

[2] 127051 Москва, Б. Каретный пер., 19, стр. 1, ИППИ РАН, Россия

[3] 141700 Долгопрудный, М.о., Институтский пер., 9, МФТИ, Россия

[4] 111123 Москва, Новогиреевская ул., 3а, Центральный НИИ эпидемиологии, Россия

*e-mail: ivan.nazarov@skolkovotech.ru

**e-mail: m.panov@skoltech.ru

Поступила в редакцию 19.03.2020 г.
Переработанный вариант 29.12.2020 г.
Принята к публикации 14.01.2021 г.

Рассматривается задача индуктивного восстановления матриц – восстановления матрицы с использованием побочных признаков для строк и столбцов. Однако во многих прикладных задачах подобная вспомогательная информация содержит избыточные или малоинформативные признаки, что делает необходимым шаг их отбора. В работе предлагается подход, основанный на факторизации матрицы с групповой LASSO регуляризацией на коэффициенты побочных признаков, который совмещает отбор признаков с восстановлением матрицы. При этом теоретически доказывается, что асимптотика ошибки восстановления предложенного подхода ниже, чем в методах, не производящих прореживание. Предлагается вычислительно эффективная итеративная процедура для одновременного восстановления матрицы и отбора признаков. Эксперименты на искусственных данных и данных из прикладных задач демонстрируют, что предложенный подход улучшает показатели качества благодаря отбору признаков. Библ. 38. Фиг. 2. Табл. 3.

Ключевые слова: индуктивное восстановление матриц, групповое прореживание, асимптотика ошибки восстановления.

DOI: 10.31857/S0044466921050070

1. ВВЕДЕНИЕ

Методы пополнения или восстановления матриц, matrix completion, находят широкое применение в рекомендательных системах [1], [2], задачах кластеризации [3], классификации с многими метками [4], [5], обработки сигналов [6], компьютерного зрения [7] и подобных. В традиционной постановке частично наблюдаемая низкоранговая матрица восстанавливается напрямую через скалярное произведение выученных строчных и колоночных факторов, или неявно с использованием ядерных методов (kernel trick). При этом каждый фактор является численным отражением признаков некоей сущности, связанной с той или иной строкой или столбцом. Теоретические основания, определяющие возможность восстановления низкоранговых матриц, рассмотрены в работах [8] и [9]. Например, в условиях независимого случайного наблюдения достаточно иметь доступ к $O(N \log^2 N)$ элементам матрицы низкого ранга $n_1 \times n_2$ для полного точного ее восстановления ($N = \max\{n_1, n_2\}$). Оценка, не зависящая от распределения, приведенная в [10], позволяет достичь полного восстановления матриц, имея $O(N^{3/2})$ наблюдения.

Зачастую в дополнение к элементам самой частично наблюдаемой матрицы доступны вспомогательные экзогенные характеристики сущностей, стоящих за ее строками и столбцами. Например, в [11] показано, что побочные признаки строк и столбцов в виде профилей пользователей или описания жанров кино, side-channel information, полезны в задаче рекомендательной системы. Подобная вспомогательная информация играет особо важную роль при решении

[1)]Работа выполнена при финансовой поддержке РФФИ (код проекта 18-37-00489).

проблемы “холодного старта” рекомендательной системы, так как в ситуации, когда необходимо предсказать “взаимодействие” наблюдавшихся ранее сущностей с новой доселе ненаблюдавшейся сущностью, единственной возможностью остается только делать выводы на основе побочных характеристик.

Подходы, тем или иными способом учитывающие побочные признаки при пополнении матрицы, носят название индуктивного восстановления матриц, inductive matrix completion (IMC), см. в том числе [12]–[18]. Ключевой теоретический результат, полученный на момент написания настоящей работы, заключается в том, что достаточный объем наблюдений для восстановления снижается до $O(\log N)$ при условии, что побочные признаки имеют “хорошую” предсказательную силу. При этом вполне естественна ситуация, когда не все признаки релевантны или имеют хорошую предсказательную силу. Таким образом, становится очевидной необходимость разработки алгоритмов IMC, работающих в условиях избыточности побочных признаков и при этом сохраняющих гарантии по асимптотике достаточного объема наблюдений для восстановления. Немногие исследования обращаются к теме развития методов индуктивного восстановления матриц с отбором побочных признаков (прореживания), [16], [19] при том, что потенциал модификаций и улучшений существующих алгоритмов восстановления матриц в этом направлений велик.

Систематическое исследование проблемы индуктивного восстановления матриц началось с работы [12], в которой показано, что для полного восстановления достаточно наблюдать $O(\log N)$ случайных элементов при условии использования нормы $\|M\|_* = \sum_i \sigma_i(M)$ сингулярных чисел $\sigma_i(M)$ матрицы M в качестве регуляризатора, также известной как ядерная норма. В [20] рассмотрено представление матрицы в виде низкорангового произведения факторов и побочных признаков и изучена итеративная покоординатная процедура, возникающая в данной параметризации задачи. Предложенная параметризация нивелирует необходимость сингулярного разложения матрицы на каждой итерации, заменяя ее попеременным решением квадратичной задачи для каждого фактора. В [18] рассмотрены невыпуклый регуляризатор на факторы и 3-х фазная итеративная процедура восстановления, завершающаяся фазой проективного градиентного спуска, которая гарантирует, что вычисленные факторы находятся в окрестности оптимального решения задачи с высокой вероятностью по наблюдаемым выборкам.

В [14] рассматривается ситуация с несовершенными побочными признаками, которые не имеют достаточной предсказательной силы для полного восстановления матрицы, и, совместив индуктивный и классический подходы к восстановлению матриц, добиваются состоятельного восстановления матриц также и в случае “совершенных” побочных признаков. В [16] рассматривается аналогичная ситуация, где используется разреживающий регуляризатор на основе нормы сингулярных чисел целевой матрицы, приводя также оценки достаточного числа наблюдаемых элементов для восстановления. Однако в работе не исследуется эффект отбора признаков на получаемое решение, и предложенная процедура имеет высокую сложность по памяти и арифметическую сложность. В [19] рассматривается комбинаторная постановка задачи индуктивного восстановления матриц с отбором признаков, и приводится оценка ускоренной асимптотики достаточного объема наблюдений при наличии шума. В аналогичной постановке индуктивного восстановления ребер в графах в [21] получены минимакс оптимальные оценки достаточного объема наблюдений в режимах низкоранговой матрицы связности и избыточности побочных признаков, а также исследован баланс между вероятностью точного восстановления и вычислительной сложностью.

В настоящей работе мы предлагаем новый алгоритм индуктивного пополнения матриц, эффективно борющийся с избыточностью побочных признаков. Алгоритм отбирает релевантные характеристики при помощи включения регуляризатора в оптимизационную задачу факторизации матрицы, наводящего групповое прореживание на оцениваемые строчные и колоночные факторы. Мы приводим теоретические гарантии на оптимальность решения предложенной задачи IMC, которое, в свою очередь, ведет к улучшенной асимптотике достаточного объема наблюдений для индуктивного восстановления матрицы в ситуации, когда большая доля побочных признаков не имеют предсказательной силы. В частности, достаточный объем наблюдений асимптотически ниже, чем в случае, когда отбор признаков не производится.

Мы также предлагаем итеративную процедуру для численного решения предложенной невыпуклой оптимизационной задачи разреженного IMC, основанную на алгоритме ADMM [22], [23]. Приводимые нами экспериментальные свидетельства, демонстрируют, что предложенная процедура восстанавливает матрицу одновременно с отбором малоинформативных побочных

признаков как на синтетических примерах, так и в наборах данных из практических приложений, при этом достигая показателей качества, сравнимых с алгоритмическими аналогами без прореживания. Реализация процедуры (https://github.com/premolab/SGIMC) позволяет индуктивно восстанавливать частично наблюдаемые матрицы больших размеров.

Изложение результатов начинается с постановки оптимизационной задачи для восстановления матриц с прореживающим регуляризатором в разд. 2. В разд. 3 приводятся оценки обобщающей способности предложенного метода и достаточного объема наблюдений для восстановления, а в разд. 4 приведена предложенная нами итеративная процедура. Затем в разд. 5 приводятся и обсуждаются полученные экспериментальные свидетельства. Изложение завершается разд. 6.

2. ПОСТАНОВКА ЗАДАЧИ ВОССТАНОВЛЕНИЯ МАТРИЦ С ПРОРЕЖИВАНИЕМ

Рассмотрим целевую матрицу $M \in \mathbb{R}^{n_1 \times n_2}$, в которой наблюдаемы только значения M_{ij} и некоторого множества $(i, j) \in \Omega \subset \{1, \ldots, n_1\} \times \{1, \ldots, n_2\}$. Предположим, что побочные признаки полностью наблюдаемы и представлены в виде матриц $X \in \mathbb{R}^{n_1 \times d_1}$ и $Y \in \mathbb{R}^{n_2 \times d_2}$ для строк и столбцов M соответственно. Также допустим, что побочная информация имеет предсказательную силу применительно к значениям в M через билинейную модель

$$M_{ij} \sim x_i^{\mathrm{T}} W y_j$$

для некоторой матрицы весов $W \in \mathbb{R}^{d_1 \times d_2}$. Целью задачи индуктивного восстановления матрицы является оценка ненаблюдаемых элементов M на основе наблюдаемых M_Ω и побочной строчных X и столбцовых Y характеристик.

Главенствующим допущением в подходах к решению задачи IMC является предположение о том, что матрица W имеет низкий ранг $k < \min(d_1, d_2)$, см. [12], [14]. При этом существуют два подхода к учету данного ограничения в итоговой оптимизационной задаче.

Подход 1. Использование ℓ_1 нормы сингулярных чисел матрицы W, *ядерная норма* $\|W\|_*$, в качестве разреживающего регуляризатора, приводящего к низкоранговым решениям $\|W\|_*$, см. [12], [14], [16].

Подход 2. Явная параметризация W в виде низкорангового произведения UV^{T}, где $U \in \mathbb{R}^{d_1 \times k}$ и $V \in \mathbb{R}^{d_2 \times k}$, см. [18], [20].

Мы сосредотачиваемся на втором подходе, поскольку он позволяет работать с матрицей весов W неявно через ее факторы, что упрощает отбор признаков в задачах с большими объемами данных $(d_1, d_2 \gg 1)$.

Предположим, что $\mathscr{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ является гладкой выпуклой функцией потерь в задаче машинного обучения, и рассмотрим следующую регуляризованную задачу минимизации:

$$\min_{U,V} \sum_{(i,j)\in\Omega} \mathscr{L}(M_{ij}, (XUV^{\mathrm{T}}Y^{\mathrm{T}})_{ij}) + \lambda_U R(U) + \lambda_V R(V), \tag{1}$$

где $R(\cdot)$ – регуляризатор, и $\lambda_U, \lambda_V \geq 0$. Обычно в задачах восстановления матриц роль R играет квадрат нормы Фробениуса [14], которая имеет вид

$$\|Z\|_F^2 = \sum_{i=1}^{d} \sum_{j=1}^{k} z_{ij}^2$$

для некоторой матрицы $Z \in \mathbb{R}^{d \times k}$. В задаче линейной регрессии данная R эквивалентна регуляризации Тихонова. В настоящей работе мы предлагаем использовать прореживающий регуляризатор $R(Z)$ вида

$$\|Z\|_{2,1} = \sum_{i=1}^{d} \left\| e_i^{\mathrm{T}} Z \right\|_2,$$

где через e_i обозначается i-й единичный базисный вектор, чья размерность однозначна определяется контекстом выражения, в котором он участвует. Матричная функция $R(Z)$ вычисляет ℓ_1

норму вектора ℓ_2 норм строк Z. Подобный составной регуляризатор позволяет построчно прореживать матрицы U и V, т.е. наводит так называемую *групповую разреженность*, что в общем итоге создает эффект отбора побочных признаков в X и Y соответственно. Таким образом, задача индуктивного восстановления матриц с отбором признаков через групповую регуляризацию, Sparse-Group penalty Inductive Matrix Completion (SGIMC), имеет вид

$$\min_{U,V} \sum_{(i,j)\in\Omega} \mathscr{L}(M_{ij}, (XUV^{\mathrm{T}}Y^{\mathrm{T}})_{ij}) + \lambda_U \|U\|_{2,1} + \lambda_V \|V\|_{2,1}. \tag{2}$$

Заметим, что численная процедура, приведенная в разд. 6, позволяет работать с комбинацией регуляризаторов. В частности, мы рассматриваем квадрат нормы Фробениуса $R(Z) = \|Z\|_{\mathrm{F}}^2$, а также поэлементную матричную L_1-норму $R(Z) = \|Z\|_{1,1} = \sum_{i=1}^{n}\sum_{j=1}^{d}|z_{ij}|$ для большего контроля над разреженностью итогового решения.

3. АНАЛИЗ АСИМПТОТИКИ ОШИБКИ ВОССТАНОВЛЕНИЯ

В данном разделе приводится анализ влияния отбора признаков на точность восстановления матриц.

Задачу индуктивного восстановления матрицы можно рассмотреть через призму машинного обучения с учителем. Действительно, наблюдаемые значения в разреженной матрице M_Ω можно рассматривать как случайную выборку значений неизвестной ненаблюдаемой матрицы W^*, полученной с использованием линейных измерений, построенных на побочных признаках X и Y. Таким образом, набор данных (X, Y, M_Ω) представляется в виде выборки $S = (x_t, y_t, b_t)_{t=1}^m$ размера $m = |\Omega|$ при фиксированном обходе индексов из Ω. Каждое значение b_t в S является результатом применения измерительного оператора A_t к W^*, который задан одноранговой матрицей $A_t = x_t y_t^{\mathrm{T}}$ размера $d_1 \times d_2$: $b_t = \langle A_t, W^*\rangle = x_t^{\mathrm{T}} W^* y_t$. С этого ракурса задача (2) эквивалентна оценке истинной W^* матрицей ранга k, заданной произведением U и V, поскольку

$$x^{\mathrm{T}}UV^{\mathrm{T}}y = \mathrm{tr}(yx^{\mathrm{T}}UV^{\mathrm{T}}) = \left\langle xy^{\mathrm{T}}, UV^{\mathrm{T}}\right\rangle = \langle A, W^*\rangle.$$

Для понимания того, возможно ли в постановке (2) оценить истинную матрицу W^* и тем самым восстановить целевую матрицу M, рассмотрим задачу (2) с дополнительными ограничениями:

$$\begin{gathered}\min_{U,V} \frac{1}{\Omega}\sum_{(i,j)\in\Omega} \mathscr{L}(M_{ij}, x_i^{\mathrm{T}} W y_j) = \frac{1}{m}\sum_{t=1}^{m} \mathscr{L}(b_t, \langle A_t, W\rangle), \\ \text{при условии } \ W = UV^{\mathrm{T}}, \quad \|U\|_{2,1} \le C_U, \quad \|V\|_{2,1} \le C_V,\end{gathered} \tag{3}$$

для некоторых неотрицательных C_U и C_V. Решение (3) в данной задаче эквивалентно отысканию функции, минимизирующей функционал

$$J : \mathscr{F} \to \mathbb{R} : f \mapsto \frac{1}{m}\sum_{t=1}^{m} \mathscr{L}(b_t, f(A_t)),$$

определенный на параметрическом классе функций $\mathscr{F}$, заданным

$$\mathscr{F} = \left\{ f : \mathbb{R}^{d_1\times d_2} \to \mathbb{R} : A \mapsto \left\langle A, UV^{\mathrm{T}}\right\rangle, (U,V) \in \Theta\right\}, \tag{4}$$

где Θ означает допустимое множество параметров, заданное прямым произведением замкнутых шаров по норме $\|\cdot\|_{2,1}$ с радиусами C_U и C_V соответственно для U и V:

$$\Theta = \left\{U \in \mathbb{R}^{d_1\times k} : \|U\|_{2,1} \le C_U\right\} \times \left\{V \in \mathbb{R}^{d_2\times k} : \|V\|_{2,1} \le C_V\right\}.$$

Для того, чтобы определить, может ли класс $\mathscr{F}$ обобщить конечный набор наблюдаемых значений M на всю матрицу, необходимо оценить асимптотику ошибки восстановления этим классом. Для этого рассмотрим распределение $\mathscr{D}$ на $\mathscr{X}\times\mathscr{T}$, где $\mathscr{X} \subset \mathbb{R}^{d_1\times d_2}$ – множество матриц $d_1 \times d_2$,

а $\mathcal{T}$ является множеством допустимых значений в матрице M. Поскольку в задаче индуктивного восстановления множество $\mathcal{Z}$ задано набором ограниченных по норме матриц ранга 1 вида $A = xy^{\mathrm{T}}$, рассмотрим такие распределения $\mathcal{D}$ над $\mathcal{X} \times \mathcal{Y} \times \mathcal{T}$, для которых справедливо, что пространства побочных признаков $\mathcal{X} \subset \mathbb{R}^{d_1}$ и $\mathcal{Y} \subset \mathbb{R}^{d_2}$ являются ограниченными множествами. Отображение $\mathcal{X} \times \mathcal{Y}$ в $\mathcal{Z}$ имеет вид $(x, y) \mapsto xy^{\mathrm{T}}$.

Оценка асимптотики ошибки восстановления также требует ограниченности функции потерь $\ell : \mathbb{R} \times \mathcal{T} \to \mathbb{R}$, относительно которой определяются понятия теоретического (ожидаемого) и эмпирического риска. В задаче восстановления бинарной матрицы M множество $\mathcal{T}$ равно $\{-1, +1\}$ и используется бинарная функция потерь (0–1 loss): $\ell(p, b) = 1_{p \neq b}$. Однако для анализа асимптотики ошибки в задаче восстановления вещественной матрицы $\mathcal{T} = [-B, +B]$ для некоторого $B > 0$ и функция потерь $\ell(p, b)$ равна $|p - b|^d$ при $d \geq 1$.

Определение 1 (Риск). Рассмотрим гипотезу, решающее правило или регрессионную функцию $f : \mathcal{Z} \to \mathbb{R}$. При заданном распределении $\mathcal{D}$, *теоретический* риск f задан выражением

$$R(f) = \mathop{\mathbb{E}}_{(z,b)\sim\mathcal{D}} \ell(f(z), b).$$

Для заданной выборки $S = (z_i, b_i)_{i=1}^m \sim \mathcal{D}$, *эмпирический* риск f вычисляется в виде

$$\hat{R}(f) = \mathop{\hat{\mathbb{E}}}_{(z,b)\sim S} \ell(f(z), b) = \frac{1}{m}\sum_{i=1}^m \ell\left(f(z_i), b_i\right),$$

где $\hat{\mathbb{E}}_{(z,b)\sim S}$ означает (условное) математическое ожидание над эмпирическим распределением, порожденным выборкой S.

Для того чтобы получить оценку асимптотики верхней границы теоретического риска $R(\hat{f})$ минимизирующей гипотезы $\hat{f} = \arg\min_{f\in\mathcal{F}} \hat{R}(f)$, также известной как *оценка обобщающей способности* класса $\mathcal{F}$, в задаче (3), которая является эквивалентным представлением задачи SGIMC, рассмотрим более простой класс линейных гипотез на некотором $\mathcal{K} \subset \mathbb{R}^q$:

$$\mathcal{H} = \left\{h : \mathcal{K} \to \mathbb{R} : v \mapsto \langle v, \beta\rangle,\ \beta \in \mathbb{R}^q,\ \|\beta\|_1 \leq C\right\}, \tag{5}$$

и оценим асимптотику теоретического риска для (5).

Заметим, что в силу того, что пространства $\mathbb{R}^{d_1\times d_2}$ и $\mathbb{R}^q$ изоморфны для $q = d_1 d_2$, класс гипотез $\mathcal{H}$ тождественен классу $\mathcal{F}_1$, более удобному для анализа в задаче оценки значений матриц:

$$\mathcal{F}_1 = \left\{f : \mathbb{R}^{d_1\times d_2} \to \mathbb{R} : A \mapsto \langle A, W\rangle,\ \|W\|_{1,1} \leq C,\ W \in \mathbb{R}^{d_1\times d_2}\right\},$$

где $\|\cdot\|_{1,1}$ означает поэлементную L_1-норму матрицы W. Таким образом, поскольку для $C = C_U C_V$ справедливо, что $\mathcal{F} \subset \mathcal{F}_1$, из оценки асимптотики верхней границы теоретического риска для класса $\mathcal{H}$ в (5) следует оценка асимптотики ошибки восстановления для класса $\mathcal{F}$ в (4). Вложение классов следует из следующего наблюдения: если W параметризована произведением $W = UV^{\mathrm{T}}$ ранга k, тогда норму $\|W\|_{1,1}$ можно ограничить сверху произведением норм $\|U\|_{2,1}$ и $\|V\|_{2,1}$. Действительно,

$$\|W\|_{1,1} = \sum_{i=1}^{d_1}\sum_{j=1}^{d_2}\left|\sum_{p=1}^{k} e_i^{\mathrm{T}} U e_p e_p^{\mathrm{T}} V^{\mathrm{T}} e_j\right| \leq \sum_{i=1}^{d_1}\sum_{j=1}^{d_2}\sum_{p=1}^{k}\left|e_i^{\mathrm{T}} U e_p\right|\left|e_j^{\mathrm{T}} V e_p\right| \leq$$
$$\leq \sum_{i=1}^{d_1}\left\|e_i^{\mathrm{T}} U\right\|_2 \sum_{j=1}^{d_2}\left\|e_j^{\mathrm{T}} V\right\|_2 = \|U\|_{2,1}\|V\|_{2,1}.$$

Основной результат работы [24] дает равномерную оценку верхней границы теоретического риска с использованием понятия радемахеровской сложности (Rademacher complexity) класса

гипотез [25]. Радемахеровская сложность класса $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ в условиях распределения $\mathcal{D}_{\mathcal{X}}$ на множестве $\mathcal{X}$ задается в виде

$$\mathcal{R}_m(\mathcal{H}) = \mathop{\mathbb{E}}_{S\sim\mathcal{D}_{\mathcal{X}}^m} \hat{\mathcal{R}}_S(\mathcal{H}), \tag{6}$$

где математическое ожидание берется по всем выборкам S размера m независимых одинаково распределенных случайных величин из $\mathcal{D}_{\mathcal{X}}$. При этом эмпирическая радемахеровская сложность класса $\mathcal{H}$ при условии заданной выборки $S = (z_i)_{i=1}^m \subset \mathcal{X}$ определяется в виде

$$\hat{\mathcal{R}}_S(\mathcal{H}) = \mathop{\mathbb{E}}_{\varepsilon\sim\{\pm1\}^m} \sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \varepsilon_i h(z_i), \tag{7}$$

где математическое ожидание обусловлено по выборке S и берется по случайным векторам $\varepsilon = (\varepsilon_i)_{i=1}^m$ независимых равномерных случайных величин из $\{-1, +1\}$. В настоящем анализе мы рассматриваем классы измеримых функций, для которых супремум в (7) измерим, что выполняется для линейных гипотез над конечномерными пространствами $\mathbb{R}^m$. Заметим, что используемое в настоящей работе определение радемахеровской сложности совпадает с определением в [26], которое отличается от определения из [24] и [25] отсутствием множителя 2.

Основная теорема об оценке обобщающей способности класса функций через радемахеровскую сложность из работ [25] и [26, теорема 3.1, стр. 35] предлагает оценку верхней границы теоретического риска, равномерную по гипотезам из $\mathcal{H}$.

Теорема 1 (переформулировка). *Рассмотрим ρ-липшицеву функцию потерь $\ell : \mathbb{R}\times\mathcal{T} \to [0,1]$, или бинарную функцию потерь ℓ с $\rho = 1/2$. Пусть $\mathcal{H}$ является классом функций из $\mathcal{X}$ в $\mathbb{R}$ и пусть $S = (z_i, b_i)_{i=1}^m$ задает выборку независимых одинаково распределенных случайных величин из $\mathcal{D}$ в пространстве $\mathcal{X}\times\mathcal{T}$. Тогда для любой $\delta \in (0,1)$ с вероятностью не ниже $1-\delta$ по выборкам $S \sim \mathcal{D}^m$ следующие неравенства выполняются одновременно (равномерно) для всех $h \in \mathcal{H}$*

$$R(h) \le \hat{R}(h) + 2\rho\mathcal{R}_m(\mathcal{H}) + \sqrt{\frac{\log\frac{2}{\delta}}{2m}},$$

$$R(h) \le \hat{R}(h) + 2\rho\hat{\mathcal{R}}_S(\mathcal{H}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

Таким образом, для того чтобы ограничить сверху теоретический риск, достаточно найти верхнюю границу эмпирической радемахеровской сложности $\hat{\mathcal{R}}_S(\mathcal{H})$. Для этого мы воспользуемся теоремой 2 из [24], которая приводит оценку сложности для класса линейных гипотез H, ограниченных шаром L_1-нормы (LASSO hypothesis).

Теорема 2 (LASSO). *Пусть задана фиксированная выборка $S = (z_i)_{i=1}^m$ из $\mathcal{X} \subset \mathbb{R}^q$ размера m. Тогда справедлива следующая оценка сверху для $\hat{\mathcal{R}}_S(\mathcal{H})$ из* (7)*:*

$$\hat{\mathcal{R}}_S(\mathcal{H}) \le \frac{C}{m}\left(2 + \sqrt{\log q}\right)\sqrt{2\sum_{i=1}^m \|z_i\|_\infty^2}. \tag{8}$$

Таким образом, для любой выборки $S = (x_t, y_t, b_t)_{t=1}^m$ размера $m = |\Omega|$ независимых одинаково распределенных элементов $\mathcal{X}\times\mathcal{Y}\times\mathcal{T}$ согласно распределению $\mathcal{D}$ с ограниченным носителем, из анализа выше и оценки (8) следует, что эмпирическая радемахеровская сложность класса гипотез $\mathcal{F}$ задачи SGIMC ограничена сверху выражением

$$\hat{\mathcal{R}}_{S_\Psi}(\mathcal{F}) \le \frac{C_U C_V}{m}\left(2 + \sqrt{\log d_1 d_2}\right)\sqrt{2\sum_{t=1}^m \|A_t\|_{\max}^2}, \tag{9}$$

где $\|A_t\|_{\max}$ равна L_∞ норме элементов матрицы A_t, а $S_{\mathscr{Z}} = (A_t)_{t=1}^m$ является проекцией выборки S на компоненту $\mathscr{Z}$, в которой $A_t = x_t y_t^{\mathrm{T}}$ преобразует точки в $\mathscr{X} \times \mathscr{Y}$ в выборку матриц из $\mathscr{Z}$. Для любого t значение нормы $\|A_t\|_{\max}$ матрицы ранга 1 ограниченно сверху произведением $\|x_t\|_\infty \|y_t\|_\infty$.

Для оценки асимптотики ошибки восстановления в задаче SGIMC воспользуемся оценкой (9) и теоремой 1 и, сделав некоторое допущение относительно гипотезы f^*, минимизирующей теоретический риск $R(f)$ по всем $f : A \mapsto \langle A, W \rangle$ и для всевозможных матриц $W \in \mathbb{R}^{d_1 \times d_2}$ ранга k. Допущение заключается в следующем: рассмотрим задачу минимизации теоретического риска с требованием низкорангового решения для распределения $\mathscr{D}$ с ограничением на носитель $\|x\|_\infty \le 1$ и $\|y\|_\infty \le 1$, но без ограничений на нормы факторов U и V:

$$\min_{U,V} \mathop{\mathbb{E}}_{(x,y,b)\sim\mathscr{D}} \ell\left(b, \left\langle xy^{\mathrm{T}}, UV^{\mathrm{T}} \right\rangle\right), \tag{10}$$

с $U \in \mathbb{R}^{d_1 \times k}$, $V \in \mathbb{R}^{d_2 \times k}$, где k равен рангу матрицы M, – и обозначим парой U_*, V_* решение задачи (10). Если задача восстановления матрицы реализуема, т.е. минимизирующая пара (U_*, V_*) задачи (10) достигает *нулевого* теоретического риска (с бинарной функцией потерь, или L_d), тогда для любой выборки независимых одинаково распределенных случайных величин из $\mathscr{D}$ пара $(\hat{U}, \hat{V})$, минимизирующая эмпирический риск, также его обнуляет. Действительно, если пара $(\hat{U}, \hat{V})$, зависящая от S, является решением задачи (3) с эмпирическим риском вместо теоретического, тогда имеем

$$\mathop{\hat{\mathbb{E}}}_{(x,y,b)\sim S} \ell\left(b, \left\langle xy^{\mathrm{T}}, U_* V_*^{\mathrm{T}} \right\rangle\right) \ge \mathop{\hat{\mathbb{E}}}_{(x,y,b)\sim S} \ell\left(b, \left\langle xy^{\mathrm{T}}, \hat{U}\hat{V}^{\mathrm{T}} \right\rangle\right).$$

Однако, поскольку ℓ является ограниченной сверху неотрицательной функцией потерь, то для любой пары (U, V) можно получить

$$\mathop{\mathbb{E}}_{(x,y,b)\sim\mathscr{D}} \ell\left(b, \left\langle xy^{\mathrm{T}}, UV^{\mathrm{T}} \right\rangle\right) = \mathop{\mathbb{E}}_{(x,y,b)\sim\mathscr{D}} \mathop{\hat{\mathbb{E}}}_{(x,y,b)\sim S} \ell\left(b, \left\langle xy^{\mathrm{T}}, UV^{\mathrm{T}} \right\rangle\right) \ge 0,$$

откуда следует, что в реализуемом случае оптимальный эмпирический риск также обнуляется.

Если рассматривается бинарная функция потерь ℓ, или липшицева функция потерь с показателем ρ, тогда в реализуемом случае задачи индуктивного восстановления матриц с отбором признаков через групповую регуляризацию оценка верхней границы теоретического риска

$$R_{\mathscr{D}}(U, V) = R_{\mathscr{D}}\left(\left\langle \cdot, UV^{\mathrm{T}} \right\rangle\right) = \mathop{\mathbb{E}}_{(x,y,b)\sim\mathscr{D}} \ell\left(b, \left\langle xy^{\mathrm{T}}, UV^{\mathrm{T}} \right\rangle\right)$$

является результатом следующей теоремы, являющейся основным результатом данного раздела.

Теорема 3 (Основной результат). *Рассмотрим задачу* (3), *в которой коэффициенты* $C_U = \|U_*\|_{2,1}$ *и* $C_V = \|V_*\|_{2,1}$ *определяются решением* $(U_*, V_*) \in \mathbb{R}^{d_2 \times k} \times \mathbb{R}^{d_2 \times k}$ *задачи* (10). *Тогда для любого* $\delta > 0$ *с вероятностью не ниже* $1 - \delta$ *справедлива следующая оценка верхней границы теоретического риска для пары* $(\hat{U}, \hat{V})$, *минимизирующей эмпирический риск*:

$$R_{\mathscr{D}}(\hat{U}, \hat{V}) \le C_U C_V \frac{2^{3/2}\rho}{\sqrt{|\Omega|}}\left(2 + \sqrt{\log d_1 d_2}\right) + 3\sqrt{\frac{\log 2/\delta}{2|\Omega|}},$$

где $|\Omega|$ *равно количеству наблюдаемых значений в целевой матрице* M.

Если предположить, что теоретический риск минимизируется разреженным решением, т.е. некоторые строки матриц U_* и V_* полностью нулевые строки, тогда их $L_{2,1}$ нормы можно ограничить

$$\|U_*\|_{2,1} \le s_1 \sqrt{k} u_\infty, \quad \|V_*\|_{2,1} \le s_2 \sqrt{k} v_\infty,$$

причем s_1 и s_2 определяются верхней оценкой количества ненулевых строк, а u_∞ и v_∞ равны максимальным по модулю значениям в U_* и, соответственно, в V_*. В условиях данного предположения выполняется следующее следствие теоремы 3.

Следствие 1. Если в дополнение к предпосылкам теоремы 3 предположить, что теоретический риск минимизируется разреженным решением (U_* и V_* имеют не более чем s_1 и s_2 ненулевых строки), тогда с вероятностью не ниже $1 - \delta$ справедлива следующая оценка верхней границы:

$$R_{\mathscr{D}}(\hat{U},\hat{V}) \le s_1 s_2 k u_\infty v_\infty \frac{2^{3/2}\rho}{\sqrt{|\Omega|}}\left(2 + \sqrt{\log(d_1 d_2)}\right) + 3\sqrt{\frac{\log 2/\delta}{2|\Omega|}},$$

u_∞ и v_∞ равны максимальным по модулю значениями в матрицах U_* и, соответственно, в V_*. Более того, если распределение $\mathscr{D}$ таково, что сами побочные признаки x и y почти наверное разрежены, тогда с вероятностью не ниже $1 - \delta$:

$$R_{\mathscr{D}}(\hat{U},\hat{V}) \le s_1 s_2 k u_\infty v_\infty \frac{2^{3/2}\rho}{\sqrt{|\Omega|}}\left(2 + \sqrt{\log(r_1 r_2)}\right) + \frac{2\rho}{\sqrt{|\Omega|}} + 3\sqrt{\frac{\log 2/\delta}{2|\Omega|}},$$

где r_1 и r_2 ограничивают сверху число ненулевых значений для каждого x и y.

Из асимптотики верхних оценок ошибок восстановления, выведенных выше, можно получить процедуру для решения задачи SGIMC с ограничениями (3), если сформулировать ее как задачу регуляризованной минимизации эмпирического риска для некоторых заданных радиусов C_U и C_V. Действительно, если предположить, что пара $(\hat{U},\hat{V})$ решает

$$\min_{U,V} \sum_{(i,j)\in\Omega} \mathscr{L}(M_{ij}, (XUV^{\mathrm{T}}Y^{\mathrm{T}})_{ij}) + \lambda_U \|U\|_{2,1} + \lambda_V \|V\|_{2,1},$$

где $\mathscr{L}$ является либо L_q функцией потерь, либо выпуклой мажорантой бинарной функции потерь в задаче классификации. Регуляризаторы вида $\|\cdot\|_{2,1}$ обеспечивают малость C_U и C_V в оценке верхней границы (0.3), что, в свою очередь, с высокой вероятностью приводит к низкому теоретическому риску оцененных матриц U и V ранга k.

Заметим, что в случае разреженных матриц истинных факторов U и V регуляризация с помощью нормы Фробениуса приводит к асимптотике ошибки восстановления вида

$$O(s_1 s_2 k^2 d_1 d_2 \log(d_1 d_2)/\varepsilon^2),$$

в то время когда предложенная выше оценка с высокой вероятностью дает асимптотику

$$O(s_1^2 s_2^2 k^2 \log(d_1 d_2)/\varepsilon^2),$$

что приводит к меньшему числу достаточных наблюдений в M_Ω в режиме высокого d и низкого s. Вдобавок, если побочные признаки сами по себе разрежены, то регуляризация нормой Фробениуса дает оценку

$$O(s_1 s_2 k^2 r_1 r_2 \log(r_1 r_2)/\varepsilon^2)$$

в то время как анализ, приведенный в данном разделе, дает асимптотику

$$O(s_1^2 s_2^2 k^2 \log(r_1 r_2)/\varepsilon^2).$$

В сравнении это означает, что различия в оценке количества достаточного числа наблюдаемых значений в M определяются соотношениями значений s_1, s_2, r_1 и r_2.

4. ПРОЦЕДУРА ВОССТАНОВЛЕНИЯ С ПРОРЕЖИВАНИЕМ

В данном разделе приводится описание итеративной вычислительной процедуры, решающей предложенную задачу индуктивного восстановления матриц с отбором побочных признаков (SGIMC).

Задача индуктивного восстановления матриц с отбором признаков через групповую регуляризацию (SGIMC) для набора данных (M_Ω, X, Y) может быть сформулирована в виде следующей

оптимизационной задачи: для заданного ранга $k \geq 1$ найти такие $U \in \mathbb{R}^{d_1 \times k}$ и $V \in \mathbb{R}^{d_2 \times k}$, которые доставляют минимум

$$J(U,V) = \sum_{(i,j)\in\Omega} \mathscr{L}(M_{ij}, e_i^{\mathrm{T}} XUV^{\mathrm{T}} Y^{\mathrm{T}} e_j) + \lambda_U \|U\|_{2,1} + \lambda_V \|V\|_{2,1}, \tag{11}$$

где $\mathscr{L}(y,p)$ является гладкой выпуклой функцией потерь. Для восстановления вещественной матрицы M функция потерь $\mathscr{L}(y,p)$ равна $\frac{1}{2}(y-p)^2$, а для восстановления бинарной матрицы со значениями ± 1 функция имеет вид $\mathscr{L}(y,p) = \log(1+e^{-yp})$.

Норма $\|U\|_{2,1}$ является групповым регуляризатором матрицы U, "мягко" отсекая строки U с низкой L_2, тем самым производя отбор побочных признаков, поскольку восстановление матрицы M происходит при помощи $XU = \sum_{p=1}^{d_1} (Xe_p)(U^{\mathrm{T}} e_p)^{\mathrm{T}}$ в (11). Заметим, что приводимая ниже итеративная процедура также позволяет регуляризовать матрицы факторов с помощью нормы Фробениуса и поэлементной L_1 нормы для поэлементного разрежения.

Задача (11) является би-выпуклой задачей, т.е. функция $J(U,V)$ выпукла по каждому аргументу в отдельности, но не в совокупности. Естественным методом в данном случае является покоординатный спуск [20] – попеременная циклическая минимизация сначала по U при фиксированной V, затем наоборот:

$$\begin{aligned} U_{t+1} &= \arg\min_{U\in\mathbb{R}^{d_1\times k}} J(U,V_t), \\ V_{t+1} &= \arg\min_{V\in\mathbb{R}^{d_2\times k}} J(U_{t+1},V), \end{aligned} \tag{12}$$

покуда относительное изменение $U_t V_t^{\mathrm{T}}$ между последовательными шагами итерации не станет ниже заранее установленного порога. В силу того, что по каждому аргументу по отдельности целевая функция строго выпуклая из-за регуляризации, решение каждой подзадачи (12) единственно, что означает сходимость итераций процедуры к стационарной точке [22].

Структура функции потерь и регуляризации (11) означает, что целевая функция J для набора данных (M_Ω, X, Y) совпадает с целевой функцией J^{T} для $(M_\Omega^{\mathrm{T}}, Y, X)$, в которой роли аргументов U и V поменяны местами (*транспонированная* задача). Таким образом, частная целевая функция $V \mapsto J(U,V)$ при фиксированном U тождественна $U \mapsto J^{\mathrm{T}}(V,U)$ для транспонированной задачи, что приводит к тому, что достаточно разработать итеративную процедуру для решения $\min_U J(U,V)$ для данных (M_Ω, X, Y) и фиксированного V, чтобы получить полную процедуру для покоординатного спуска (12).

Частная задача (11) по U при фиксированном V имеет вид

$$\min_{U\in\mathbb{R}^{d_1\times k}} \sum_{(i,j)\in\Omega} \mathscr{L}(M_{ij}, p_{ij}) + \lambda_U R(U), \tag{13}$$

где $p_{ij} = e_i^{\mathrm{T}}(XUQ^{\mathrm{T}})e_j$, а $Q = YV$ является матрицей $n_2 \times k$. Мы предлагаем численно решать задачу (13) с помощью Метода переменных множителей (Alternating Direction Method of Multipliers, ADMM), предложенного в [27] и [28], с гарантиями сходимости, исследованными в [29] и [30]. Применительно к (13) итерации метода принимают вид

$$U_{t+1} = \arg\min_U \sum_{\omega\in\Omega} \mathscr{L}(M_\omega, p_\omega) + \frac{\lambda_R}{2}\|U\|_F^2 + \frac{1}{2\eta}\|U - (Z_t - \Phi_t)\|_F^2, \tag{14}$$

$$Z_{t+1} = \arg\min_Z \lambda_U \|Z\|_{2,1} + \frac{1}{2\eta}\|Z - (U_{t+1} + \Phi_t)\|_F^2, \tag{15}$$

$$\Phi_{t+1} = \Phi_t + (U_{t+1} - Z_{t+1}),$$

где $\eta > 0$, двойственная переменная Φ является матрицей $d_1 \times k$ и $\frac{\lambda_R}{2}\|U\|_F^2$ является вспомогательным регуляризатором, обеспечивающим сильную выпуклость целевой функции на каждой итерации.

4.1. Вычисление шага для U

Шаг для U, (14), является задачей минимизации гладкой выпуклой функции с квадратичным регуляризатором. Для L_2 функции потерь решение этой подзадачи имеет явный вид, выводимый из решения метода наименьших квадратов, а для гладких выпуклых функций потерь более общего вида $\mathcal{L}$, или в условиях нецелесообразности обращения матриц, U-шаг решается численно. Аналогичная проблема без специфичного для ADMM квадратичного регуляризатора решается в работе [31] методом сопряженных градиентов с доверительной областью (TRON), предложенного в [32] для решения линейных моделей высокой размерности. Градиент и произведение гессиан-вектор для шага (14) из некоторой точки U, которые необходимы для метода сопряженных градиентов, приводятся соответственно в (16) и (17) (см. также [31]):

$$\operatorname{grad}_U = X^{\mathrm{T}}GQ + \left(\lambda_R + \frac{1}{\eta}\right)U - \frac{1}{\eta}(Z_t - \Phi_t), \tag{16}$$

$$\operatorname{Hess} V_U(D) = X^{\mathrm{T}}(H \odot (XDQ^{\mathrm{T}}))Q + \left(\lambda_R + \frac{1}{\eta}\right)D. \tag{17}$$

Здесь матрицы X и Q имеют тот же смысл, что в (13), $D \in \mathbb{R}^{d_1 \times k}$ является "вектором" для произведения гессиан-вектор, $\odot$ обозначает поэлементное умножение согласованных по размерности матриц (произведение Адамара). При этом $n_1 \times n_2$ матрицы G в (16) и H в (17) имеют паттерн разреженности, идентичный матрице M_Ω, и вычисляются соответственно через $\mathcal{L}'(M_{ij}, p_{ij})$ и $\mathcal{L}''(M_{ij}, p_{ij})$ для $(i, j) \in \Omega$.

В работе [31] предложены ускоренные процедуры для ключевых матричных операций, необходимых для вычисления (16) и (17). Операция $S \mapsto X^{\mathrm{T}}SQ$, происходящая в обоих выражениях, отображает $n_1 \times n_2$ разреженную матрицу S с индексами Ω в плотную матрицу $d_1 \times k$, в то время как операция $D \mapsto XDQ^{\mathrm{T}}$ переводит плотную $d_1 \times k$ матрицу D в разреженную матрицу размера $n_1 \times n_2$ с индексами Ω, т.е. имеющую разреженность, идентичную целевой матрице M. Для плотной матрицы Q обе опреации имеют арифметическую сложность порядка $O(k|\Omega| + k \cdot \operatorname{nnz}(X))$, где $\operatorname{nnz}(X)$ равно $n_1 d_1$ при условии плотной матрицы X, и количеству ненулевых значений, если X разрежена.

4.2. Решение шага для Z

Целевая функция на шаге Z (15) раскладывается на d_1 независимых подзадач, по одной на каждую строку матрицы Z:

$$z_j = \arg\min_z \frac{1}{2}\|z - a_j\|_2^2 + \eta\lambda_U \|z\|_2, \quad j = 1, \ldots, d_1,$$

где $a_j = e_j^{\mathrm{T}}(U_{t+1} + \Phi_t)$. Каждая задача j имеет явное решение, вычислимое через оператор группового сжатия $z_j = \max\left\{1 - \frac{\eta\lambda_U}{\|a_j\|_2}, 0\right\} a_j$ (group shrinkage operator) [33].

5. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

В данном разделе приводится экспериментальное сравнение предложенного метода индуктивного восстановления матриц с групповым регуляризатором (SGIMC) с методом IMC, предложенным в [31], а также со стандартным методом восстановления матриц (MF), основанного на факторном разложения матриц, при помощи стохастического градиентного спуска. Раздел начинается с численного эксперимента на искусственных данных, нацеленного на изучение эффектов гиперпараметров задачи и ее размерности на качество восстановления матриц. Затем алгоритмы индуктивного восстановления матриц сравниваются в задаче кластеризации и восстановления матриц на реальных наборах данных. Мы не сравниваемся с процедурой IMC из работы [16], по причине того, что существующая реализация этого алгоритма не была работоспособной даже на задачах самой низкой размерности из рассматриваемых.

5.1. Искусственные данные

В экспериментах с искусственными данными мы рассматриваем задачу индуктивного восстановления разреженной $n_1 \times n_2$ матрицу M, наблюдаемые значения которой находятся по индексам Ω. Качество восстановления метода определяется по наименьшему значению целевой метрики, рассчитанному по значениям элементов матрицы M, отсутствующих в M_Ω. Сама целевая метрика равна наименьшей относительной ошибке восстановления $\left\|\hat{M} - M\right\|_F \big/ \left\|M\right\|_F$ с $\hat{M} = X\hat{U}\hat{V}^{\mathrm{T}}Y^{\mathrm{T}}$ по всем значениям коэффициентов регуляризации $\lambda = \lambda_U = \lambda_V$ в задаче (1) из некоторого множества. Задачей данного эксперимента является получение ответа на вопрос, помогает ли встроенный в SGIMC отбор побочных признаков при восстановлении матрицы, а также на вопрос, сравнима ли предложенная процедура с аналогами по качеству в ситуации отсутствия избыточности в побочных признаках. Детали проведенных экспериментов заключаются в следующем: побочные признаки X и Y задаются независимыми случайными гауссовскими матрицами размерности $n_1 \times d$ и $n_2 \times d$ с распределением $\mathcal{N}(0, 0.05)$, и значениями $n_1 = 800$, $n_2 = 1600$ при разных d. Истинные значения факторов U^* и V^* задаются первыми $k = 25$ колонками единичной матрицы размера $d \times d$, создавая тем самым ситуацию, когда истинное число информативных побочных признаков d^* совпадает с k. Сама целевая матрица равна $M = XU^*(YV^*)^{\mathrm{T}} + \varepsilon$, где $\varepsilon \sim \mathcal{N}(0, 0.005)$.

Коэффициенты регуляризации λ_U и λ_V приравниваются и выбираются из множества $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. При этом в каждом эксперименте варьируются также следующие параметры:

- предполагаемый ранг $\hat{k}$ матрицы M либо недооценивает (20) истинный ранг $k = 25$, либо переоценивает его (30);
- число избыточных неинформативных побочных признаков равно $d - d^* \geq 0$;
- показатель разреженности матрицы M_Ω, т.е. доля наблюдаемых значений в ней, выбираются из $\rho = \dfrac{|\Omega|}{n_1 n_2}$.

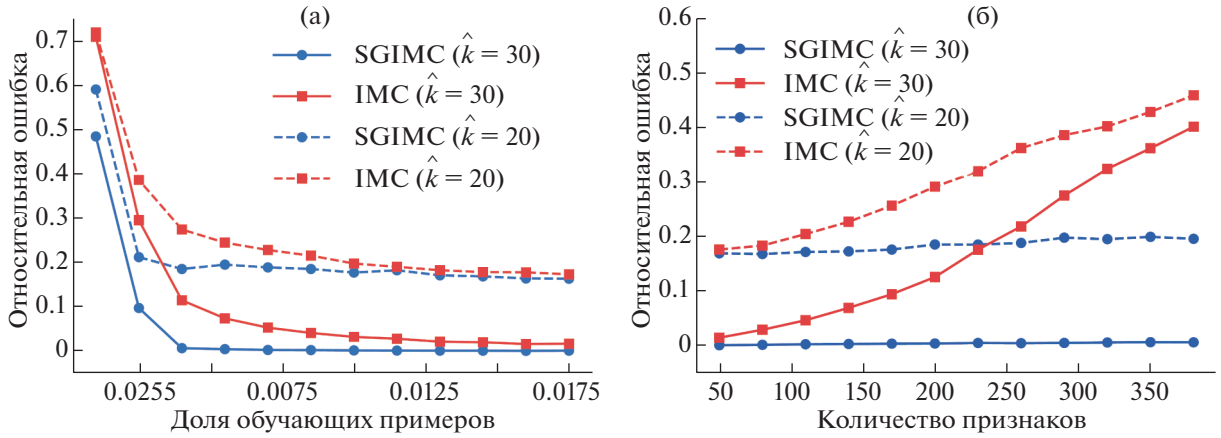
5.1.1. Показатель разреженности ρ. В данном эксперименте число признаков фиксировано $d = 100$ и предполагаемый ранг матрицы совпадает с истинным рангом $\hat{k} = k$. Показатель разреженности матрицы ρ пробегает значения от 0.0005 до 0.02 с шагом 0.0015. В данной постановке методы IMC и SGMIC достигают почти точного восстановления, когда матрица M достаточно плотная (показатель $\rho > 0.1$). Приведенные на фиг. 1а результаты работы методов в режиме $\rho < 0.01$ показывают, что SGIMC требует меньшего объема наблюдаемых значений в матрице M по сравнению с IMC для достижения сопоставимых ошибок восстановления. В то же самое время переоценка ранга также позволяет получить почти точное восстановление M.

5.1.2. Избыточные признаки. В этом эксперименте показатель ρ зафиксирован на уровне 0.2 и число побочных признаков d варьируется от 50 до 400 с шагом 50. Добавленные избыточные признаки сверх первых d^* являются случайным шумом, значения целевой матрицы от которого не зависят. Это позволяет проверить качество отбора признаков в присутствии полностью неинформативных побочных признаков. Фиг. 1б демонстрирует, что процедура SGIMC отличает информативные признаки от малоинформативных, и показывает систематически хороший результат как в режиме переоценки, так и недооценки ранга.

5.2. Данные из прикладных задач

В этом разделе мы применяем процедуры IMC, SGIMC и MF на реальных наборах данных для того, чтобы сравнить их качества восстановления.

5.2.1. Кластеризация с примерами. Рассмотрим задачу кластеризации через обучение на примерах, или, иными словами, задачу выявления классов эквивалентности между объектами. Имеется матрица X признаков размера $n \times d$ для n сущностей и задача состоит в том, чтобы построить бинарный классификатор, определяющий, принадлежат ли сущности i и j одному и тому же классу или разным классам. Таким образом, исходный набор данных состоит из матрицы M с $M_{ij} = 1$, если i и j принадлежат одному и тому классу, и -1 в противном случае.



Фиг. 1. Относительная ошибка восстановления в эксперименте с искусственными данными: а – изменение ρ разреженности матрицы M_Ω, б – добавление малоинформативных признаков $d > d^*$.

Были выбраны три набора данных из [34] для кластеризации с примерами с помощью индуктивного восстановления матрицы: “Mushrooms”, “Segment” и “Covtype” в табл. 1. Заметим, что по причине сильной несбалансированности набор “Covtype” был предобработан для балансировки классов с помошью случайного сэмплирования из доминирующего класса и полного сохранения класса, находящегося в меньшинстве.

Ввиду того, что матрица попарной принадлежности M доступна полностью, каждый набор данных случайно разбивается на обучающую и тестовую выборки, причем доля обучающих примеров варьируется от 0.0005 до 0.02 из общего числа наблюдений. Качество восстановления матрицы измеряется точностью классификации на тестовой подвыборке.

Эксперименты показывают, что матрицы попарной принадлежности каждого набора данных имеют низкий ранг, и что качество кластеризации существенно зависит от $\hat{k} = 2, \ldots, 20$. Стратегия выбора оптимального значения коэффициентов регуляризации и усреднения по независимым повторениям эксперимента аналогична п. 13.

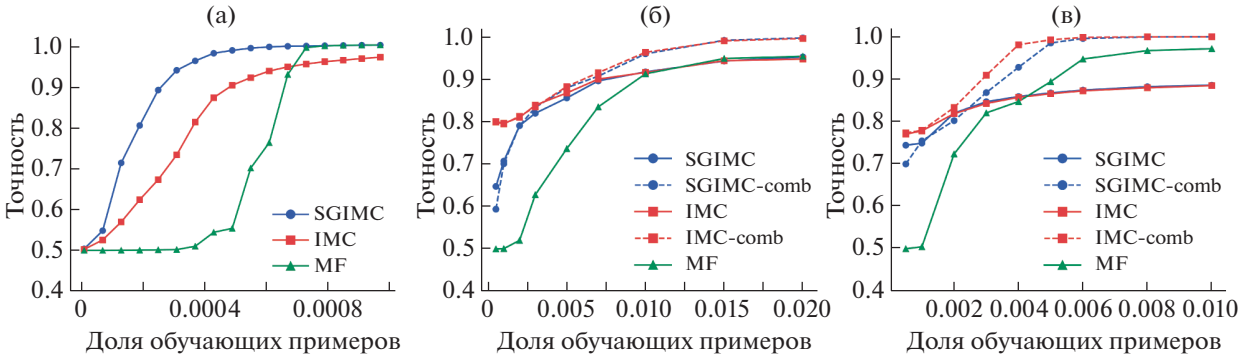
Предварительный анализ показывает, что для набора данных “Covtype” исходных побочных признаков недостаточно, чтобы IMC или SGIMC достигли точности выше 0.9, даже если ранг матрицы переоценен (фиг. 2в). Для того чтобы результаты IMC и SGIMC были сравнимы с результатами MF, к побочным признакам X были добавлены колонки диагональной единичной матрицы, что эквивалентно введению в модель фиктивных переменных (dummy variable), отражающих каждый отдельный объект. Это обогащение побочных признаков переводит методы индуктивного восстановления матриц в область методов трансдуктивного восстановления [14]. На фиг. 2б и 2в результаты обогащения побочных признаков обозначены через *SGIMC-comb* и *IMC-comb*.

В табл. 2 приведены результаты эксперимента на полуискусственных данных, полученных добавлением *умышленно* малоинформативных данных к побочным признакам набора “Segment”. Точность на тестовой подвыборке явно показывает необходимость отбора и исключения шумных и неинформативных признаков: точность восстановления процедуры IMC [31], которая не производит отбор признаков, ниже чем восстановление с групповым регуляризатором SGIMC.

5.2.2. Резистентность и восприимчивость бактерии *M. tuberculosis*. Данный набор является объединением наборов из работ [35]–[38]. Он состоит из реакций 4734 штаммов возбудителей ту-

Таблица 1. Общие характеристики наборов данных для кластеризации с примерами

Количество	“Mushrooms”	“Segment”	“Covtype”
наблюдений n	8124	2319	7370
признаков d	112	18	54
классов K	2	7	7



Фиг. 2. Точность восстановления в задаче кластеризации с примерами на разных наборах данных: а – “Mushrooms”, б – “Segment”, в – “Covtype”.

беркулеза *Mycobacterium tuberculosis* на 13 существующих антитуберкулезных препаратов. Задача состоит в том, чтобы предсказать реакцию штамма на каждый препарат (резистентность против восприимчивости). Для каждого штамма имеется сопутствующий ему вектор бинарных признаков длины 355709, отвечающих за наличие или отсутствие определенной мутации в геноме бактерии. При этом к каждому препарату также прилагается набор из 28 бинарных признаков, определяющих наличие специфических химических свойств у препарата. Для задач данного рода крайне важно, чтобы модель была интерпретируемой, что означает необходимость отбора нерелевантных свойств.

Поскольку число восприимчивых штаммов значительно превышает число штаммов, имеющих резистентность, качество восстановления измеряется с помощью метрики F_1, рассчитываемой на тестовой части набора. Значения метрики для каждого антибиотика по отдельности и всех в совокупности, представленные в табл. 3, были получены усреднением по десяти случайным разбиениям данных на обучающую и тестовую подвыборки в соотношении 1 : 1.

Результаты проведенного эксперимента позволяют заключить, что SGIMC может классифицировать реакцию штаммов на большинство препаратов лучше по F_1, чем IMC. При этом IMC

Таблица 2. Точность классификации для набора “Segment”

Дополнительные признаки	SGIMC	IMC
0	0.901 ± 0.003	0.895 ± 0.007
50	0.885 ± 0.003	0.839 ± 0.011
100	0.880 ± 0.006	0.822 ± 0.006
200	0.869 ± 0.007	0.795 ± 0.005
300	0.871 ± 0.019	0.769 ± 0.006
400	0.851 ± 0.014	0.754 ± 0.007

Таблица 3. Метрика F_1 на наборе данных *M. tuberculosis*

Препарат	SGIMC	IMC	Препарат	SGIMC	IMC
Все препараты	**0.59**	0.57	Capreomycin	**0.34**	0.28
Isoniazid	**0.89**	0.86	Amikacin	**0.47**	0.42
Ethambutol	**0.62**	0.61	Moxifloxacin	**0.45**	0.38
Rifampicin	**0.89**	0.88	Kanamycin	0.40	0.40
Pyrazinamide	0.53	0.53	Prothionamide	0.52	0.52
Streptomycin	0.84	**0.85**	Ciprofloxacin	0.52	**0.67**
Ofloxacin	**0.48**	0.42	Ethionamide	**0.50**	0.47

использует каждый из 355709 побочных признаков штаммов, что не позволяет получить осмысленную интерпретацию резистентности *M. tuberculosis* к препаратам, в то время как SGIMC отбирает всего ≈6000 из них и достигает сопоставимых показателей качества.

6. ЗАКЛЮЧЕНИЕ

В данной работе предлагается новый подход к индуктивному восстановлению матриц, который использует прореживающие регуляризаторы для отбора побочных признаков SGIMC. Эксперименты демонстрируют, что с помощью нового метода можно достичь высокого качества восстановления матриц как на искусственных наборах данных, так и на данных из прикладных задач. Более того, в условиях наличия большого числа малоинформативных побочных признаков метод и предложенная вычислительная процедура работают лучше, чем аналоги. Теоретический анализ показывает, что регуляризатор $L_{1,2}$, наводящий групповое прореживание, позволяет улучшить асимптотическую верхнюю оценку ошибки восстановления матрицы.

Дальнейшее развитие метода индуктивного восстановления матриц с отбором признаков через групповую регуляризацию SGIMC предлагается совершать в направлении поиска иных патернов разреженности, нежели чем построчно или по столбцам, которые естественно ожидать, например, в задачах классификации с многими метками. Дальнейшее теоретическое развитие метода может продолжаться в двух потенциальных направлениях. Первое касается разработки процедуры, имеющей глобальные гарантии сходимости, подобные [18], но работающие в условиях функции потерь более общего вида, нежели чем квадратичная, и с негладкими выпуклыми регуляризаторами, которые наводят разреженность и отбирают признаки. Второе направление затрагивает вопрос обобщения результата работы [21] в сторону постановки задачи индуктивного восстановления матриц общего плана для того, чтобы получить общую минимакс оценку границ ошибки восстановления.

СПИСОК ЛИТЕРАТУРЫ

1. *Rennie J.D.M., Srebro N.* Fast maximum margin matrix factorization for collaborative prediction // Proc. of the 22nd Internat. Conference on Machine Learning. 2005. P. 713–719.
2. *Koren Y., Bell R., Volinsky C.* Matrix factorization techniques for recommender systems // Computer. 2009. V. 42. № 8. P. 30–37.
3. *Yi J., Yang T., Jin R., Jain A.K., Mahdavi M.* Robust ensemble clustering by matrix completion // 2012 IEEE 12th Internat. Conference on Data Mining (ICDM). 2012. P. 1176–1181.
4. *Argyriou A., Evgeniou T., Pontil M.* Convex multi-task feature learning // Machine Learning. 2008. V. 73. № 3. P. 243–272.
5. *Cabral R.S., Torre F., Costeira J.P., Bernardino A.* Matrix completion for multi-label image classification // Advances in Neural Information Proc. Systems. 2011. P. 190–198.
6. *Weng Z., Wang X.* Low-rank matrix completion for array signal processing // 2012 IEEE Intern. Conference on Acoustics, Speech and Signal Proc. (ICASSP). 2012. P. 2697–2700.
7. *Chen P., Suter D.* Recovering the missing components in a large noisy low-rank matrix: Application to SFM // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004. V. 26. № 8. P. 1051–1063.
8. *Candès E.J., Recht B.* Exact matrix completion via convex optimization // Foundations of Comput. Math. 2009. V. 9. № 6. P. 717–772.
9. *Candès E.J., Tao T.* The power of convex relaxation: Near–optimal matrix completion // IEEE Transactions on Information Theory. 2010. V. 56. № 5. P. 2053–2080.
10. *Shamir O., Shalev-Shwartz S.* Matrix completion with the trace norm: learning, bounding, and transducing // J. of Machine Learning Research. 2014. V. 15. № 1. P. 3401–3423.
11. *Hannon J., Bennett M., Smyth B.* Recommending twitter users to follow using content and collaborative filtering approaches // Proc. of the Fourth ACM Conference on Recommender Systems. 2010. P. 199–206.
12. *Xu M., Jin R., Zhou Z.-H.* Speedup matrix completion with side information: Application to multi-label learning // Advances in Neural Information Proc. Systems. 2013. P. 2301–2309.
13. *Natarajan N., Dhillon I.S.* Inductive matrix completion for predicting gene-disease associations // Bioinformatics. 2014. V. 30. № 12. P. i60–i68.
14. *Chiang K.-Y., Hsieh C.-J., Dhillon I.S.* Matrix completion with Noisy side information // Proc. of the 28th Internat. Conference on Neural Information Proc. Systems – Vol. 2. 2015. P. 3447–3455.
15. *Si S., Chiang K.-Y., Hsieh C.-J., Rao N., Dhillon I.S.* Goal-directed inductive matrix completion // Proc. of the 22nd ACM SIGKDD Intern. Conference on Knowledge Discovery and Data Mining. 2016. P. 1165–1174.

16. *Lu J., Liang G., Sun J., Bi J.* A sparse interactive model for matrix completion with side information // Advances in Neural Information Proc. Systems. 2016. P. 4071–4079.

17. *Guo Y.* Convex Co-Embedding for Matrix Completion with Predictive Side Information // AAAI. 2017. P. 1955–1961.

18. *Zhang X., Du S., Gu Q.* Fast and sample efficient inductive matrix completion via multi-phase procrustes flow // Proc. of the 35th International Conference on Machine Learning. 2018. P. 5756–5765.

19. *Soni A., Chevalier T., Jain S.* Noisy inductive matrix completion under sparse factor models // 2017 IEEE Intern. Symposium on Information Theory (ISIT). 2017. P. 2990–2994.

20. *Jain P., Netrapalli P., Sanghavi S.* Low-rank matrix completion using alternating minimization // Proc. of the Forty-fifth Annual ACM Symposium on Theory of Computing. 2013. P. 665–674.

21. *Berthet Q., Baldin N.* Statistical and computational rates in graph logistic regression // Intern. Conference on Artificial Intelligence and Statistics. 2020. P. 2719–2730.

22. *Bertsekas D.P., Tsitsiklis J.N.* Parallel and distributed computation: numerical methods. US: Prentice hall Englewood Cliffs, 1989.

23. *Boyd S., Parikh N., Chu E., Peleato B., Eckstein J.* Distributed optimization and statistical learning via the alternating direction method of multipliers // Foundations and Trends in Machine Learning. 2011. V. 3. № 1. P. 1–122.

24. *Maurer A., Pontil M.* Structured sparsity and generalization // J. of Machine Learning Research. 2012. V. 13. P. 671–690.

25. *Bartlett P.L., Mendelson S.* Rademacher and Gaussian complexities: Risk bounds and structural results // J. of Machine Learning Research. 2002. V. 3. P. 463–482.

26. *Mohri M., Rostamizadeh A., Talwalkar A.* Foundations of Machine Learning. US: The MIT Press, 2012.

27. *Glowinski R., Marroco A.* Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation–dualité d'une classe de problémes de Dirichlet non linкires // ESAIM: Math. Model. and Numerical Analysis. 1975. V. 9. P. 41–76.

28. *Gabay D., Mercier B.* A dual algorithm for the solution of nonlinear variational problems via finite element approximation // Computers & Math. with Applications. 1976. V. 2. № 1. P. 17–40.

29. *Gabay D.* Chapter IX Applications of the Method of Multipliers to Variational Inequalities // Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary–Value Problems. 1983. P. 299–331.

30. *Eckstein J., Bertsekas D.P.* On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators // Math. Progr. 1992. V. 55. № 1. P. 293–318.

31. *Yu H.-F., Jain P., Kar P., Dhillon I.S.* Large-scale multi-label learning with missing labels // Proc. of the 31st Intern. Conference on Machine Learning. 2014. P. 593–601.

32. *Lin C.-J., Weng R.C., Keerthi S.S.* Trust region newton method for logistic regression // J. Mach. Learn. Res. 2008. V. 9. P. 627–650.

33. *Simon N., Friedman J., Hastie T., Tibshirani R.* A sparse-group Lasso // J. of Computational and Graphical Statistics. 2013. V. 22. № 2. P. 231–245.

34. *Chang C.-C., Lin C.-J.* LIBSVM: a library for support vector machines // ACM Transactions on Intelligent Systems and Technology (TIST). 2011. V. 2. № 3. P. 1–27.

35. *Farhat M.R., Shapiro B.J., Kieser K.J., et al.* Genomic analysis identifies targets of convergent positive selection in drug-resistant Mycobacterium tuberculosis // Nature Genetics. 2013. V. 45. P. 1183–1189.

36. *Walker T.M., Kohl T.A., Omar S.V., et al.* Whole-genome sequencing for prediction of Mycobacterium tuberculosis drug susceptibility and resistance: a retrospective cohort study // The Lancet Infectious Diseases. Appl. 2015. V. 15. № 10. P. 1193–1202.

37. *Pankhurst L.J., Elias C. del O., Votintseva A.A., et al.* Rapid, comprehensive, and affordable mycobacterial diagnosis with whole–genome sequencing: a prospective study // The Lancet Respiratory Medicine. 2016. V. 4. № 1. P. 49–58.

38. *Coll F., Phelan J., Hill-Cawthorne G.A., et al.* Genome-wide analysis of multi- and extensively drug-resistant Mycobacterium tuberculosis // Nature Genetics. 2018. V. 50. № 2. P. 307–316.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 759–775

ОБЩИЕ ЧИСЛЕННЫЕ МЕТОДЫ

УДК 519.61

ВЫЧИСЛЕНИЕ СОБСТВЕННЫХ ВЕКТОРОВ НЕСИММЕТРИЧНЫХ ТРЕХДИАГОНАЛЬНЫХ МАТРИЦ[1)]

© 2021 г. П. Ван Дорен[1],*, Т. Лаудадио[2],, Н. Мастронарди[2],*****

[1] *Department of Mathematical Engineering, Catholic University of Louvain, Louvain-la-Neuve, Belgium*

[2] *Istituto per le Applicazioni del Calcolo, Bari, Italy*

**e-mail: paul.vandooren@uclouvain.be*

***e-mail: t.laudadio@cnr.it*

****e-mail: n.mastronardi@cnr.it*

Поступила в редакцию 24.11.2020 г.
Переработанный вариант 24.11.2020 г.
Принята к публикации 14.01.2021 г.

Вычисление собственного разложения матриц является одной из наиболее изученных задач в вычислительной линейной алгебре. В частности, задачи нахождения собственных значений вещественных несимметричных трехдиагональных матриц возникают в самых разных приложениях. В этой статье рассматривается задача вычисления собственного вектора, соответствующего известному собственному значению вещественной несимметричной трехдиагональной матрицы, для чего разрабатывается алгоритм, комбинирующий итерации QR- и QL-алгоритмов со сдвигами, равными известному собственному значению. Численные эксперименты показывают надежность предложенного метода. Библ. 19. Фиг. 8. Табл. 2.

Ключевые слова: несимметричные трехдиагональные матрицы, собственные вектора, матрицы Бесселя.

DOI: 10.31857/S0044466921050082

1. ВВЕДЕНИЕ

Задачи нахождения собственных значений вещественных несимметричных трехдиагональных матриц возникают в самых разных приложениях. Например, несимметричная задача на собственные значения может быть сведена к несимметричной трехдиагональной форме за конечное число шагов [1]–[3]. В разреженном случае несимметричный алгоритм Ланцоша производит несимметричную трехдиагональную матрицу с помощью процесса биортогонализации Ланцоша [4]. Нули обобщенных полиномов Бесселя могут быть вычислены как собственные значения соответствующих матриц Бесселя, которые являются несимметричными трехдиагональными [5], [6]. Другие задачи на собственные значения, включая несимметричные трехдиагональные матрицы, могут быть найдены в [7].

Некоторые методы вычисления собственных значений несимметричных трехдиагональных матриц известны в литературе. Метод, описанный в [7], требует вычисления $p(\lambda)/p'(\lambda) = -1/\operatorname{tr}((T-\lambda I)^{-1})$, где $p(\lambda)$ является характеристическим полиномом матрицы T, что и делается с использованием QR-разложения $T-\lambda I$ и семисепарабельной структуры $(T-\lambda I)^{-1}$. Алгоритм, описанный в [8], основывается на LR-методе [9]. Неявный QR-метод [10], [11] также может быть использован для вычисления собственной структуры трехдиагональной матрицы. Но, к сожалению, он не использует трехдиагональную структуру задачи, и, таким образом, сложность кубически зависит от размера матрицы. Эта работа фокусируется на вычислении собственных векторов трехдиагональных матриц в случае, когда соответствующие собственные значения известны.

Хорошо известно, что в точной арифметике после одной итерации QR (QL)-метода со сдвигом, равным собственному значению, это собственное значение возникает в нижнем правом

[1)]Работа Т. Лаудадио и Н. Мастронарди выполнена при частичной финансовой поддержке GNCS–INdAM. Работа П. Ван Дорена выполнена при частичной финансовой поддержке CNR, Italy в рамках Short Term Mobility Program.

(верхнем левом) углу матрицы и может быть отброшено [12]. К сожалению, оба метода могут страдать от прямой неустойчивости при работе в арифметике конечной точности [13], [14]. Для того, чтобы избежать явления неустойчивости, мы комбинируем итерации QR- и QL-методов со сдвигами, равными собственному значению λ.

В этой работе мы фокусируемся на вычислении левого собственного вектора, ассоциированного с известным собственным значением несимметричной трехдиагональной матрицы. Для краткости мы опускаем описание алгоритма вычисления правого собственного вектора, поскольку эта задача полностью аналогична. Численные примеры показывают надежность предложенного подхода.

Текст организован следующим образом. В разд. 2 приведены обозначения и базовые определения. В разд. 3 выделены основные особенности QR-метода. Предложенный алгоритм описан в разд. 4. В разд. 5 показано, как избежать комплексной арифметики в случае комплексно-сопряженных собственных значений. В разд. 6 мы приводим ряд численных примеров и завершаем статью разделом с заключительными замечаниями.

2. ОБОЗНАЧЕНИЯ И ОПРЕДЕЛЕНИЯ

Матрицы обозначены с помощью заглавных букв, а их элементы с помощью строчных, т.е. элемент (i, j) матрицы T обозначен как $t_{i,j}$.

Подматрица матрицы B, взятая на строках $i, i+1, i+2, \ldots, i+k$, где $1 \le i \le i+k \le n$, и столбцах $j, j+1, j+2, \ldots, j+\ell$, где $1 \le j \le j+\ell \le n$, обозначена через $B_{i:i+k,\,j:j+\ell}$.

Единичная матрица порядка n обозначена через I_n или I, если это не вызывает неоднозначности. Матрица $T - \varkappa I$, где $\varkappa \in \mathbb{R}$, обозначена через $T(\varkappa)$.

Главная диагональ матрицы $B \in \mathbb{R}^{m\times n}$ обозначена через $\operatorname{diag}(B)$, а i-й вектор канонического базиса $\mathbb{R}^n$ обозначен через $\mathbf{e}_i^{(n)}$ или просто $\mathbf{e}_i$, если это не вызывает неоднозначности. Машинная точность обозначена через ε_M.

Определение 1. Столбцы $B \in \mathbb{R}^{m\times n}$, $m \ge n$, называются ε-линейно зависимыми, если $\sigma_n(B) \le \varepsilon \|B\|_2$, где $\sigma_n(B)$ есть n-е сингулярное число B, а ε – произвольная точность.

3. ВЫЧИСЛЕНИЕ СОБСТВЕННОГО ВЕКТОРА С ПОМОЩЬЮ QR И QL-МЕТОДОВ

Пусть $T \in \mathbb{R}^{n\times n}$ – неприводимая несимметричная трехдиагональная матрица

$$T = \begin{bmatrix} \alpha_1 & \gamma_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{bmatrix},$$

т.е. $\beta_i \ne 0$, $\gamma_i \ne 0$, $i = 1, \ldots, n-1$, и пусть λ_i $(i = 1, \ldots, n)$ – ее собственные значения. Поскольку T – неприводимая, геометрическая кратность ее собственных значений равна 1.

Сначала мы опишем, как левый собственный вектор $\mathbf{y}$, соответствующий собственному значению λ матрицы T, может быть получен применением одной итерации QR- или QL-методов со сдвигом λ. К сожалению, оба метода могут страдать от прямой неустойчивости при работе в арифметике конечной точности. Далее мы анализируем причины такого поведения, а затем предлагаем алгоритм, основанный на подходящей комбинации вышеупомянутых методов, для преодоления явления неустойчивости.

3.1. QR-метод

Пусть $\varkappa \in \mathbb{C}$. Пусть $\hat{Q}\hat{R}$ является QR-разложением $T(\varkappa)$, где

$$\hat{R} = \begin{bmatrix} \hat{\rho}_1 & \hat{\gamma}_1 & \hat{\beta}_1 & & \\ & \hat{\rho}_2 & \hat{\gamma}_2 & \ddots & \\ & & \ddots & \ddots & \hat{\beta}_{n-2} \\ & & & \hat{\rho}_{n-1} & \hat{\gamma}_{n-1} \\ & & & & \hat{\rho}_n \end{bmatrix},$$

и $\hat{Q} = \hat{G}_{n-1}^H \hat{G}_{n-2}^H \cdots \hat{G}_1^H$, где $\hat{G}_i$ – вращения Гивенса:

$$\hat{G}_i = \begin{bmatrix} I_{i-1} & & & \\ & c_i & s_i & \\ & -\overline{s}_i & c_i & \\ & & & I_{n-i-1} \end{bmatrix}, \quad c_i^2 + |s_i|^2 = 1, \quad i = 1, \ldots, n-1.$$

Коэффициенты c_i, $i = 1, \ldots, n-1$, вращений Гивенса вещественны [12, с. 243]. Следовательно, $\hat{Q}$ является унитарной верхней хессенберговой матрицей:

$$\hat{Q} = \begin{bmatrix} c_1 & -s_1c_2 & s_1s_2c_3 & \ddots & (-1)^{n-2}c_{n-1}\prod\limits_{i=1}^{n-2} s_i & (-1)^{n-1}\prod\limits_{i=1}^{n-1} s_i \\ \overline{s}_1 & c_1c_2 & -c_1s_2c_3 & \ddots & (-1)^{n-3}c_1c_{n-1}\prod\limits_{i=2}^{n-2} s_i & (-1)^{n-2}c_1\prod\limits_{i=2}^{n-1} s_i \\ & \overline{s}_2 & c_2c_3 & \ddots & \vdots & \vdots \\ & & \ddots & \ddots & -c_{n-3}s_{n-2}c_{n-1} & c_{n-3}s_{n-2}s_{n-1} \\ & & & \overline{s}_{n-2} & c_{n-2}c_{n-1} & -c_{n-2}s_{n-1} \\ & & & & \overline{s}_{n-1} & c_{n-1} \end{bmatrix}.$$

В точной арифметике, если $\varkappa \equiv \lambda_i$, $i \in \{1, \ldots, n\}$, то оказывается, что $\hat{\rho}_n = 0$ и

$$\mathbf{y}_i \equiv \hat{Q}(:, n) \tag{1}$$

является левым собственным вектором, соответствующим $\varkappa$.

Как было отмечено выше, в арифметике конечной точности может возникнуть прямая неустойчивость при QR-разложении $T(\lambda_i)$, и последний столбец $\hat{Q}$ может оказаться далек от искомого собственного вектора [13], [15].

В частности, если определить $\hat{R}^{(0)}(\lambda_i) = T - \lambda_i I_n$, $\hat{R}^{(i)}(\lambda_i) = \hat{G}_i \hat{R}^{(i-1)}(\varkappa)$, $i = 1, \ldots, n-1$, прямая неустойчивость возникнет на j-м шаге QR-факторизации тогда и только тогда, когда $\hat{R}^{(j)}_{1:j+1,1:j+1}(\varkappa)$ близка к вырожденной и c_j пренебрежимо мало. Это явление уже было описано в [13] для симметричного случая.

3.2. QL-метод

Пусть $\tilde{Q}\tilde{L} = T - \varkappa I_n$ является QL-разложением $T(\varkappa)$, полученным применением последовательности вращений Гивенса:

$$\tilde{G}_i = \begin{bmatrix} I_{n-i-1} & & & \\ & g_i & h_i & \\ & -\overline{h}_i & g_i & \\ & & & I_{i-1} \end{bmatrix}, \quad g_i^2 + |h_i|^2 = 1, \quad i = 1, \ldots, n-1.$$

Оказывается, что

$$\tilde{L} = \begin{bmatrix} \tilde{\rho}_1 & & & \\ \tilde{\gamma}_1 & \tilde{\rho}_2 & & \\ \tilde{\beta}_1 & \tilde{\gamma}_2 & \ddots & \\ & \ddots & \ddots & \tilde{\rho}_{n-1} \\ & & \tilde{\beta}_{n-2} & \tilde{\gamma}_{n-1} \quad \tilde{\rho}_n \end{bmatrix}$$

и $\tilde{Q} = \tilde{G}_{n-1}^H \tilde{G} H_{n-2} \cdots \tilde{G}_1^H$ является нижней хессенберговой матрицей:

$$\tilde{Q} = \begin{bmatrix} g_{n-1} & \overline{h}_{n-1} & & & & \\ -g_{n-2}h_{n-1} & g_{n-2}g_{n-1} & \overline{h}_{n-2} & & & \\ g_{n-3}h_{n-2}h_{n-1} & -g_{n-3}h_{n-2}g_{n-1} & \ddots & \ddots & & \\ \vdots & \vdots & \ddots & g_3 g_2 & \overline{h}_2 & \\ (-1)^{n-2} g_1 \prod_{i=2}^{n-1} h_i & (-1)^{n-3} g_1 g_{n-1} \prod_{i=2}^{n-2} h_i & \ddots & -g_3 h_2 g_1 & g_2 g_1 & \overline{h}_1 \\ (-1)^{n-1} \prod_{i=1}^{n-1} h_i & (-1)^{n-2} g_{n-1} \prod_{i=1}^{n-2} h_i & \ddots & g_3 h_2 h_1 & -g_2 h_1 & g_1 \end{bmatrix}.$$

В точной арифметике, если $\varkappa \equiv \lambda_i$, $i \in \{1, \ldots, n\}$, то $\tilde{\rho}_1 = 0$ и

$$\mathbf{y}_i \equiv \tilde{Q}(:, 1) \tag{2}$$

является левым собственным вектором, соответствующим λ_i.

В этом случае также может возникнуть прямая неустойчивость, и первый столбец $\tilde{Q}$ может быть отличным от искомого собственного вектора.

В частности, если определить $\tilde{L}^{(0)}(\lambda_i) = T(\lambda_i)$, $\tilde{L}^{(i)}(\lambda_i) = \tilde{G}_i \tilde{L}^{(i-1)}(\lambda_i)$, $i = 1, \ldots, n-1$, прямая неустойчивость возникнет на j-м шаге при QL-разложении тогда и только тогда, когда $\tilde{L}^{(j)}_{n-j:n,n-j:n}(\varkappa)$ очень близка к вырожденной, и первый элемент собственного вектора, соответствующего наименьшему собственному значению, мал [13]. Из уравнения (2) этот элемент задается g_j. Таким образом, прямая неустойчивость возникает, если $\tilde{L}(j)_{n-j:n,n-j:n}(\lambda_i)$ близка к вырожденной и g_j пренебрежимо мал.

Чтобы выяснить, на каком шаге QR- и QL-методов возникает прямая неустойчивость, рассмотрим.

Следствие 1 (см. [16, с. 149]). Пусть $A \in \mathbb{C}^{m \times n}$ и пусть B является ее подматрицей, полученной удалением r столбцов из A. Тогда

$$\sigma_k(A) \geq \sigma_k(B) \geq \sigma_{k+r}(A), \quad k = 1, \ldots, \min\{m, n\}, \tag{3}$$

где $\sigma_\ell(A) \equiv 0$ при $\ell > \min\{m, n\}$.

Пусть λ – собственное значение T. Обозначим сингулярные числа $\hat{R}^{(i)}_{1:i+1,:}(\lambda)$ как $\sigma_j(\hat{R}^{(i)}_{1:i+1,:}(\lambda))$, $j = 1, \ldots, i+1$, а сингулярные числа $\tilde{L}^{(i)}_{n-i:n,:}(\lambda)$ как $\sigma_j(\tilde{L}^{(i)}_{n-i:n,:}(\lambda))$, $j = 1, \ldots, i+1$, при $i = 1, \ldots, n-1$ соответственно. Из следствия 1 имеем

$$\sigma_{i+1}(\hat{R}^{(i)}_{1:i+1,:}(\lambda)) \geq \sigma_{i+2}(\hat{R}^{(i+1)}_{1:i+2,:}(\lambda)),$$
$$\sigma_{i+1}(\tilde{L}^{(i)}_{n-i:n,:}(\lambda)) \geq \sigma_{i+2}(\tilde{L}^{(i+1)}_{n-i-1:n,:}(\lambda)), \quad i = 1, \ldots, n-2.$$

Предположим, что $\sigma_{n-1}(T(\lambda)) \gg \varepsilon > \sigma_n(T(\lambda)) = 0$.

Следующая лемма показывает, что матрицы $\hat{R}^{(j)}_{1:j+1,:}(\lambda)$ и $\tilde{L}^{(n-j-2)}_{j+2:n,:}(\lambda)$ не могут одновременно иметь сингулярные числа меньше ε.

Лемма 1. *Пусть* $k_1, k_2 \in \{1, \ldots, n\}$, $k_1 \neq k_2$, *и*

$$\Delta = T(\lambda)[\mathbf{e}_{k_1} \ \ \mathbf{e}_{k_2}].$$

Тогда

$$\|\Delta\|_F \geq \|\Delta\|_2 \geq \sigma_{n-1}(T(\lambda)), \tag{4}$$

где $\sigma_{n-1}(T(\lambda))$ *является* $(n-1)$*-м сингулярным числом* $T(\lambda)$.

Доказательство. Из (3) мы имеем, что

$$\sigma_1(\Delta) = \|\Delta\|_2 \geq \sigma_{n-1}(T(\lambda)),$$

где $\sigma_1(\Delta)$ – максимальное сингулярное число Δ, откуда следует (4).

Рассмотрим для каждого j факторизацию $T(\lambda)$:

$$T(\lambda) = \left[\begin{array}{c|c} \hat{Q}^{(j)} & \\ \hline & \tilde{Q}^{(n-j-2)} \end{array}\right] \left[\begin{array}{c} \hat{R}^{(j)}_{1:j+1,:}(\lambda) \\ \hline \tilde{L}^{(n-j-2)}_{j+2:n,:}(\lambda) \end{array}\right],$$

где $\hat{Q}^{(j)} = \hat{G}j^H \hat{G}^H_{j-1} \cdots \hat{G}^H_1$ и $\tilde{Q}^{(n-j-2)} = \tilde{G}^H_{n-j-2}\tilde{G}^H_{n-j-3} \cdots \tilde{G}^H_2\tilde{G}^H_1$. Из леммы 1 следует, что только одна из 2 подматриц может быть ε-зависима при $\varepsilon \leq \sigma_{n-1}(T(\lambda))$.

Обозначим $\hat{\sigma}_i = \sigma_i(\hat{R}^{(i-1)}_{1:i,:}(\lambda))$ и $\tilde{\sigma}_i = \sigma_{n-i+1}(L^{(n-i)}_{i:n,:}(\lambda))$, $i = 1, \ldots, n$.

Лемма 2. *Последовательность*

$$\{\max(\hat{\sigma}_i, \tilde{\sigma}_i)\}_{i=1}^n = \{\hat{\sigma}_1, \ldots, \hat{\sigma}_{j-1}, \max(\hat{\sigma}_j, \tilde{\sigma}_j), \tilde{\sigma}_{j+1}, \ldots, \tilde{\sigma}_n\}$$

унимодальна и имеет минимальное значение, равное $\max(\hat{\sigma}_j, \tilde{\sigma}_j)$ *при некотором* $j \in \{1, \ldots, n\}$. *Более того, только один элемент последовательности может быть меньше* $\sigma_{n-1}(T(\lambda))$.

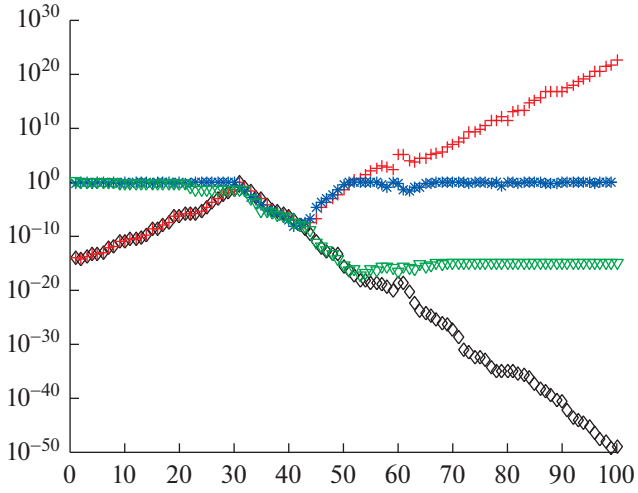
Тогда из анализа возмущений при QR-разложении следует [17], что прямая неустойчивость при вычислении $\hat{Q}$ и, следовательно, косинусов c_i не возникает на первых $j-1$ шагах QR-разложения, а прямая неустойчивость при вычислении $\tilde{Q}$ не возникает на первых $n-j-1$ шагах QL-факторизации, и, следовательно, для косинусов g_i. Для того, чтобы выбрать подходящий индекс j, можно воспользоваться последовательностями $\{c_j\}_{j=1}^{n-1}$ и $\{g_j\}_{j=1}^{n-1}$, как показано в следующих примерах.

Пример 1. Пусть $T \in \mathbb{R}^{n\times n}$, $n = 100$, является несимметричной неприводимой трехдиагональной матрицей, чьи элементы порождены функцией `randn` из `MATLAB`, и пусть λ является собственным значением T. Пусть $(\lambda, \mathbf{y})$ является левой собственной парой T, полученной применением нескольких шагов метода обратных итераций к соответствующей паре, полученной функцией `eig` из `MATLAB`. Заметим, что для таких матриц поведение, которое мы собираемся описать, наблюдается для всех собственных значений.

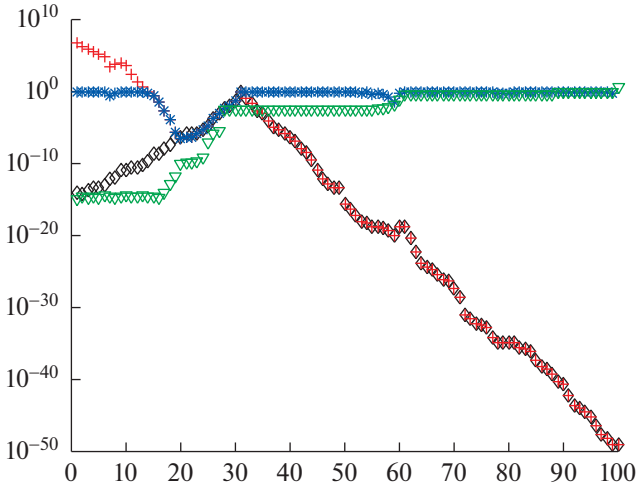
Последовательность косинусов $\{c_i\}_{i=1}^{n-1}$, порожденная QR-методом со сдвигом, равным λ (обозначено "$*$"), последовательность $\{\hat{\sigma}_i\}_{i=1}^n$ (обозначено "$\triangledown$"), модули элементов левого собственного вектора $\mathbf{y}$, соответствующего λ (обозначено "$\diamond$"), и модули элементов последнего столбца $\hat{Q}$ (обозначено "+") изображены на фиг. 1 в логарифмическом масштабе.

Последовательность косинусов $\{g_i\}_{i=1}^{n-1}$, порожденная QL-методом со сдвигом, равным λ (обозначено "$*$"), последовательность $\{\tilde{\sigma}_i\}_{i=1}^n$ (обозначено "$\triangledown$"), модули элементов левого собственного вектора $\mathbf{y}$, соответствующего λ (обозначено "$\diamond$"), и модули элементов первого столбца $\tilde{Q}$ (обозначено "+") изображены на фиг. 2 в логарифмическом масштабе.

На фиг. 1 можно видеть, что элементы последовательности $\{c_i\}_{i=1}^{n-1}$ имеют поведение, похожее на элементы последовательности $\{\hat{\sigma}_i\}_{i=1}^n$ до тех пор, пока значение больше $\sqrt{\varepsilon_M}$. Затем возникает прямая неустойчивость, и оставшаяся часть элементов первой последовательности расходится с элементами второй. Аналогично, модули элементов последнего столбца $\hat{Q}$ соответствуют элементам $\mathbf{y}$ до тех пор, пока не возникает прямая неустойчивость.



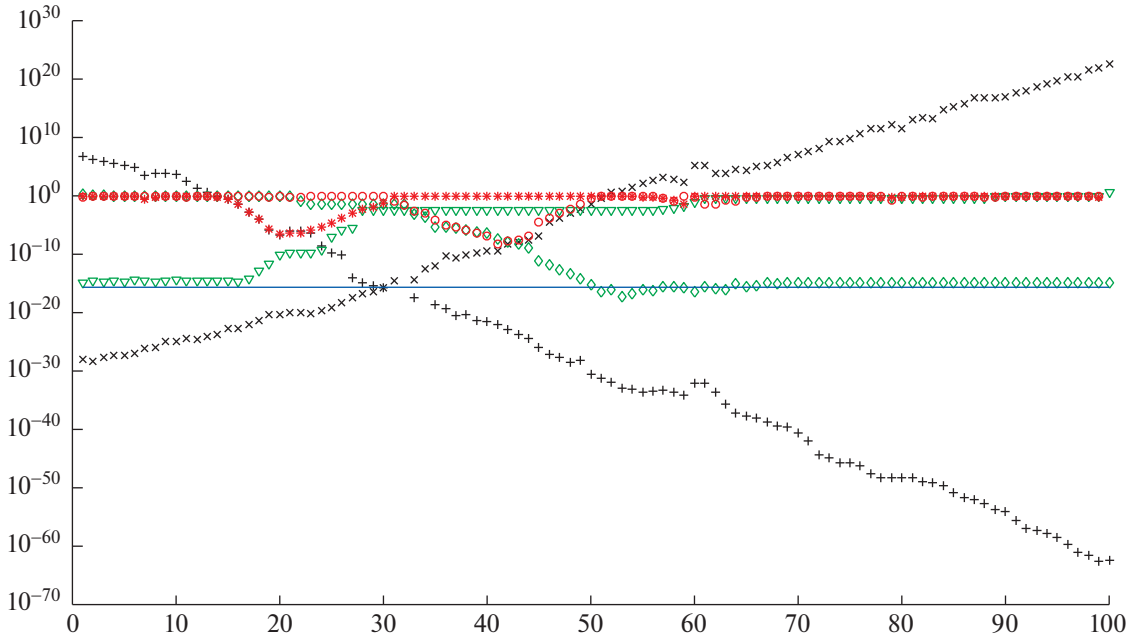
Фиг. 1. Пример 1: график $\{c_i\}_{i=1}^{n-1}$ (“$*$”), $\{\hat{\sigma}_i\}_{i=1}^{n}$ (“$\triangledown$”), модули элементов левого собственного вектора $\mathbf{y}$, соответствующего λ (“$\diamond$”), и модули элементов последнего столбца $\hat{Q}$ (“+”), вычисленные на одной итерации QR-метода, примененного к $T(\lambda)$.



Фиг. 2. Пример 1: график $\{g_i\}_{i=1}^{n-1}$ (“$*$”), $\{\tilde{\sigma}_i\}_{i=1}^{n}$ (“$\triangledown$”), модули элементов левого собственного вектора $\mathbf{y}$, соответствующего λ (“$\diamond$”), и модули элементов первого столбца $\tilde{Q}$ (“+”), вычисленные на одной итерации QL-метода, примененного к $T(\lambda)$.

С другой стороны, фиг. 2 показывает, что последние элементы последовательностей $\{g_i\}_{i=1}^{n-1}$ и $\{\tilde{\sigma}_i\}_{i=1}^{n}$ имеют похожее поведение до тех пор, пока элементы первой последовательности больше $\sqrt{\varepsilon_M}$. Это явление также описано в [13] для симметричных трехдиагональных матриц. Элементы $\mathbf{y}$ и первый столбец $\tilde{Q}$ также имеют похожее поведение.

Покомпонентные ошибки $\left|\mathbf{y} - \hat{Q}(:,n)\right|$ (обозначено “$\times$”) и $\left|\mathbf{y} - \tilde{Q}(:,1)\right|$ (обозначено “+”), последовательности $\{\hat{\sigma}_i\}_{i=1}^{n}$ (обозначено “$\diamond$”), $\{\tilde{\sigma}_i\}_{i=1}^{n}$ (обозначено “$\circ$”), $\{c_i\}_{i=1}^{n-1}$ (обозначено “$\circ$”) и $\{g_i\}_{i=1}^{n-1}$ (“$*$”), и машинная точность ε_M (обозначено синей линией) изображены в логарифмическом масштабе на фиг. 3.



Фиг. 3. Пример 1: график $\left|\mathbf{y} - \hat{Q}(:,n)\right|$ (“×”), $\left|\mathbf{y} - \tilde{Q}(:,1)\right|$ (“+”), $\{\hat{\sigma}_i\}_{i=1}^{n}$ (“◇”), $\{\tilde{\sigma}_i\}_{i=1}^{n}$ (“▽”), $\{c_i\}_{i=1}^{n-1}$ (“○”) и $\{g_i\}_{i=1}^{n-1}$ (“∗”), машинной точности ε_M (синия линия) в логарифмическом масштабе.

Можно видеть, что покомпонентные ошибки $\left|\mathbf{y} - \hat{Q}(:,1)\right|$ всегда меньше ε_M до 30 позиции, в то время как $\left|\mathbf{y} - \tilde{Q}(:,1)\right|$ всегда меньше ε_M после 30 позиции, как и предсказано леммой 1.

Следовательно, для всех λ_i существует индекс j такой, что

$$\left|\mathbf{y}_k - \hat{Q}(k,1)\right| \le v \cdot \varepsilon_M, \quad 1 \le k \le j,$$
$$\left|\mathbf{y}_k - \tilde{Q}(k,1)\right| \le v \cdot \varepsilon_M, \quad j+1 \le k \le n,$$

где $v \sim O(n)$.

Подводя итог, элементы последнего столбца $\hat{Q}$ вычисляются точно в верхней части до тех пор, пока не возникает прямая неустойчивость на QR-итерации, т.е. до тех пор, пока не появится индекс $j \in \{1, \ldots, n\}$ такой, что $T_{1:j,:}$ имеет почти линейно зависимые столбцы. С другой стороны, прямая неустойчивость во время QL-итерации может возникнуть только после $n - j - 1$ шагов, и элементы первого столбца $\tilde{Q}$ вычисляются точно в нижней части. Таким образом, важно правильно определить индекс j.

Замечание 1. Заметим, что в то время как последовательности $\{\hat{\sigma}_i\}_{i=1}^{n}$ и $\{\tilde{\sigma}_i\}_{i=1}^{n}$ монотонны, последовательности $\{c_i\}_{i=1}^{n-1}$ и $\{g_i\}_{i=1}^{n-1}$ монотонными не являются. Однако в следующем разделе мы покажем, как найти подходящий индекс j по элементам этих последовательностей.

4. КОМБИНИРОВАНИЕ QR И QL-МЕТОДОВ ДЛЯ ВЫЧИСЛЕНИЯ СОБСТВЕННОГО ВЕКТОРА

В этом разделе описан метод нахождения индекса j, используемого для построения искомого собственного вектора при помощи вычисления первых j элементов последнего столбца $\hat{Q}$ и последних $n - j$ элементов первой строки $\tilde{Q}$.

Пусть $\{\hat{c}_i\}_{i=1}^n$ и $\{\tilde{g}_i\}_{i=1}^n$ являются, соответственно, последовательностями коэффициентов вращений Гивенса, необходимых для вычисления QR и QL-разложений матрицы $T(\lambda)$ в точной арифметике.

Если $\sigma_{n-1}(T(\lambda)) \gg \sigma_n(T(\lambda))$ и прямая неустойчивость возникла на j-м шаге QR-метода со сдвигом λ, то последовательность $\{c_i\}_{i=1}^n$ начинает расходиться с последовательностью $\{\hat{c}_i\}_{i=1}^n$ в окрестности индекса j. Как следствие, последний столбец $\hat{Q}$ соответствует $\mathbf{y}$ в первых j элементах.

Аналогично, последовательность $\{g_i\}_{i=1}^n$ начинает расходиться с последовательностью $\{\tilde{g}_i\}_{i=1}^n$ в окрестности индекса $n-j$, и последние $n-j$ элементов первого столбца $\tilde{Q}$ совпадают с соответствующими элементами $\mathbf{y}$.

Следовательно, искомый индекс j может быть вычислен следующим образом.

Положим $i_1 = 1$, $i_2 = n$, $R = T(\lambda)$, $L = T(\lambda)$ и вычислим вращения Гивенса

$$\hat{G}_{i_1} = \begin{bmatrix} I_{i_1-1} & & & \\ & c_{i_1} & s_{i_1} & \\ & -s_{i_1} & c_{i_1} & \\ & & & I_{n-i_1-1} \end{bmatrix} \quad \text{и} \quad \tilde{G}_{i_2-1} = \begin{bmatrix} I_{i_2-2} & & & \\ & g_{i_2} & h_{i_2} & \\ & -h_{i_2} & g_{i_2} & \\ & & & I_{n-i_2} \end{bmatrix},$$

участвующие в соответствующих шагах QR и QL-методов со сдвигом λ.

Если $c_{i_1} \geq g_{i_2-1}$, то $\hat{G}_{i_1}$ применяется к R : $R \leftarrow \hat{G}_{i_1} R$ и положим $i_1 = i_1 + 1$, в противном случае $\tilde{G}_{i_2}$ применяется к L : $L \leftarrow \tilde{G}_{i_2} L$ и положим $i_2 = i_2 - 1$. Затем повторяем процедуру до тех пор, пока $i_1 > i_2$.

Заметим, что последний столбец $\hat{Q}$ в (1) зависит от всех коэффициентов Гивенса c_i и s_i, $i = 1, \ldots, n-1$, а первый столбец $\tilde{Q}$ в (2) зависит от всех коэффициентов Гивенса g_i и h_i, $i = 1, \ldots, n-1$.

На первый взгляд можно подумать, что даже если “разделяющий” индекс j уже найден, все равно необходимо вычислить и последний столбец $\hat{Q}$, и первый столбец $\tilde{Q}$ для того, чтобы построить искомый собственный вектор.

Далее мы покажем, что искомая аппроксимация собственного вектора может быть вычислена, зная только c_i и s_i, $i = 1, \ldots, j-1$, и g_i и h_i, $i = 1, \ldots, n-j+1$. На самом деле, если индекс j определен, можно заметить, что “хорошая” часть вектора (1) может быть записана в виде

$$\hat{\mathbf{y}}_{1:j} = \begin{bmatrix} (-1)^{n-1} \prod_{i=1}^{n-1} s_i \\ (-1)^{n-2} c_1 \prod_{i=2}^{n-1} s_i \\ (-1)^{n-3} c_2 \prod_{i=3}^{n-1} s_i \\ \vdots \\ (-1)^{n-j+2} c_{j-3} \prod_{i=j-2}^{n-1} s_i \\ (-1)^{n-j+1} c_{j-2} \prod_{i=j-1}^{n-1} s_i \\ (-1)^{n-j} c_{j-1} \prod_{i=j}^{n-1} s_i \end{bmatrix} = \gamma^{(u)} \hat{\mathbf{y}}^{(u)},$$

в то время как “хорошая” часть вектора (2) может быть записана в виде

$$\tilde{\mathbf{y}}_{j:n} = \begin{bmatrix} (-1)^{j-1} g_{n-j} \prod_{i=n-j+1}^{n-1} h_i \\ (-1)^{j} g_{n-j-1} \prod_{i=n-j}^{n-1} h_i \\ (-1)^{j+1} g_{n-j-2} \prod_{i=n-j-1}^{n-1} h_i \\ \vdots \\ (-1)^{n-3} g_2 \prod_{i=3}^{n-1} h_i \\ (-1)^{n-2} g_1 \prod_{i=2}^{n-1} h_i \\ (-1)^{n-1} \prod_{i=1}^{n-1} h_i \end{bmatrix} = \gamma^{(b)} \tilde{\mathbf{y}}^{(b)},$$

где $\gamma^{(u)} = (-1)^{n-j} \prod_{i=j}^{n-1} s_i$, $\gamma^{(b)} = (-1)^{j-1} \prod_{i=n-j+1}^{n-1} h_i$,

$$\hat{\mathbf{y}}^{(u)} = \begin{bmatrix} (-1)^{j-1} \prod_{i=1}^{j-1} s_i \\ (-1)^{j-2} c_1 \prod_{i=2}^{j-1} s_i \\ (-1)^{j-3} c_2 \prod_{i=3}^{j-1} s_i \\ \vdots \\ (-1)^{2} c_{j-3} \prod_{i=j-2}^{j-1} s_i \\ -c_{j-2} s_j \\ c_{j-1} \end{bmatrix}, \quad \tilde{\mathbf{y}}^{(b)} = \begin{bmatrix} g_{n-j} \\ -g_{n-j-1} h_{n-j} \\ (-1)^{2} g_{n-j-2} \prod_{i=n-j-1}^{n-j} h_i \\ \vdots \\ (-1)^{n-j-2} g_2 \prod_{i=3}^{n-j} h_i \\ (-1)^{n-j-1} g_1 \prod_{i=2}^{n-j} h_i \\ (-1)^{n-j} \prod_{i=1}^{n-j} h_i \end{bmatrix}.$$

Тогда для начала мы нормализуем оба вектора следующим образом:

$$\mathbf{y}_{1:j} = \frac{\hat{\mathbf{y}}^{(u)}_{1:j}}{\hat{\mathbf{y}}^{(u)}_{j}}, \quad \mathbf{y}_{j+1:n} = \frac{\overset{(b)}{2:n-j+1}}{\tilde{\mathbf{y}}^{(b)}_{1}},$$

т.е. мы разделим первый вектор на его последнюю компоненту, а второй вектор на его первую компоненту для того, чтобы иметь 1 на j-й позиции первого вектора и на первой позиции второго. А затем, наконец, нормализуем $\mathbf{y}$, т.е. $k\mathbf{y} = \mathbf{y}/\|\mathbf{y}\|_2$.

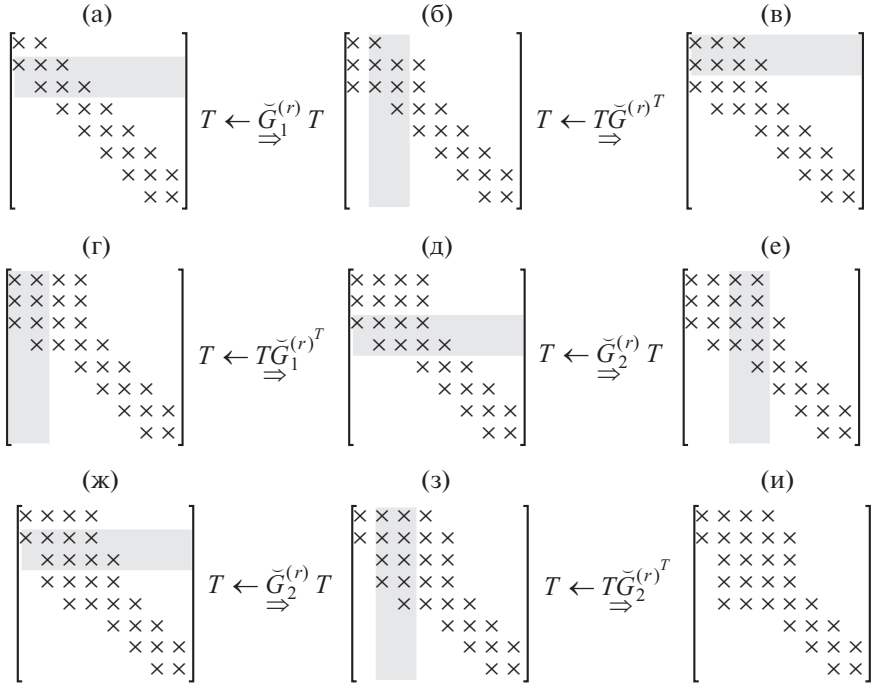
Соответствующий псевдокод на MATLAB приведен в табл. 1.

Вместо того, чтобы останавливать алгоритм, когда $i_1 > i_2$, выполним еще j_f шагов QR-метода и j_b шагов QL-метода, где $j_f = \min\{j + k, n\}$ и $j_b = \max\{1, j - k\}$ при $k \in \{1, \ldots, n\}$. В наших экспериментах было использовано $k = \lfloor \sqrt{n} \rfloor$. Затем вычислим новый вектор $\check{\mathbf{c}} = [\check{c}_1, \ldots, \check{c}_n]^{\mathrm{T}}$, где $\check{c}_i = c_i + g_{i+1}$, $i = 1, \ldots, n$, и выберем $j \in [j_b, j_f]$ таким, что $\check{c}_j \geq \check{c}_\ell$, $\ell \in [j_b, j_f]$, $j \neq \ell$.

Соответствующий код на MATLAB может быть получен от авторов.

Таблица 1. Псевдокод на MATLAB для вычисления собственного вектора несимметричной трехдиагональной матрицы при известном собственном значении λ

```
function[y] = left_eigv(T, λ, n)
R = T − λI_n;  L = R;
i_1 = 1;  i_2 = n;
G_1 = givens(R(i_1, i_1), R(i_1 + 1, i_1));  G_2 = givens(L(i_2, i_2), L(i_2 − 1, i_2));
c_1(1) = G_1(i_1, 1);  s_1(1) = G_1(i_1, 2);  c_2(n − 1) = G_2(i_1, 1);  s_2(i_2 − 1) = G_2(1, 2);
while i_1 <= i_2,
      if c_1(i_1) > c_2(i_2),
         R(i_1 : i_1 + 1, :) = G_1 * R(i_1 : i_1 + 1, :);
         i_1 = i_1 + 1;
         if  i_1 < n,
            G_1 = givens(R(i_1, i_1), R(i_1 + 1, i_1));
            c_1(i_1) = G_1(1, 1);  s_1(i_1) = G_1(1, 2);
         end
      else
         G_2 = G_2.';
         L(i_2 − 1 : i_2, :) = G_2 * L(i_2 − 1 : i_2, :);
         i_2 = i_2 − 1;
         if i_2 > 1,
            G_2 = givens(L(i_2, i_2), L(i_2 − 1, i_2));
            c_2(i_2 − 1) = G_2(1, 1);  c_2(i_2 − 1) = G_2(1, 1);
         end
      end
end
% computation of the eigenvector
j = min(i_1, i_2);
y_1 = zeros(n, 1);  y_2 = zeros(n, 1);
y_1(j) = 1;  y_2(j) = 1;
for i = j − 1 : −1 : 1,
     G = [ c_1(i)   s_1(i)
          −s_1(i)  c_1(i) ];
     y_1(i : i + 1) = G.^T T * y_1(i : i + 1);
end
for i = j : n − 1,
     G = [ c_2(i)   s_2(i)
          −s_2(i)  c_2(i) ];
     y_2(i : i + 1) = G * y_2(i : i + 1);
end
y_1 = y_1/y_1(j);  y_2 = y_2/y_2(j);  y = [y_1(1 : j); y_2(j + 1 : n)];
y = y/‖y‖_2;
```



Фиг. 4. Графическое изображение первых двух шагов одной IQR-итерации со сдвигом λ.

5. КАК ИЗБЕЖАТЬ КОМПЛЕКСНОЙ АРИФМЕТИКИ

Если T является несимметричной трехдиагональной матрицей, ее собственные значения могут быть комплексно-сопряженными.

Множители QR- и QL-разложения $T(\lambda)$ с комплексно-сопряженным сдвигом λ также являются комплексными. Для того, чтобы избежать комплексной арифметики, могут быть применены неявные QR/QL (IQR/IQL) методы с двойным сдвигом [10], [11]. В точной арифметике после одной итерации IQR-метода, примененного к T, с двойным сдвигом λ и $\overline{\lambda}$, где $\overline{\lambda}$ обозначает комплексное сопряжение к λ, мы имеем блочно-хессенбергову структуру

$$\begin{bmatrix} H_1^{(r)} & B \\ & H_2^{(r)} \end{bmatrix},$$

где $H_1^{(r)} \in \mathbb{R}^{(n-2)\times(n-2)}$ и $H_2^{(r)} \in \mathbb{R}^{2\times 2}$ являются хессенберговыми, а λ и $\overline{\lambda}$ – собственные значения $H_2^{(r)}$. Более того, последние два столбца множителя Q являются вещественной и мнимой частями соответствующего собственного вектора.

К сожалению, одна итерация IQR-метода (IQL-метода) преобразовывает трехдиагональную матрицу T в верхнюю (нижнюю) хессенбергову матрицу H, требуя $O(n^2)$ операций в арифметике конечной точности.

Поскольку построение собственного вектора с помощью предложенного метода опирается только на коэффициенты, участвующие во вращениях Гивенса, мы покажем, что нет необходимости обновлять всю хессенбергову матрицу H. Для того, чтобы вычислить коэффициенты Гивенса, достаточно только элементов, близких к трехдиагональной матрице. Продемонстрируем это с помощью первых двух шагов одной IQR-итерации со сдвигом λ, примененным к T. Эти шаги графически изображены на фиг. 4, где серая область обозначает часть матрицы, измененной при умножении на вращения Гивенса.

Пусть $T_0 = T$. Пусть $\mathbf{v} \in \mathbb{R}^n$ является первым столбцом $(T - \lambda I_n)(T - \overline{\lambda} I_n)$. Тогда только первые три элемента $\mathbf{v}$ отличны от нуля.

На первом шаге IQR-итерации строятся два вращения Гивенса $\overset{\vee}{G}{}_1^{(r)}$ и $\overset{\cup}{G}{}_1^{(r)}$ такие, что

$$\overset{\vee}{G}{}_1^{(r)}\overset{\cup}{G}{}_1^{(r)}\mathbf{v} = \mathbf{e}_1,$$

где $\mathbf{e}_1 \in \mathbb{R}^n$ является первым вектором канонического базиса в $\mathbb{R}^n$. Затем первый шаг заканчивается применением следующего преобразования подобия к T:

$$\overset{\vee}{G}_1\overset{\cup}{G}{}_1^{(r)}T\overset{\cup}{G}{}_1^{(r)^{\mathrm{T}}}\overset{\vee}{G}{}_1^{(r)^{\mathrm{T}}}. \tag{5}$$

Умножение $T \leftarrow \overset{\cup}{G}{}_1^{(r)}T$ (фиг. 4а $\rightarrow$ б) приводит к "выпуклости" в позиции $(3,1)$, в то время как $T \leftarrow T\overset{\cup}{G}{}_1^{(r)^{\mathrm{T}}}$ (фиг. 4б $\rightarrow$ в) приводит к "выпуклости" в позиции $(4,2)$. Кроме того, $T \leftarrow \overset{\vee}{G}{}_1^{(r)}T$ и $T \leftarrow T\overset{\vee}{G}{}_1^{(r)^{\mathrm{T}}}$ (фиг. 4в $\rightarrow$ г и фиг. 4г $\rightarrow$ д соответственно) приводят к "выпуклости" в позиции $(4,1)$. Заметим, что в последующих операциях элементы первой строки T (изображены синим на фиг. 4г) не играют роли при вычислении следующих коэффициентов Гивенса и могут быть отброшены.

На втором шаге вращения Гивенса $\overset{\vee}{G}{}_2^{(r)}$ и $\overset{\cup}{G}{}_2^{(r)}$ применены к T для того, чтобы восстановить хессенбергову структуру, зануляя элементы $(4,1)$ и $(3,1)$ (обозначено "$\otimes$" на фиг. 4д и ж). Поскольку $\overset{\vee}{G}{}_2^{(r)}$ и $\overset{\cup}{G}{}_2^{(r)}$ действуют на 4-ю и 3-ю строки и на 3-ю и 2-ю строки соответственно, появляются "выпуклости" в позициях $(4,2)$, $(5,2)$ и $(5,3)$ (фиг. 4и). Заметим, что после умножения $T \leftarrow \overset{\cup}{G}{}_2^{(r)}T$ вторая строка T (изображено синим на фиг. 4г) не играет роли при вычислении других коэффициентов Гивенса, и может быть отброшена.

Подводя итог, в точной арифметике на каждом шаге IQR-метода нужно обновить только несколько элементов матрицы T для того, чтобы вычислить последовательности вращений Гивенса $\{\overset{\vee}{G}{}_i^{(r)}\}_{i=1}^{n-1}$ и $\{\overset{\cup}{G}{}_i^{(r)}\}_{i=1}^{n-1}$, причем число операций линейно зависит от n – порядка матрицы T. Похожим образом на каждом шаге IQL-метода с двойным сдвигом λ и $\overline{\lambda}$ нужно обновить только несколько элементов матрицы T для того, чтобы вычислить две последовательности вращений Гивенса $\{\overset{\vee}{G}{}_i^{(l)}\}_{i=1}^{n-1}$ и $\{\overset{\cup}{G}{}_i^{(l)}\}_{i=1}^{n-1}$, которые преобразовывают матрицу в подобную нижнюю блочно-хессенбергову матрицу

$$\begin{bmatrix} H_1^{(l)} & \\ F & H_2^{(l)} \end{bmatrix},$$

где $H_1^{(l)} \in \mathbb{R}^{2\times 2}$ и $H_2^{(l)} \in \mathbb{R}^{(n-2)\times(n-2)}$ являются хессенберговыми, а λ и $\overline{\lambda}$ – собственные значения $H_1^{(l)}$. Левый собственный вектор $\mathbf{y}$, соответствующий λ, может быть получен либо из последних двух столбцов $Q^{(r)}$, либо из первых двух столбцов $Q^{(l)}$:

$$\mathbf{y} \equiv \mathbf{y}^{(r)} \equiv \mathbf{y}^{(l)},$$

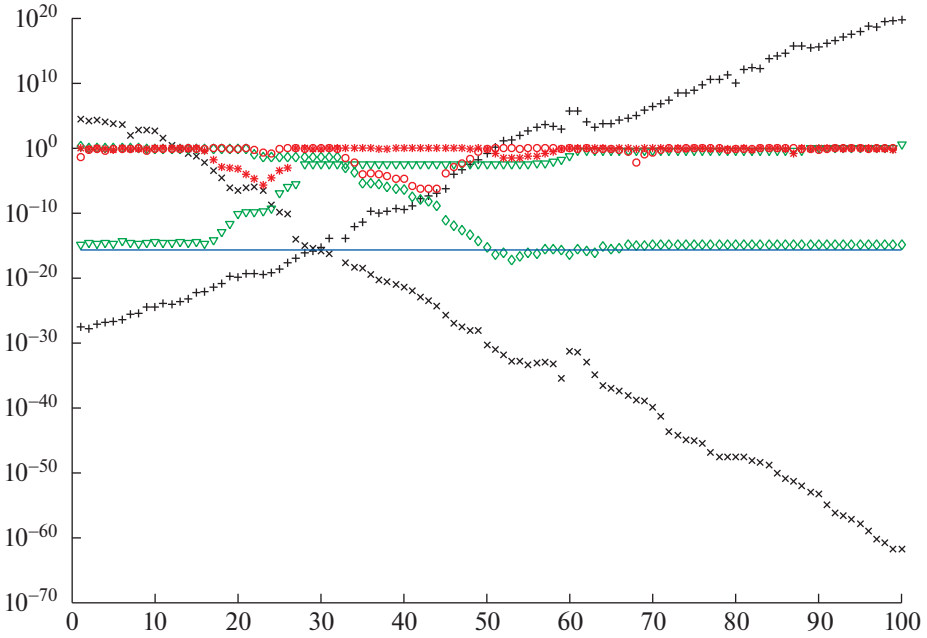
где

$$\mathbf{y}^{(r)} = Q^{(r)}(:, n-1) + i \cdot Q^{(r)}(:, n), \quad \mathbf{y}^{(l)} = Q^{(l)}(:, 1) + i \cdot Q^{(l)}(:, 2), \tag{6}$$

и

$$Q^{(r)} = \overset{\cup}{G}{}_1^{(r)^{\mathrm{T}}}\overset{\vee}{G}{}_1^{(r)^{\mathrm{T}}} \cdots \overset{\cup}{G}{}_{n-1}^{(r)^{\mathrm{T}}}\overset{\vee}{G}{}_{n-1}^{(r)^{\mathrm{T}}},$$
$$Q^{(l)} = \overset{\cup}{G}{}_1^{(l)^{\mathrm{T}}}\overset{\vee}{G}{}_1^{(l)^{\mathrm{T}}} \cdots \overset{\cup}{G}{}_{n-1}^{(l)^{\mathrm{T}}}\overset{\vee}{G}{}_{n-1}^{(l)^{\mathrm{T}}}.$$

В арифметике конечной точности аналогично случаю, анализируемому в разд. 3, при вычислении вышеупомянутых последовательностей может возникнуть прямая неустойчивость. Индекс j может быть найден из последовательностей $\check{c}_i^{(r)}$, $i = 1, \ldots, n-1$, и $\check{c}_i^{(l)}$, $i = 1, \ldots, n-1$, порожденных итерацией IQR и IQL-методов с двойным сдвигом λ и $\overline{\lambda}$. Для краткости мы опускаем детали.



Фиг. 5. Пример 1: график $\left|\mathbf{y} - \tilde{\mathbf{y}}^{(r)}\right|$ (“×”), $\left|\mathbf{y} - \tilde{\mathbf{y}}^{(l)}\right|$ (“+”), $\{\hat{\sigma}_i\}_{i=1}^{n}$ (“◇”), $\{\tilde{\sigma}_i\}_{i=1}^{n}$ (“▽”), $\{\check{c}_i^{(r)}\}_{i=1}^{n-1}$ (“○”) и $\{\check{c}_i^{(l)}\}_{i=1}^{n-1}$ (“∗”), машинная точность ε_M (синяя линия) в логарифмическом масштабе.

Применим IQR- и IQL-методы с двойным сдвигом к данным из примера 1. В частности, собственные векторы, обозначенные соответственно $\tilde{\mathbf{y}}^{(r)}$ и $\tilde{\mathbf{y}}^{(l)}$ в (6), были вычислены в арифметике конечной точности. На фиг. 5 в логарифмическом масштабе изображены покомпонентные ошибки $\left|\mathbf{y} - \tilde{\mathbf{y}}^{(r)}\right|$ (обозначено “×”) и $\left|\mathbf{y} - \tilde{\mathbf{y}}^{(l)}\right|$ (обозначено “+”), последовательности $\{\hat{\sigma}_i\}_{i=1}^{n}$ (обозначено “◇”), $\{\tilde{\sigma}_i\}_{i=1}^{n}$ (обозначено “○”), $\{\check{c}_i^{(r)}\}_{i=1}^{n-1}$ (обозначено “○”) и $\{\check{c}_i^{(l)}\}_{i=1}^{n-1}$ (“∗”), машинная точность ε_M (обозначено синей линией). Поведение векторов и последовательностей, изображенное на фиг. 5, то же самое, как описано в примере 4.

6. ЧИСЛЕННЫЕ ПРИМЕРЫ

Все численные эксперименты этого раздела были выполнены в `MATLAB Ver. 2014b` с машинной точностью $\varepsilon_M \sim 2.22 \times 10^{-16}$.

Пример 2. В этом примере рассматривается несимметричная трехдиагональная матрица $T_n \in \mathbb{R}^{n\times n}$, $n = 200$, элементы которой порождены функцией `randn` в `MATLAB`.

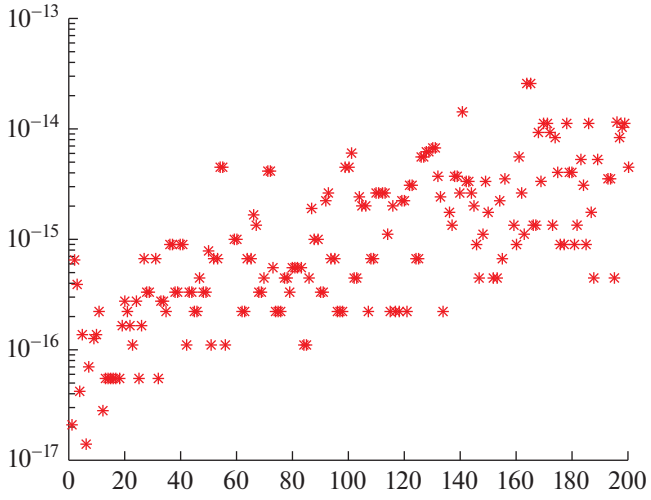
С помощью описанного метода были вычислены собственные векторы $\hat{\mathbf{y}}_i$, $i = 1, \ldots, n$, а затем собственные значения были пересчитаны с помощью отношения Рэлея:

$$\hat{\lambda}_i = \frac{\hat{\mathbf{y}}_i^{\mathrm{H}} T \hat{\mathbf{y}}_i}{\hat{\mathbf{y}}_i^{\mathrm{H}} \hat{\mathbf{y}}_i}, \quad i = 1, \ldots, n.$$

На фиг. 6 изображены модули разности $\left|\hat{\lambda}_i - \lambda_i\right|$, $i = 1, \ldots, n$, где λ_i – собственные значения, вычисленные функцией `eig` в `MATLAB`. Можно видеть, что разность имеет порядок ε_M.

Для каждого собственного вектора $\tilde{\mathbf{y}}_i^{(l)}$, $i = 1, \ldots, n$, вычисленного с помощью предложенного метода, обозначим

$$\nu_i = \left\| \tilde{\mathbf{y}}_i^{(l)^{\mathrm{H}}} T - (\tilde{\mathbf{y}}_i^{(l)^{\mathrm{H}}} T \tilde{\mathbf{y}}_i^{(l)}) \tilde{\mathbf{y}}_i^{(l)^{\mathrm{H}}} \right\|_2$$



Фиг. 6. Пример 2: ошибка $\left|\hat{\lambda}_i - \lambda_i\right|$, $i = 1, \ldots, n$, в логарифмическом масштабе.

норму i-й невязки. Тогда оказывается, что

$$1.16 \times 10^{-13} = \max_k \nu_k \le \nu_i \le \min_k \nu_k = 2.18 \times 10^{-16}.$$

Можно сделать вывод, что собственные векторы, полученные с помощью предложенной процедуры, вычислены достаточно точно.

Пример 3. В этом примере мы рассмотрим матрицу Бесселя

$$B_n^{(a,b)} = \begin{bmatrix} \alpha_1 & \gamma_1 & & & \\ \beta_1 & \alpha_2 & \gamma_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \alpha_{n-1} & \gamma_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix},$$

где $n = 50$ и

$$\alpha_1 = -\frac{b}{a}, \quad \alpha_j = -b\frac{a-2}{(2j+a-2)(2j+a-4)}, \quad j = 2, \ldots, n,$$

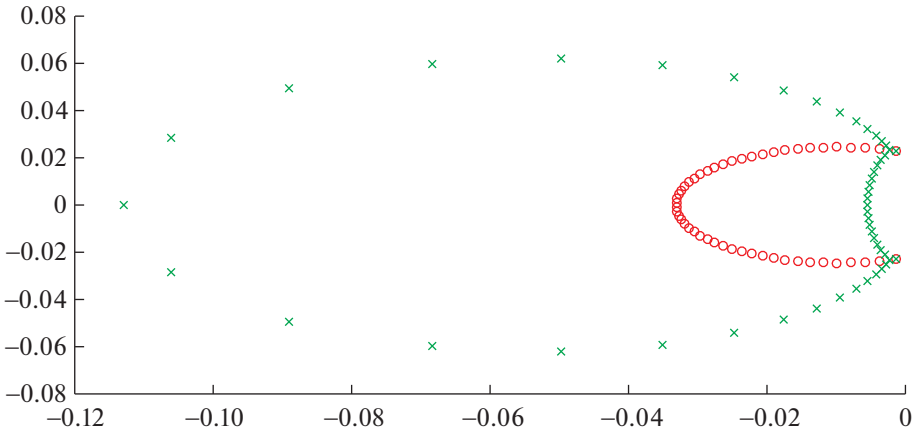
$$\beta_1 = -\frac{\alpha_1}{a+1}, \quad \alpha_j = -b\frac{j}{(2j+a-1)(2j+a-2)}, \quad j = 2, \ldots, n-1,$$

$$\gamma_1 = -\alpha_1, \quad \gamma_j = b\frac{j+a-2}{(2j+a-2)(2j+a-3)}, \quad j = 2, \ldots, n-1.$$

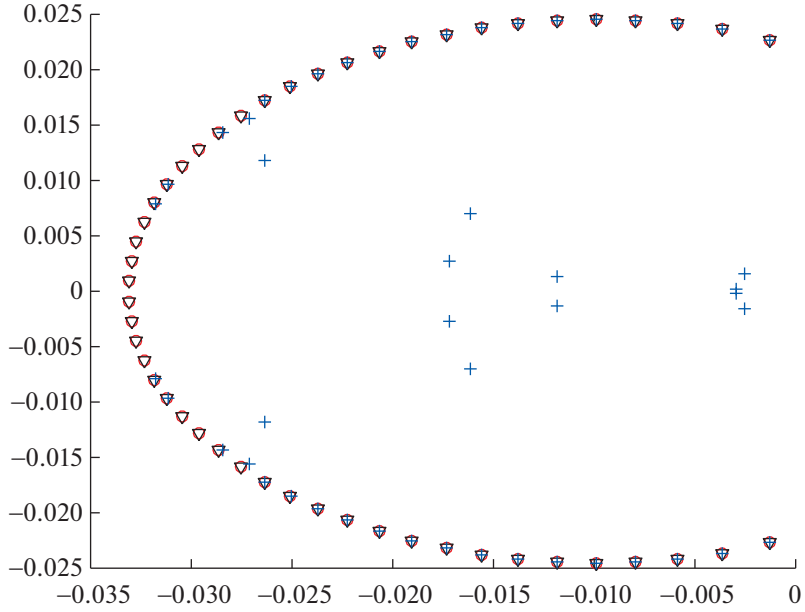
Собственные значения матриц $B_n^{(a,b)}$ с ростом n страдают от плохой обусловленности [5]. Положим $a = -4.5$ и $b = 2$, как это сделано в [5], и будем обозначать $B_{50}^{(-4.5,2)}$ просто как B. Эти матрицы имеют хорошо отделимые комплексные собственные значения.

Матрица B и ее собственные значения λ_i, $i = 1, \ldots, 50$, были вычислены в MATLAB с использованием арифметики переменной точности с 128 значащими цифрами, а затем округлены до двойной точности. Таким образом, были получены $\tilde{B}$ и $\tilde{\lambda}_i$, $i = 1, \ldots, 50$.

Пусть $\hat{\lambda}_i$, $i = 1, \ldots, 50$, являются собственными значениями $\tilde{B}$, полученными с помощью `eig` в MATLAB. На фиг. 7 $\{\tilde{\lambda}_i\}_{i=1}^{50}$ и $\}\hat{\lambda}_i\{_{i=1}^{50}$ обозначены “$\circ$” и “$\times$” соответственно. Заметим, что $\tilde{\lambda}_i$ и $\hat{\lambda}_i$ сильно отличаются, поскольку, как было отмечено выше, собственные значения B страдают от плохой обусловленности с ростом n [5].



Фиг. 7. Пример 3: собственные значения матрицы B обозначены “$\circ$”, а величины, вычисленные с помощью `eig` в MATLAB, обозначены “$\times$”.



Фиг. 8. Пример 3. Графики $\tilde{\lambda}_i$, $\tilde{\lambda}_i^{(1)}$ и $\tilde{\lambda}_i^{(2)}$, $i = 1, \ldots, 50$, обозначены, соответственно, как “$\bigcirc$”, “$\bigtriangledown$” и “+”.

Левые собственные векторы, соответствующие собственным значениям $\tilde{\lambda}_i$ матрицы $\tilde{B}$, были вычислены с помощью предложенного метода и обозначены $\tilde{\mathbf{y}}_i^{(1)}$. Последний столбец матрицы Q из QR-разложения матрицы $\tilde{B} - \tilde{\lambda}_i I$ обозначен через $\tilde{\mathbf{y}}_i^{(2)}$. Таким образом, новые аппроксимации собственных значений $\tilde{B}$ были вычислены как отношения Рэлея:

$$\tilde{\lambda}_i^{(1)} = \tilde{\mathbf{y}}_i^{(1)\mathrm{T}} \tilde{B} \tilde{\mathbf{y}}_i^{(1)}, \quad \tilde{\lambda}_i^{(2)} = \tilde{\mathbf{y}}_i^{(2)\mathrm{T}} \tilde{B} \tilde{\mathbf{y}}_i^{(2)}, \quad i = 1, \ldots, 50.$$

Собственные значения $\tilde{\lambda}_i$ и вычисленные аппроксимации ($\tilde{\lambda}_i^{(1)}$ и $\tilde{\lambda}_i^{(2)}$, $i = 1, \ldots, 50$) собственных значений $\tilde{B}$ изображены на фиг. 8 и обозначены, соответственно, символами “$\bigcirc$”, “$\bigtriangledown$” и “+”.

Заметим, что $\tilde{\lambda}_i$ и $\tilde{\lambda}_i^{(1)}$ хорошо совпадают, в то время как $\tilde{\lambda}_i^{(2)}$ отклоняется от $\tilde{\lambda}_i$ с ростом отрицательной мнимой части.

Таблица 2. $\max_i \left|\tilde{\lambda}_i - \tilde{\lambda}_i^{(1)}\right|$ и $\max_i \left|\tilde{\lambda}_i - \tilde{\lambda}_i^{(2)}\right|$

$\max_i \left\|\tilde{\lambda}_i - \tilde{\lambda}_i^{(1)}\right\|$	$\max_i \left\|\tilde{\lambda}_i - \tilde{\lambda}_i^{(2)}\right\|$
3.06×10^{-15}	3.01×10^{2}

В табл. 2 приведены $\max_i \left|\tilde{\lambda}_i - \tilde{\lambda}_i^{(1)}\right|$ и $\max_i \left|\tilde{\lambda}_i - \tilde{\lambda}_i^{(2)}\right|$.

Заметим, что если вычислить аппроксимацию собственного вектора $\breve{\mathbf{y}}$ для одного из собственных значений $\tilde{\lambda}_i$ матрицы $\tilde{B}$ с помощью метода обратных итерацией, то вектор $\breve{\mathbf{y}}$ оказывается далек от $\tilde{\mathbf{y}}_i^{(1)}$. Это поведение описано в [18]. Более того, соответствующее отношение Рэлея $\breve{\mathbf{y}}^{\mathrm{H}}\tilde{B}\breve{\mathbf{y}}/(\breve{\mathbf{y}}^{\mathrm{H}}\breve{\mathbf{y}})$ далеко от собственного значения $\tilde{\lambda}_i$.

Для каждого собственного вектора $\tilde{\mathbf{y}}_i^{(1)}$, $i = 1, \ldots, 50$, вычисленного с помощью предложенного метода, обозначая через

$$\nu_i = \left\|\tilde{\mathbf{y}}_i^{(1)^{\mathrm{H}}}\tilde{B} - (\tilde{\mathbf{y}}_i^{(1)^{\mathrm{H}}}\tilde{B}\tilde{\mathbf{y}}_i^{(1)})\tilde{\mathbf{y}}_i^{(1)^{\mathrm{H}}}\right\|_2$$

норму i-й невязки, получаем, что

$$3.06\times10^{-15} = \max_k \nu_k \le \nu_i \le \min_k \nu_k = 2.61\times10^{-16}.$$

Пример 4. Рассмотрим в этом примере матрицы Клемента порядка $n = 200$ [7], также называемые матрицами Сильвестра–Каца [19]:

$$C_n = \begin{bmatrix} & \gamma_1 & & & \\ \beta_1 & & \gamma_2 & & \\ & \beta_2 & & \ddots & \\ & & \ddots & & \gamma_{n-1} \\ & & & \beta_{n-1} & \end{bmatrix},$$

где $\gamma_i = i$ и $\beta_i = n - i$, $i = 1, \ldots, n-1$.

Собственные значения таких матриц равны [19] $\lambda_k = n - 2(k-1)$ при $k = 1, \ldots, n$, а соответствующая левая матрица собственных векторов [19] равна:

$$u_{k,j} = \sum_{i=0}^{\min(k-1, j-1)} (-1)^i \binom{j-1}{i}\binom{n-j-1}{k-i-1}.$$

Также, в этом примере для каждого собственного вектора $\tilde{\mathbf{y}}_i^{(1)}$, $i = 1, \ldots, n$, вычисленного с помощью предложенного метода, обозначая через

$$\nu_i = \left\|\tilde{\mathbf{y}}_i^{(1)^{\mathrm{H}}}C_n - (\tilde{\mathbf{y}}_i^{(1)^{\mathrm{H}}}C_n\tilde{\mathbf{y}}_i^{(1)})\tilde{\mathbf{y}}_i^{(1)^{\mathrm{H}}}\right\|_2$$

норму i-й невязки, получаем, что

$$5.67\times10^{-13} = \max_k \nu_k \le \nu_i \le \min_k \nu_k = 4.49\times10^{-16}.$$

7. ЗАКЛЮЧЕНИЕ

В этой статье рассмотрена задача вычисления собственного вектора, соответствующего вычисленному собственному значению λ несимметричной трехдиагональной матрицы T. Сначала описано, как левый собственный вектор $\mathbf{y}$, соответствующий собственному значению λ матрицы T, может быть получен применением одной итерации QR- или QL-метода со сдвигом λ. К сожалению, оба метода страдают от прямой неустойчивости при работе в арифметике конечной точности.

Был предложен метод вычисления индекса j, который комбинирует собственные векторы, полученные итерацией QR- и итерацией QL-методов со сдвигом λ, избегая неустойчивости. Индекс j вычисляется рассмотрением двух последовательностей косинусов, порожденных итерацией QR- и QL-методов со сдвигом λ. Искомый собственный вектор получается из первых j коэффициентов Гивенса, порожденных QR-методом и первых $n - j$ коэффициентов Гивенса, порожденных QL-методом.

Итоговая сложность вычисления собственного вектора линейно зависит от размера матрицы.

Результаты численных экспериментов выглядят очень многообещающими, показывая, что собственные векторы вычисляются с высокой точностью.

СПИСОК ЛИТЕРАТУРЫ

1. *Sidje R.B., Burrage K. QRT*: A *QR*- based tridiagonalization algorithm for nonsymmetric matrices // SIAM J. Matrix Anal. Appl. 2005. V. 26. № 3. P. 878–900.
2. *Dongarra J.J., Geist G.A., Romine C.H.* Algorithm 710: FORTRAN subroutines for computing the eigenvalues and eigenvectors of a general matrix by reduction to general tridiagonal form // ACM Trans. Math. Softw. 1992. V. 18. № 4. P. 392–400.
3. *Wang H.H., Gregory R.T.* On the reduction of an arbitrary real square matrix to tridiagonal form // Math. Comput. 1964. V. 18. P. 501–505.
4. *Freund R.W., Gutknecht M.H., Nachtigal N.M.* An implementation of the lookahead Lanczos algorithm for non-Hermitian matrices // SIAM J. Sci. Comput. 1993. V. 14. P. 137–158.
5. *Ferreira C., Parlett B., Dopico F.M.* Sensitivity of eigenvalues of an unsymmetric tridiagonal matrix // Numerische Mathematik. 2012. V. 122. № 3. P. 527–555.
6. *Pasquini L.* Accurate computation of the zeros of the generalized Bessel polynomials // Numerische Mathematik. 2000. V. 86. № 3. P. 507–538.
7. *Bini D.A., Gemignani L., Tisseur F.* The Ehrlich–Aberth method for the nonsymmetric tridiagonal eigenvalue problem // SIAM J. Matrix Anal. Appl. 2005. V. 27. № 1. P. 153–175.
8. *Dax A., Kaniel S.* The *ELR* method for computing the eigenvalues of a general matrix // SIAM Journal on Numerical Analysis. 1981. V. 18. № 4. P. 597–605.
9. *Rutishauser H.* Solution of eigenvalue problems with the *LR*-transformation // Nat. Bur. Standards. 1958. V. 49. P. 47–81.
10. *Francis J.* The *QR* transformation a unitary analogue to the *LR* transformation. I // Comput. J. 1961. V. 4. P. 265–271.
11. *Francis J.* The *QR* transformation a unitary analogue to the *LR* transformation. II // Comput. J. 1961. V. 4. P. 332–345.
12. *Golub G.H., Van Loan C.F.* Matrix Computations, 4th ed. Baltimore: Johns Hopkins University Press, 2013.
13. *Parlett B.N., Le J.* Forward instability of tridiagonal *QR* // SIAM J. Matrix Anal. Appl. 1993. V. 14. № 1. P. 279–316.
14. *Watkins D.S.* The transmission of shifts and shift blurring in the *QR* algorithm // Linear Algebra and its Applications. 1996. V. 241–243. P. 877–896.
15. *Mastronardi N., Taeter H., Dooren P.V.* On computing eigenvectors of symmetric tridiagonal matrices // Structured Matrices in Numerical Linear Algebra: Analysis, Algorithms and Applications. 2019. V. 30. P. 181–195.
16. *Horn R.A., Johnson C.R.* Topics in Matrix Analysis. New York: Cambridge University Press, 1991.
17. *Chang X.-W., Paige C.C., Stewart G.W.* Perturbation analyses for the *QR* factorization // SIAM J. Matrix Anal. Appl. 1997. V. 18. № 3. P. 775–791.
18. *Ipsen I.C.F.* Computing an eigenvector with inverse iteration // SIAM Review. 1997. V. 39. № 2. P. 254–291.
19. *Chu W., Wang X.* Eigenvectors of tridiagonal matrices of Sylvester type // Calcolo. 2008. V. 45. № 4. P. 217–233.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 776–786

ОБЩИЕ ЧИСЛЕННЫЕ МЕТОДЫ

УДК 519.65

О TT-РАНГАХ ПРИБЛИЖЕННЫХ ТЕНЗОРИЗАЦИЙ НЕКОТОРЫХ ГЛАДКИХ ФУНКЦИЙ[1)]

© 2021 г. Л. И. Высоцкий[1,2]

[1] 119333 Москва, ул. Губкина, 8, Ин-т вычисл. матем. им. Г.И. Марчука РАН, Россия

[2] 119991 Москва, Ленинские горы, МГУ им. М.В. Ломоносова, ВМК, Россия

e-mail: vysotskylev@yandex.ru

Поступила в редакцию 24.11.2020 г.
Переработанный вариант 24.11.2020 г.
Принята к публикации 14.01.2021 г.

Исследуются “тензоризации” функций, т.е. тензоры с элементами $A(i_1,\ldots,i_d) = f(x(i_1,\ldots,i_d))$, где $f(x)$ – некоторая функция, заданная на отрезке, а $\{x(i_1,\ldots,i_d)\}$ – сетка на этом отрезке. Для таких тензоров ставится задача приближения тензорами, допускающими TT (Tensor Train)-разложение с малыми TT-рангами. Для класса функций, являющихся следами аналитических в некоторых эллипсах на комплексной плоскости функций комплексного переменного, получены верхние и нижние оценки TT-рангов оптимальных приближений. Указанные оценки применены к тензоризациям полиномиальных функций. В частности, известная верхняя граница TT-рангов приближений таких функций улучшена до $O(\log n)$, где n – степень полинома. Библ. 10.

Ключевые слова: TT-разложение, tensor train, TT-ранги, тензоризации функций, приближения.

DOI: 10.31857/S0044466921050173

1. ВВЕДЕНИЕ

Относительно недавние успехи по преодолению “проклятия размерности” позволяют оперировать с дискретизациями функций на очень густых сетках. Например (в духе работ [1]–[5]), заданную на отрезке $[L, R]$ функцию $f(x)$ можно дискретизировать, т.е. превратить в вектор с элементами $f(x(i))$, где $\{x(i)\}_{i=0}^{N-1}$ – произвольная сетка на $[L, R]$. Если $N = n_1 \times \ldots \times n_d$, то полученный вектор можно рассматривать (см. [6]) как тензор (многомерный массив) размером $n_1 \times \ldots \times n_d$ с элементами:

$$A(i_1,\ldots,i_d) = f(x(i_1N_1 + \ldots + i_dN_d)), \quad N_k = n_{k+1}\cdots n_d, \quad i_k \in \{0,\ldots,n_k - 1\}.$$

Тензор A будем называть *тензоризацией* функции f на сетке $\{x(i)\}$. Этот тензор на практике часто допускает приближение другим тензором B, имеющим малопараметрическое представление. К примеру, можно использовать TT-разложение [2], [7] (tensor train, тензорный поезд):

$$B(i_1,\ldots,i_d) = \sum_{\alpha_1,\ldots,\alpha_{d-1}} H_1(i_1,\alpha_1)\cdots H_k(\alpha_{k-1},i_k,\alpha_k)\cdots H_d(\alpha_{d-1},i_d), \tag{1}$$

где H_k имеет размеры $r_{k-1} \times n_k \times r_k$, а индекс суммирования α_k пробегает значения от 1 до r_k. Числа r_k называются TT-рангами представления (1), причем для единообразия определения тензоров H_k считается, что $r_0 = r_d = 1$.

Стоит также отметить, что в случае тензоризаций высокой размерности, когда число N очень велико, вычисление и хранение в памяти элементов тензора A не представляется возможным. Тем не менее иногда есть возможность получить сразу представление (1), вычисляя $f(x(i))$ лишь для небольшого количества точек $x(i)$. Например, можно использовать *крестовое приближение* (*TT-cross approximation*) из [8].

[1)]Работа выполнена при поддержке Московского центра фундаментальной и прикладной математики (соглашение 075-15-2019-1624 с Минобрнауки РФ).

Величины ТТ-рангов играют ключевую роль для применимости ТТ-разложения, ведь требуемая для хранения тензоров H_k память и сложность распространенных операций над тензорами в этом формате растут пропорционально полиномам малой степени от r_k [7]. Поэтому представляется интересным исследовать, насколько малы могут быть ТТ-ранги приближений тензоризаций функций.

Для некоторых классов функций известны компактные представления для тензоризаций, например, для экспоненты, синуса и некоторых других [1], [3]. Для тензоризации полинома степени n на равномерной сетке известно [2], [3], что все ее ТТ-ранги ограничены $n+1$. В [2] было показано, что ТТ-ранги приближений тензоризаций так называемых *асимптотически гладких* (т.е. бесконечно дифференцируемых с "не слишком быстро" растущими при приближении к сингулярностям производными) функций ограничены числом, растущим при уменьшении требуемой погрешности ε как $-\log_2(\varepsilon)$. В [5] были получены оценки ТТ-рангов приближений тензоризаций полиномов на равномерных сетках, улучшающие известную оценку $n+1$.

В данной работе для тензоризаций вещественнозначных функций, являющихся следами аналитических в некотором эллипсе функций комплексного переменного, доказаны верхние и нижние оценки ТТ-рангов приближений. Полученные оценки были применены к тензоризациям полиномов, существенно улучшив известные результаты: вместо $O(\sqrt[3]{n})$ из [5] доказана оценка $O(\ln n)$.

2. НЕОБХОДИМЫЕ ОПРЕДЕЛЕНИЯ

Определение 1. *Матрицами развертки* тензора $A \in \mathbb{R}^{n_1\times\ldots\times n_d}$ называются матрицы

$$A_k \in \mathbb{R}^{(n_1\cdots n_k)\times(n_{k+1}\cdots n_d)}, \quad A_k(i_1,\ldots,i_k;i_{k+1},\ldots,i_d) = A(i_1,\ldots,i_d),$$

где группы индексов до и после точки с запятой образуют так называемые *мультиндексы*, отождествляемые со своим номером (нумерация с нуля) в лексикографическом (словарном) порядке.

Для тензора $A \in \mathbb{R}^{n_1\times\ldots\times n_d}$ определим нормы

$$\|A\|_\infty = \max|A(i_1,\ldots,i_d)| \quad \text{и} \quad \|A\|_F = \sqrt{\sum (A(i_1,\ldots,i_d))^2}.$$

В основном мы будем рассматривать "кубические" тензоризации на равномерных сетках, т.е. тензоры, все размеры которого равны одному и тому же числу b, а элементы заданы значениями функции $f(x)$ в равноудаленных узлах. Такие d-мерные тензоризации с шагом $(R-L)b^{-d}$ на отрезке $[L,R]$ будем обозначать через $T_{b,d,[L,R]}(f)$.

Для ТТ-разложения вида (1) d-мерного тензора $B \in \mathbb{R}^{b\times\ldots\times b}$ определим $\mathcal{R}(H_1,\ldots,H_d)$ как максимальный из его ТТ-рангов. Далее определим $\mathcal{R}(B) = \min \mathcal{R}(H_1,\ldots,H_d)$, где минимум берется по всевозможным ТТ-разложениям вида (1) тензора B. Для оценки ТТ-рангов наилучших приближений нам понадобится функция

$$\mathcal{R}_\varepsilon(A) = \min_{B:\|B-A\|_F\le\varepsilon} \mathcal{R}(B).$$

Определение 2. Эллипс на комплексной плоскости с центром в нуле и осями, параллельными осям координат, чьи полуоси равны $(\rho+\rho^{-1})/2$ и $(\rho-\rho^{-1})/2$ для некоторого $\rho>1$, будем называть *эллипсом Бернштейна* и обозначать через Γ_ρ. Также эллипс Бернштейна можно рассматривать (см. [9]) как образ окружности $\{\rho e^{i\varphi}: \varphi\in[0,2\pi)\}$ под действием отображения Жуковского $w \to (w+w^{-1})/2$. Такой же эллипс, но с центром в точке $z\in\mathbb{C}$, будем обозначать через $\Gamma_\rho(z)$.

Определение 3. *Чебышёвской сеткой* на $[-1,1]$ с числом узлов n будем называть упорядоченное множество точек $\{x_0,\ldots,x_{n-1}\}$, где $x_j = \cos(\pi/(2n)+\pi j/n)$. Известно [9], что числа x_j являются корнями *полинома Чебышёва* степени n, т.е. $T_n(x) = \cos(\arccos(nx))$.

Круг радиуса r с центром в точке z на комплексной плоскости будем обозначать через $U(r,z)$.

3. ВЕРХНИЕ ОЦЕНКИ

Докажем некоторые верхние оценки TT-рангов и связанных с ними величин для приближений тензоризаций функций на равномерных сетках. Основная лемма 1 уточняет известную (см., например, [9]) оценку точности приближения функции, являющейся следом аналитической в эллипсе Бернштейна. Теорема 1 демонстрирует возможность низкорангового приближения матриц развертки тензоризаций функций, аналитических в круге, чей диаметр является отрезком дискретизации. Следствие 1 дает оценку непосредственно для величины $\mathcal{R}_\epsilon(T_{b,d,[L,R]}(f))$ для функций f с указанным свойством. Наконец, теорема 2 рассматривает более простой случай, когда функция является аналитической в достаточно большом эллипсе. Заметим, что теорему 1 может иметь смысл применять даже для функций, являющихся аналитическими в произвольно больших множествах, так как в оценку теоремы 2 входит максимум модуля $f(z)$ на некотором эллипсе – число, которое может быть нежелательно велико для интересующей нас функции $f(z)$. Эта особенность будет использована ниже при применении полученных результатов к полиномам.

Лемма 1. *Пусть $f(x)$ – след аналитической внутри эллипса Бернштейна Γ_ρ, $\rho > 1$, функции $f(z)$, причем $|f(z)| \le M$ на Γ_ρ. Пусть также $P_n(x)$ – полином Лагранжа степени n для $f(x)$ на чебышёвской сетке на $[-1,1]$ с $(n+1)$ узлом $\{x_0,\dots,x_n\}$. Тогда для всех $x \in [-1,1]$ выполнено следующее*:

$$|f(x) - P_n(x)| \le \frac{M}{\rho^{n+1} - \rho^{-n-1}} \frac{\rho + \rho^{-1}}{\frac{1}{2}(\rho + \rho^{-1} - 1)}.$$

Доказательство. Зафиксируем произвольное $x \in [-1,1]\backslash\{x_0,\dots,x_n\}$. Далее рассмотрим функцию

$$F(z) = \frac{f(z)}{(z-x)\prod_{j=0}^{n}(z-x_j)}$$

и, применяя теорему о вычетах и домножая на $\prod_j (x - x_j)$ (т.е. рассуждая аналогично [9, Теорема 13.6]), придем к представлению:

$$\prod_{j=0}^{n}(x-x_j)\int_{\Gamma_\rho} F(z)dz = 2\pi i\left(f(x) - \underbrace{\sum_{\ell=0}^{n} f(x_\ell)\frac{\prod_{j\ne\ell}(x-x_j)}{\prod_{j\ne\ell}(x_\ell - x_j)}}_{P_n(x)}\right).$$

Так как x_j суть в точности корни полинома Чебышёва $T_{n+1}(x)$ степени $n+1$, то можно записать

$$T_{n+1}(x)\int_{\Gamma_\rho}\frac{f(z)dz}{(z-x)T_{n+1}(z)} = 2\pi i(f(x) - P_n(x)).$$

Используя доказанное в [9, Теорема 13.6] для $z \in \Gamma_\rho$ неравенство $|T_{n+1}(z)| \ge (\rho^{n+1} - \rho^{-n-1})/2$, а также очевидное для $x \in [-1,1]$ неравенство $|T_{n+1}(x)| \le 1$, при $x \in [-1,1]$ получаем

$$|f(x) - P_n(x)| \le \frac{1}{2\pi}|T_{n+1}(x)|\left|\int_{\Gamma_\rho}\frac{f(z)dz}{(z-x)T_{n+1}(z)}\right| \le \frac{1}{2\pi}\int_{\Gamma_\rho}\frac{M|dz|}{|z-x||T_{n+1}(z)|} \le$$
$$\le \frac{M}{2\pi}\frac{2}{\rho^{n+1} - \rho^{-n-1}}\int_{\Gamma_\rho}\frac{|dz|}{|z-x|} \le \frac{M}{\pi(\rho^{n+1} - \rho^{-n-1})}\frac{|\Gamma_\rho|}{\min\limits_{z\in\Gamma_\rho}|z-x|}.$$

Для оценки длины эллипса Γ_ρ воспользуемся упомянутым фактом, что первый является образом окружности $S_\rho = \{\rho e^{i\varphi} : \varphi \in [0, 2\pi)\}$ под действием отображения $w \to (w + w^{-1})/2$:

$$|\Gamma_\rho| = \int\limits_{\Gamma_\rho} |dz| = \int\limits_{S_\rho} \left|\frac{1}{2} dw - \frac{1}{2} w^{-2} dw\right| = \int\limits_0^{2\pi} \frac{1}{2} \left|1 - (\rho e^{i\varphi})^{-2}\right| \rho d\varphi \le \frac{1}{2}(1 + \rho^{-2})\rho 2\pi.$$

Далее вычислим $\min |z - x|^2$ для $z \in \Gamma_\rho$ и $x \in [-1, 1]$. Обозначим через r_1 и r_2 большую и меньшую полуоси эллипса Γ_ρ соответственно. Тогда точки этого эллипса параметризуются углом φ: $z = r_1 \cos\varphi + i r_2 \sin\varphi$. Преобразуем минимизируемое выражение

$$|z - x|^2 = (r_1 \cos\varphi - x)^2 + (r_2 \sin\varphi)^2 = (r_1^2 - r_2^2)\cos^2\varphi - 2r_1 x \cos\varphi + x^2 + r_2^2. \tag{2}$$

В силу симметрии достаточно рассмотреть $x \ge 0$. При фиксированном x выражение (2) – это квадратный трехчлен относительно $\cos\varphi$, причем вершина соответствующей параболы имеет абсциссу $r_1 x/(r_1^2 - r_2^2) = r_1 x$. При $r_1 x \le 1$ минимум по φ достигается при $\cos\varphi = r_1 x$ и равен r_2^2. При $r_1 x > 1$ минимум достигается при $\cos\varphi = 1$ и равен $(r_1 - x)^2$, причем это выражение достигает минимума по x при $x = 1$. Так как $r_1 - 1 < r_2$, то и $\min |z - x| = r_1 - 1 = (\rho + \rho^{-1})/2 - 1$.

В итоге имеем

$$|f(x) - P_n(x)| \le \frac{M\rho\pi(1 + \rho^{-2})}{\pi(\rho^{n+1} - \rho^{-n-1})\left(\frac{1}{2}(\rho + \rho^{-1}) - 1\right)} = \frac{M}{\rho^{n+1} - \rho^{-n-1}} \frac{\rho + \rho^{-1}}{\frac{1}{2}(\rho + \rho^{-1}) - 1}.$$

Теорема 1. *Пусть $f : [L, R] \to \mathbb{R}$ является следом аналитической в круге с диаметром $[L, R]$ функции $f(z)$, причем $|f(z)| \le M$ для всех z из этого круга.*

Тогда для произвольного $\varepsilon > 0$, каждой матрицы развертки $A_k \in \mathbb{R}^{b^k \times b^{d-k}}$ тензора $T_{b,d,[L,R]}(f)$ и любого натурального $\mu \in [1, b^k - 1]$ существует матрица $B_k \in \mathbb{R}^{b^k \times b^{d-k}}$ такая, что

$$\operatorname{rank} B_k \le 2\mu + s + 1 \quad \text{и} \quad \|A_k - B_k\|_\infty \le \varepsilon,$$

где $s = \lfloor \log_\rho(3M/\varepsilon + 1) \rfloor$, $\rho = 2\mu + 1 + 2\sqrt{\mu^2 + \mu}$.

Доказательство. Обозначим через h шаг дискретизации: $h = (R - L)b^{-d}$. Строка матрицы A_k с индексом j соответствует отрезку $[L + jh, L + (j + 1)h]$.

Рассмотрим функцию

$$g_j(y) = f\left(L + jh + \frac{1 + y}{2} h\right), \quad y \in [-1, 1],$$

а также соответствующую функцию комплексного аргумента $g_j(w)$, $w \in U((\rho + \rho^{-1})/2, 0)$.

Отображение $w \mapsto \alpha + \beta w$ переводит круг $U((\rho + \rho^{-1})/2, 0)$ в $U(\beta(\rho + \rho^{-1})/2, \alpha)$. В нашем случае $\alpha = L + jh + h/2$, $\beta = h/2$.

Покажем, что при $j = \mu, \mu + 1, \ldots, b^k - \mu - 1$ круг $U(\beta(\rho + \rho^{-1})/2, \alpha)$ лежит внутри круга с диаметром $[L, R]$, т.е.

$$L + jh + \frac{h}{2} - \frac{h}{4}(\rho + \rho^{-1}) \ge L,$$
$$L + jh + \frac{h}{2} + \frac{h}{4}(\rho + \rho^{-1}) \le R.$$

Действительно, для заданного в условии значения ρ верно равенство $\rho + \rho^{-1} = 4\mu + 2$, поэтому указанная пара условий переписывается в виде

$$\begin{cases} (j - \mu)h \ge 0, \\ L + (j + \mu + 1)h \le R, \end{cases} \quad \Leftrightarrow \quad \begin{cases} j \ge \mu, \\ j \le b^k - \mu - 1. \end{cases}$$

Внутри круга с диаметром $[L, R]$ по условию выполнено неравенство $|f(z)| \le M$, поэтому и $|g_j(w)| \le M$ для $w \in U((\rho + \rho^{-1})/2, 0)$, а значит, и для $w \in \Gamma_\rho$. Поэтому из леммы 1 следует, что для полинома Лагранжа $\hat{P}_{s,j}(y)$ степени s на чебышёвской сетке для $g(y)$ верно неравенство:

$$\begin{aligned} |g_j(y) - \hat{P}_{s,j}(y)| &\le \frac{\rho + \rho^{-1}}{\frac{1}{2}(\rho + \rho^{-1}) - 1} \frac{M}{\rho^{s+1} - \rho^{-s-1}} = \frac{4\mu + 2}{2\mu} \frac{M}{\rho^{s+1} - \rho^{-s-1}} \le \\ &\le \frac{3M}{\rho^{s+1} - 1} \le \frac{3M}{3M/\varepsilon + 1 - 1} = \varepsilon. \end{aligned} \tag{3}$$

Пусть $P_{s,j}(x) = \hat{P}_{s,j}\left(\frac{2}{h}(x - L - jh) - 1\right)$. Из неравенства (3) получаем, что $|f(x) - P_{s,j}(x)| \le \varepsilon$ на отрезке $[L + jh, L + (j+1)h]$ для $j = \mu, \mu + 1, \ldots, b^k - \mu - 1$. Поэтому строки a_j^{T} с этими индексами приблизим строками:

$$\mathbf{P}_{s,j}^{\mathrm{T}} = [P_{s,j}(L + h(j + \ell b^{k-d}))]_{\ell=0}^{b^{d-k}-1}.$$

Так как степень полиномов $P_{s,j}(x)$ не превосходит s, все строки $\mathbf{P}_{s,j}^{\mathrm{T}}$ лежат в линейной оболочке векторов $p_0, \ldots, p_s \in \mathbb{R}^{d-k}$, $p_t(\ell) = \ell^t$.

Строки a_j^{T} матрицы A_k с индексами $j \in \{0, \ldots, \mu - 1\} \cup \{b^k - \mu, \ldots, b^k - 1\}$ “приблизим” ими самими, т.е. матрицу B_k построим в виде

$$\left[a_0^{\mathrm{T}}, \ldots, a_{\mu-1}^{\mathrm{T}}, \mathbf{P}_{s,\mu}^{\mathrm{T}}, \ldots, \mathbf{P}_{s,b^k-\mu-1}^{\mathrm{T}}, a_{b^k-\mu}^{\mathrm{T}}, \ldots, a_{b^k-1}^{\mathrm{T}}\right]^{\mathrm{T}}.$$

Очевидно, $\operatorname{rank} B_k \le 2\mu + s + 1$ и $\|A_k - B_k\|_\infty \le \varepsilon$.

Следствие 1. В условиях теоремы 1 для любого $\hat{\varepsilon} > 0$ выполнено

$$\mathcal{R}_{\hat{\varepsilon}}(T_{b,d,[L,R]}(f)) \le \hat{s} + 3, \quad \text{где} \quad \hat{s} = \left\lfloor \log_\rho(3Mb^{d/2}\sqrt{d-1}\hat{\varepsilon}^{-1} + 1) \right\rfloor, \quad \rho = 3 + 2\sqrt{2}.$$

Доказательство. Положим в теореме 1 $\mu = 1$ (чтобы удовлетворить условию $\mu \in [1, b^k - 1]$ для всех $k = 1, \ldots, d-1$) и

$$\varepsilon = \frac{\hat{\varepsilon}}{b^{d/2}\sqrt{d-1}}$$

и рассмотрим приближения B_k к матрицам развертки A_k со свойствами $\operatorname{rank} B_k \le \hat{s} + 3$ и $\|A_k - B_k\|_\infty \le \varepsilon$. Из последнего неравенства следует, что $\|A_k - B_k\|_F \le \varepsilon b^{d/2}$.

Теорема 2.2 из [8] гарантирует существование тензора $\widehat{B}$ с TT-рангами r_k, приближающего тензор A в норме Фробениуса с точностью:

$$\sqrt{\sum_{k=1}^{d-1} \varepsilon_k^2},$$

где r_k суть ε_k-ранги матриц развертки A_k.

Положим $\varepsilon_k = \varepsilon b^{d/2}$, тогда $r_k \le \operatorname{rank} B_k \le \hat{s} + 3$ и при этом

$$\left\|A - \widehat{B}\right\|_F \le \sqrt{d-1}\varepsilon b^{d/2} = \hat{\varepsilon}.$$

Теорема 2. *Пусть $\hat{\Gamma}$ – образ эллипса Γ_ρ под действием линейного отображения*

$$\xi(w) = \frac{R + L}{2} + \frac{R - L}{2} w,$$

а $f(z)$ – аналитическая внутри $\hat{\Gamma}$ функция, для которой $|f(z)| \le M$ для $z \in \hat{\Gamma}$ и имеющая вещественнозначный след на $[L, R]$.

Тогда для произвольного $\varepsilon > 0$ *и натуральных* b *и* d *существует тензор* B *такой, что*

$$\mathcal{R}(B) \le \lfloor \log_\rho(3M/\varepsilon + 1) \rfloor, \quad \left\| T_{b,d,[L,R]}(f) - B \right\|_\infty \le \varepsilon.$$

Доказательство. Будем обозначать через $(f \circ \xi)(x)$ композицию функций f и ξ, т.е. $(f \circ \xi)(x) = f(\xi(x))$. Заметим, что $(f \circ \xi)(z)$ аналитична в Γ_ρ как композиция аналитических функций f и ξ. Приблизим $(f \circ \xi)(x)$ полиномом Лагранжа $\hat{P}_s(x)$ степени s на чебышёвской сетке на $[-1,1]$, $(f \circ \xi)(x)$ – след аналитической в Γ_ρ функции $(f \circ \xi)(z)$. Аналогично доказательству теоремы 1 можно показать, что для параметра s из условия выполнено неравенство:

$$\left|(f \circ \xi)(x) - \hat{P}_s(x)\right| \le \varepsilon \quad \forall x \in [-1,1], \;\Rightarrow\; \left|(f \circ \xi)(\xi^{-1}(y)) - \hat{P}_s(\xi^{-1}(y))\right| \le \varepsilon \quad \forall y \in [L,R], \;\Rightarrow$$
$$\Rightarrow \; \left|f(y) - (\hat{P}_s \circ \xi^{-1})(y)\right| \le \varepsilon \quad \forall y \in [L,R],$$

где $(\hat{P}_s \circ \xi^{-1})(y)$ есть полином степени не более s, поэтому согласно [3], [2] его тензоризация B допускает ТТ-разложение с ТТ-рангами, не превосходящими $s+1$.

4. НИЖНИЕ ОЦЕНКИ

Рассмотрим пример функции $f(x)$, являющейся следом аналитической в $\mathbb{C}$ функции. Для приближенных ТТ-рангов тензоризации этой функции будет доказана нижняя оценка (теорема 3).

Зафиксируем натуральные $b \ge 2$, $d \ge 2$ и $k \ge d/3$. Определим $f(z) = \sin(2\pi b^{2k} z^2)$, $z \in \mathbb{C}$, и обозначим через $f(x)$ ее (вещественнозначный) след на $\mathbb{R}$. Пусть $A = T_{b,d,[0,1]}(f)$.

Введем в рассмотрение функции:

$$f_s(\xi) = f\left(\left(s + \frac{\xi}{2\pi}\right) b^{-k}\right), \quad \xi \in [0, 2\pi], \quad s \in \{0, \ldots, b^k - 1\}.$$

Строка матрицы развертки A_k с индексом s есть дискретизация функции $f_s(\xi)$ на равномерной сетке на $[0, 2\pi]$.

Для всех $s,t \in \{0, \ldots, b^k - 1\}$ определим интегралы

$$d_{s,t} = \frac{1}{\pi} \int_0^{2\pi} f_s(\xi) f_t(\xi) d\xi.$$

Для тех же s,t введем величины (здесь $a_{s,i}$ – элементы матрицы A_k):

$$\hat{d}_{s,t} = \frac{2\pi}{b^{d-k}} \frac{1}{\pi} \sum_{i=0}^{b^{d-k}-1} a_{s,i} a_{t,i}. \tag{4}$$

Матрицу из элементов $\hat{d}_{s,t}$ обозначим через $\hat{D}$.

Лемма 2. *Для всех* $s,t \in \{0, \ldots, b^k - 1\}$ *верно неравенство*

$$\left|d_{s,t} - \delta_{s,t}\right| \le \frac{2}{(s+t)^2},$$

где $\delta_{s,t}$ *– символ Кронекера.*

Доказательство. Преобразуем выражения для $f_s(\xi)$:

$$f_s(\xi) = \sin\left(2\pi b^{2k}\left(s^2 + \frac{s\xi}{\pi} + \frac{\xi^2}{4\pi^2}\right) b^{-2k}\right) = \sin\left(2s\xi + \frac{1}{2\pi}\xi^2\right).$$

Отсюда получим

$$d_{s,t} = \frac{1}{\pi}\int_0^{2\pi} f_s(\xi) f_t(\xi) d\xi = \frac{1}{2\pi}\int_0^{2\pi} \cos(2(s-t)\xi) - \cos\left(2(s+t)\xi + \frac{1}{\pi}\xi^2\right) d\xi =$$

$$= \frac{1}{2\pi}\begin{cases} 2\pi - \int_0^{2\pi} \cos\left(4s\xi + \frac{1}{\pi}\xi^2\right) d\xi, & \text{если } s = t, \\ -\int_0^{2\pi} \cos\left(2(s+t)\xi + \frac{1}{\pi}\xi^2\right) d\xi, & \text{если } s \neq t. \end{cases} \tag{5}$$

Пусть $\beta = 1/\pi$, $p = 2(s+t)$. Исследуем интеграл $\int_0^{2\pi} e^{ip\xi + i\beta\xi^2} d\xi$:

$$\int_0^{2\pi} e^{ip\xi + i\beta\xi^2} d\xi = \int_0^{2\pi} \frac{1}{ip} e^{i\beta\xi^2} de^{ip\xi} = \frac{1}{ip}\left(e^{i\beta\xi^2} e^{ip\xi}\Big|_0^{2\pi} - \int_0^{2\pi} i\beta 2\xi e^{ip\xi} e^{i\beta\xi^2} d\xi \right).$$

Первый член в скобках равен 0, так как $e^{i\beta(2\pi)^2} = e^{i4\pi} = 1 = e^{ip2\pi}$. Повторяя интегрирование по частям, получаем

$$\int_0^{2\pi} e^{ip\xi + i\beta\xi^2} d\xi = \frac{-2i\beta}{ip}\int_0^{2\pi} \xi e^{ip\xi} e^{i\beta\xi^2} d\xi = \frac{-2\beta}{p}\frac{1}{ip}\int_0^{2\pi} \xi e^{i\beta\xi^2} de^{ip\xi} =$$

$$= \frac{-2\beta}{ip^2}\left(\xi e^{i\beta\xi^2} e^{ip\xi}\Big|_0^{2\pi} - \int_0^{2\pi} e^{ip\xi} e^{i\beta\xi^2} (1 + i\beta 2\xi^2) d\xi \right) = \frac{-2\beta}{ip^2}\left(2\pi - \int_0^{2\pi} e^{ip\xi} e^{i\beta\xi^2} (1 + i\beta 2\xi^2) d\xi \right).$$

Взяв вещественную часть от обеих частей полученного равенства, имеем

$$\int_0^{2\pi} \cos(p\xi + \beta\xi^2) d\xi = -\frac{2\beta}{p^2}\operatorname{Im}\int_0^{2\pi} e^{ip\xi} e^{i\beta\xi^2} (1 + i\beta 2\xi^2) d\xi.$$

Поэтому верно

$$\left| \int_0^{2\pi} \cos(p\xi + \beta\xi^2) d\xi \right| \leq \frac{2\beta}{p^2}\int_0^{2\pi} (1 + 2\beta\xi^2) d\xi = \frac{4\beta\pi}{p^2} + \frac{4\beta^2}{3p^2} 8\pi^3 = \frac{1}{p^2}\left(4 + \frac{32\pi}{3}\right).$$

Отсюда и из (5) следует, что

$$|d_{s,t} - \delta_{s,t}| \leq \frac{1}{8\pi(s+t)^2}\left(4 + \frac{32\pi}{3}\right) \leq \frac{2}{(s+t)^2}.$$

Лемма 3. *Для всех $s,t \in \{0,\ldots,b^k - 1\}$ верно неравенство*

$$|\hat{d}_{s,t} - d_{s,t}| \leq 2\pi b^{k-d}(2s + 2t + 4).$$

Доказательство. Напомним, что *формулой левых прямоугольников* для вычисления интеграла $\int_{x_i}^{x_{i+1}} \varphi(x) dx$ называется выражение $(x_{i+1} - x_i)\varphi(x_i)$, а составной формулой для сетки $\{x_0,\ldots,x_n\}$ с шагом h называется сумма $\sum_{i=0}^{n-1} \varphi(x_i)h$. Простая формула имеет погрешность:

$$\left| \int_{x_i}^{x_{i+1}} \varphi(x) dx - \int_{x_i}^{x_{i+1}} \varphi(x_i) dx \right| \leq \int_{x_i}^{x_{i+1}} |\varphi'(\xi(x))| (x - x_i) dx \leq \|\varphi'\|_{C[x_i, x_{i+1}]} \frac{1}{2}(x_{i+1} - x_i)^2,$$

где $\xi(x) \in [x_i, x_{i+1}]$. При применении составной формулы на отрезке $[L, R]$ с шагом h погрешность не превосходит $\|\varphi'\|_{C[L,R]}(R - L)h/2$.

Ясно видно, что выражение (4) для $\hat{d}_{s,t}$ есть в точности составная формула левых прямоугольников для вычисления интеграла $d_{s,t}$ на отрезке $[0, 2\pi]$. Погрешность аппроксимации интеграла оценивается (подразумевается норма $\|\cdot\|_{C[0,2\pi]}$) в виде

$$\left|\hat{d}_{s,t} - d_{s,t}\right| \le \left\|\frac{d}{d\xi}\left(\frac{1}{\pi} f_s \cdot f_t\right)\right\| 2\pi \frac{2\pi}{2b^{d-k}} = 2\pi b^{k-d} \|f_s' f_t + f_s f_t'\| \le$$
$$\le 2\pi b^{k-d}(\|f_s'\| + \|f_t'\|) \le 2\pi b^{k-d}\left(2s + \frac{2\pi}{\pi} + 2t + \frac{2\pi}{\pi}\right) = 2\pi b^{k-d}(2s + 2t + 4).$$

Лемма 4. *Пусть F — главная подматрица матрицы $\hat{D}$, находящаяся на пересечении строк и столбцов с индексами $s, t \in [\ell, u)$, где*

$$u = \left\lfloor b^{\frac{d-k}{2}}/8 \right\rfloor, \quad \ell = \left\lceil b^{\frac{d-k}{2}}/16 \right\rceil,$$

причем $b^{d-k} \ge 2^{10}$. Тогда младшее сингулярное число $\sigma_{\min}(F) > 1/2$.

Доказательство. Обратим внимание, что заявленное в начале раздела условие $k \ge d/3$ гарантирует, что $\ell, u < b^k$, поэтому выбирать подматрицу с указанными индексами в матрице $\hat{D} \in \mathbb{R}^{b^k \times b^k}$ корректно.

Пусть $G = F - I$. Заметим, что

$$\sigma_{\min}(F) = \sigma_{\min}(I + G) = \left\|(I + G)^{-1}\right\|_2^{-1} = \left\|I - G + G^2 - G^3 + \ldots\right\|_2^{-1} \ge$$
$$\ge (\|I\|_2 + \|G\|_2 + \|G\|_2^2 + \ldots)^{-1} = \left(\frac{1}{1 - \|G\|}\right)^{-1} = 1 - \|G\|_2. \tag{6}$$

Из лемм 2 и 3 следует, что все элементы матрицы $\hat{D} - I$, а значит, и $F - I$, по модулю не превосходят

$$\frac{2}{(s+t)^2} + 2\pi b^{k-d}(2s + 2t + 4).$$

Поэтому имеем

$$\|G\|_2 = \|F - I\|_2 \le \|F - I\|_F \le (u - \ell)\|F - I\|_\infty \le$$
$$\le (u - \ell)\left(\frac{2}{(s+t)^2} + 2\pi b^{k-d}(2s + 2t + 4)\right) \le \frac{b^{\frac{d-k}{2}}}{16}\left(\frac{2}{4\ell^2} + \frac{2\pi \cdot 4u}{b^{d-k}}\right) \le \frac{256}{32b^{\frac{d-k}{2}}} + \frac{8\pi}{128}.$$

По условию $b^{d-k} \ge 2^{10}$, тогда получаем

$$\|G\|_2 \le \frac{1}{4} + \frac{\pi}{16} < \frac{1}{2}.$$

Отсюда и из неравенства (6) следует утверждение леммы.

Теорема 3. *Пусть $b \ge 2$, $d \ge 2$ и $k \ge d/3$ — натуральные числа, удовлетворяющие условию $b^{d-k} \ge 2^{10}$, а $f(x) = \sin(2\pi b^{2k} x^2)$, $x \in [0,1]$. Тогда верно неравенство*

$$\mathcal{R}_\varepsilon\left(T_{b,d,[0,1]}(f)\right) \ge \frac{1}{16} b^{\frac{d-k}{2}} - 2, \quad \text{где} \quad \varepsilon = \frac{1}{2} b^{\frac{d-k}{2}}.$$

Доказательство. Исследуем сингулярные числа $\sigma_s(A_k)$ матрицы A_k. Для этого воспользуемся известным фактом, что

$$\sigma_s(A_k) = \sqrt{\lambda_s(A_k^{\mathrm{T}} A_k)} = \sqrt{\sigma_s(A_k^{\mathrm{T}} A_k)}. \tag{7}$$

Из определения (4) матрицы $\hat{D}$ следует, что $A_k^{\mathrm{T}} A_k = b^{d-k}\hat{D}/2$. Далее, для главной подматрицы F матрицы $\hat{D}$ из леммы можно применить теорему о чередовании сингулярных чисел симметричной матрицы [9] и получить, что $\sigma_{u-\ell}(\hat{D}) \geq 1/2$. Отсюда и из равенства (7) вытекает

$$\sigma_{u-\ell}(A_k) \geq \frac{1}{\sqrt{2}} b^{\frac{d-k}{2}} \sqrt{\sigma_{u-\ell}(\hat{D})} > \frac{1}{2} b^{\frac{d-k}{2}}.$$

Рассмотрим теперь произвольный тензор B со свойством $\|A - B\|_F \leq \frac{1}{2} b^{(d-k)/2}$. Аналогичное неравенство верно и для матриц развертки A_k и B_k. По теореме Эккарта–Юнга [10] наилучшее приближение в норме Фробениуса ранга $u - \ell - 1$ имеет погрешность как минимум

$$\sqrt{\sigma_{u-\ell}^2(A_k) + \ldots + \sigma_N^2(A_k)} \geq \sigma_{u-\ell}(A_k) > \frac{1}{2} b^{\frac{d-k}{2}},$$

где $N = \min\{b^k, b^{d-k}\}$ – меньший из размеров матрицы A_k. Поэтому верно

$$\operatorname{rank} B_k \geq u - \ell \geq \left(\frac{1}{8} b^{\frac{d-k}{2}} - 1\right) - \left(\frac{1}{16} b^{\frac{d-k}{2}} - 1\right) = \frac{1}{16} b^{\frac{d-k}{2}} - 2.$$

Рассмотрим произвольное TT-разложение тензора B вида (1). Для k-й матрицы развертки можно написать

$$B_k(i_1, \ldots, i_k; i_{k+1}, \ldots, i_d) = \sum_{\alpha_k=1}^{r_k} H'(i_1, \ldots, i_k, \alpha_k) H''(\alpha_k, i_{k+1}, \ldots, i_d),$$

откуда следует, что $r_k \geq \operatorname{rank} B_k \geq \frac{1}{16} b^{\frac{d-k}{2}} - 2$, а в силу произвольности B и

$$\mathcal{R}_\varepsilon(A) \geq \frac{1}{16} b^{\frac{d-k}{2}} - 2.$$

Чтобы наглядно связать верхнюю оценку из данного раздела и нижние оценки из предыдущего, имеет смысл зафиксировать некоторые величины и перейти к асимптотической нотации. Именно, зафиксируем натуральное $b \geq 2$, произвольно малое $\varepsilon > 0$ и отрезок $[L, R]$ на вещественной прямой. Определим функцию

$$\mathcal{R}_{b,\varepsilon,[L,R]}(M, d) = \max_{|f(z)| \leq M \text{ на } U} \mathcal{R}_\varepsilon(T_{b,d,[L,R]}(f)).$$

Здесь максимум берется по всем аналитическим в круге U с диаметром $[L, R]$ функциям $f(z)$, имеющим вещественный след на $[L, R]$.

Следствие 1 гарантирует оценку $\mathcal{R}_{b,\varepsilon,[L,R]}(M, d) = O(\ln M + d)$ при $M, d \to \infty$. С другой стороны, для функции $f(z)$ из данного раздела несложно оценить максимум модуля на единичном круге:

$$|f(z)| = \left|\sin(2\pi b^{2k} z^2)\right| = \left|\frac{1}{2i}\left(e^{2\pi i b^{2k} z^2} - e^{-2\pi i b^{2k} z^2}\right)\right| \leq e^{2\pi b^{2k}|z^2|} \leq e^{2\pi b^{2k}}. \tag{8}$$

Пусть $k = \lceil d/3 \rceil$, тогда по теореме 3 максимальный TT-ранг есть как минимум:

$$\frac{1}{16} b^{\frac{d-k}{2}} - 2 = \Omega(b^k) = \Omega\left(\sqrt{\ln M(d)}\right) \quad \text{при} \quad d \to \infty.$$

В последнем равенстве через $M(d)$ обозначено число $e^{2\pi b^{2d/3}}$. Получается, что

$$\mathcal{R}_{b,\varepsilon,[L,R]}(M(d), d) = \Omega\left(\sqrt{\ln M(d)}\right) \quad \text{при} \quad d \to \infty.$$

5. ПРИМЕНЕНИЕ К ПОЛИНОМАМ

В работе [5] рассматривались TT-ранги приближений к тензоризациям полиномов. В данном разделе мы применим результаты предыдущих разделов к полиномам, существенно улучшив

верхние оценки из указанной работы. Также мы получим нижние оценки ТТ-рангов ε-приближений для этого класса функций.

Утверждение 1. *Пусть* $p(x) = p_0 + p_1 x + \ldots + p_n x^n$ *— полином с вещественными коэффициентами. Пусть* $A = T_{b,d,[0,1]}(f)$ *и* $M = \sum_{i=0}^{n} |p_i|$. *Зафиксируем произвольное* $\varepsilon > 0$.

Тогда для каждой матрицы развертки A_k *тензора* A, $k = 1, \ldots, d-1$, *и натурального* $\mu \in [1, b^k - 1]$ *существует матрица* B_k *такая, что*

$$\operatorname{rank} B_k \le 2\mu + s + 1 \quad \text{и} \quad \|A_k - B_k\|_\infty \le \varepsilon,$$

где

$$s = \lfloor \log_\rho(3M/\varepsilon + 1) \rfloor, \quad \rho = 2\mu + 1 + 2\sqrt{\mu^2 + \mu}.$$

Доказательство. Достаточно заметить, что на единичном круге на комплексной плоскости (а тем более, в круге с диаметром $[0,1]$) аналитическое продолжение $p(z)$ полинома $p(x)$ ограничено по модулю суммой $\sum_i |p_i|$, и применить теорему 1.

Замечание. Если в духе работы [5] ограничить коэффициенты p_i по модулю фиксированной константой и поинтересоваться асимптотическим поведением ε-рангов матриц развертки при стремлении n к бесконечности, то доказанное утверждение даст оценку вида $O(\ln n)$ в отличие от оценки $O(\sqrt[3]{n})$, полученной в [5].

Утверждение 2. *Пусть* $b \ge 2$, $d \ge 2$ и $k \ge d/3$ *— натуральные числа, удовлетворяющие условию* $b^{d-k} \ge 2^{10}$.

Тогда для любого $n \ge \left\lfloor \log_2(1 + 14 b^{k/2} e^{10 b^{2k}}) \right\rfloor$ *существует полином* $P_n(x) = p_0 + \ldots + p_n x^n$ *такой, что верно следующее*:

1. $\sum_{i=0}^{n} |p_i| \le 2^{2n} (n+1)^2$;

2. $\mathcal{R}_\varepsilon(T_{b,d,[0,1]}(P_n)) \ge \frac{1}{16} b^{\frac{d-k}{2}} - 2$, *где* $\varepsilon = \frac{1}{4} b^{\frac{d-k}{2}}$.

Доказательство. Возьмем функцию $f(x)$ из теоремы 3. Обозначим через $P_n(x)$ полином Лагранжа степени n, интерполирующий $f(x)$ на чебышёвской сетке с $n+1$ узлом на $[-1,1]$.

Сначала оценим коэффициенты $p(x)$. Полином Лагранжа можно записать в следующем виде [9]:

$$P_n(x) = \sum_{j=0}^{n} \frac{f(x_j)\omega(x)}{(x - x_j)\omega'(x_j)}, \quad \omega(x) = (x - x_0) \cdots (x - x_n),$$

где x_j суть корни полинома Чебышёва степени $n+1$, т.е.

$$x_j = \cos\left(\frac{\pi}{2(n+1)} + \frac{\pi}{n+1} j\right), \quad \omega(x) = 2^{-n} \cos((n+1) \arccos x).$$

Оценим снизу модуль $\omega'(x_j)$:

$$\omega'(x_j) = 2^{-n}(n+1) \frac{1}{\sqrt{1 - x_j^2}} \sin((n+1) \arccos x_j) = \frac{2^{-n}(n+1)}{\sin\left(\frac{\pi}{2(n+1)} + \frac{\pi}{n+1} j\right)} \sin\left(\frac{\pi}{2} + \pi j\right).$$

Поэтому $|\omega'(x_j)| \ge 2^{-n}(n+1)$. Коэффициент при x^m многочлена $\omega(x)/(x - x_j)$ есть, очевидно,

$$\sum_{\substack{0 \le j_1 < \ldots < j_{n-m} \le n \\ j_\ell \ne j}} (-1)^{n-m} x_{j_1} \cdots x_{j_{n-m}},$$

т.е. по модулю не превосходит C_n^{n-m}. Итого получаем

$$\sum_{m=0}^{n}|p_m| \le \sum_{j=0}^{n}\left|\frac{f(x_j)}{\omega'(x_j)}\right|\sum_{m=0}^{n}C_n^{n-m} \le (n+1)2^n(n+1)2^n = 2^{2n}(n+1)^2.$$

Применим лемму 1 для оценки отклонения $f(x)$ от $P_n(x)$ на $[-1,1]$:

$$|f(x)-P_n(x)| \le \frac{M}{\rho^{n+1}-\rho^{-n-1}}\frac{\rho+\rho^{-1}}{\frac{1}{2}(\rho+\rho^{-1}-1)}, \quad M = \max_{z\in\Gamma_\rho}|f(z)|.$$

Положим $\rho = 2$ и оценим величину M, исходя из рассуждения, аналогичного (8):

$$M \le e^{2\pi\left(\frac{\rho+\rho^{-1}}{2}\right)^2 b^{2k}} \le e^{10b^{2k}}.$$

Таким образом, с учетом неравенства для n (из условия утверждения) имеем для всех $x \in [-1,1]$:

$$|f(x)-P_n(x)| \le \frac{e^{10b^{2k}}}{14b^{k/2}e^{10b^{2k}}}\frac{2.5}{0.75} \le \frac{1}{4}b^{-k/2}.$$

Это означает, что

$$\left\|T_{b,d,[0,1]}(f)-T_{b,d,[0,1]}(P_n)\right\|_F \le \frac{1}{4}b^{\frac{d-k}{2}} = \varepsilon.$$

Теперь возьмем произвольный тензор B, приближающий $T_{b,d,[0,1]}(P_n)$ с ошибкой не более ε. Заметим, что по неравенству треугольника для нормы он дает приближение $T_{b,d,[0,1]}(f)$ с ошибкой не более 2ε. Из теоремы 3 сразу получаем неравенство

$$\mathcal{R}(B) \ge \frac{1}{16}b^{\frac{d-k}{2}} - 2,$$

а в силу произвольности B и свойство из условия.

Автор выражает благодарность Виктории Владимировне Высоцкой за помощь в оформлении статьи.

СПИСОК ЛИТЕРАТУРЫ

1. *Oseledets I.* Constructive representation of functions in low-rank tensor formats // Constructive Approximat. 2013. V. 37. № 1. P. 1–18.
2. *Grasedyck L.* Polynomial approximation in hierarchical Tucker format by vector–tensorization. Marburg: Philipps-Universität, 2010.
3. *Khoromskij B. O*($d \log N$)-Quantics approximation of N-d tensors in high-dimensional numerical modeling // Constructive Approximat. 2011. V. 34. № 2. P. 257–280.
4. *Tyrtyshnikov E.E.* Tensor approximations of matrices generated by asymptotically smooth functions // Mat. Sb. 2003. V. 194. I. 6. P. 941–954.
5. *Vysotsky L.* On tensor-train ranks of tensorized polynomials // Large-Scale Scientific Computing. LSSC 2019. Lect. Notes in Comp. Sc. 2020. V. 11958. P. 189–196.
6. *Oseledets I.* Approximation of $2^d \times 2^d$ matrices using tensor decomposition // SIAM Journal on Matrix Anal. and Appl. 2009. V. 31. № 4. P. 2130–2145.
7. *Oseledets I.* Tensor-train decomposition // SIAM Journal on Scientific Comp. 2011. V. 33. № 5. P. 2295–2317.
8. *Oseledets I., Tyrtyshnikov E.* TT-cross approximation for multidimensional arrays // Linear Algebra and Its Appl. 2010. V. 432. № 1. P. 70–88.
9. *Тыртышников Е.Е.* Методы численного анализа. М.: Академия, 2007.
10. *Eckart C., Young G.* The approximation of one matrix by another of lower rank // Psychometrika. 1936. V. 1. № 3. P. 211–218.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 787–799

ОБЩИЕ ЧИСЛЕННЫЕ МЕТОДЫ

УДК 519.614

НОВЫЕ АЛГОРИТМЫ ДЛЯ РЕШЕНИЯ НЕЛИНЕЙНОЙ ПРОБЛЕМЫ СОБСТВЕННЫХ ЗНАЧЕНИЙ

© 2021 г. В. Гандер[1,2]

[1] *8092 Zurich, Ramistrasse 1010, ETH, Switzerland*

[2] *Hong Kong Baptist University, 224 Waterloo Rd, Kowloon Tong, Hong Kong*

**e-mail: gander@inf.ethz.ch*

Поступила в редакцию 24.12.2020 г.
Переработанный вариант 24.12.2020 г.
Принята к публикации 14.01.2021 г.

Для решения нелинейной проблемы собственных значений предлагаются алгоритмы, использующие методы третьего порядка для вычисления нулей уравнения $\det A(\lambda) = 0$. Производные определителя вычисляются с помощью алгоритмического дифференцирования. Специальные алгоритмы представлены в случае ленточных матриц. Библ. 11. Фиг. 6. Табл. 2.

Ключевые слова: нелинейная проблема собственных значений, методы третьего порядка, алгоритмическое дифференцирование.

DOI: 10.31857/S0044466921050094

1. ВВЕДЕНИЕ

Пусть $A\colon \lambda \mapsto \mathbb{C}^{n\times n}$ – аналитическое отображение на открытой области $\{\lambda\} \subset \mathbb{C}$. Наша задача: найти λ такое, что $f(\lambda) = \det A(\lambda) = 0$.

Для вычисления нуля функции f можно рассмотреть метод Ньютона:

$$\lambda_{k+1} = \lambda_k - \frac{f(\lambda_k)}{f'(\lambda_k)},$$

использующий производные определителя. *Формула Якоби*, широко известная и обсуждаемая в учебниках линейной алгебры, дает для них явное выражение

$$f'(\lambda) = \det A(\lambda)\,\mathrm{tr}(A^{-1}(\lambda)A'(\lambda)).$$

Ньютоновская коррекция получает вид

$$\frac{f(\lambda_k)}{f'(\lambda_k)} = \frac{1}{\mathrm{tr}(A^{-1}(\lambda_k)A'(\lambda_k))}.$$

Альтернативным подходом к вычислению производных, а значит, и ньютоновской коррекции, является алгоритмическое дифференцирование (см. [1]–[3]).

2. ВЫЧИСЛЕНИЕ ОПРЕДЕЛИТЕЛЕЙ И НЬЮТОНОВСКОЙ КОРРЕКЦИИ

При получении определителя обычно строится LU-разложение по методу Гаусса. Пусть $PA = LU$, где P включает перестановки, связанные с частичным выбором ведущего элемента, L – нижняя унитреугольная матрица и U – верхняя треугольная матрица. Тогда

$$\det(A) = \pm u_{11}u_{22}\cdots u_{nn}.$$

Когда LU-разложение записывается на месте матрицы A, текущее значение определителя на k-м шаге исключения умножается на ведущий элемент: $f := f \times a_{kk}$. Это довольно быстро ведет к появлению очень больших или очень малых чисел. Поэтому лучше перейти к логарифмам: $\log f := \log f + \log(a_{kk})$.

Заметим, что производная логарифма

$$\frac{d}{d\lambda}\log f(\lambda) = \frac{f'(\lambda)}{f(\lambda)}$$

является обратной величиной к ньютоновской коррекции. Таким образом, если производная логарифма вычисляется алгоритмическим дифференцированием

$$\text{logf} := \text{logf} + \log(a_{kk}), \quad \Rightarrow \quad \text{logfp} := \text{logfp} + \frac{a'_k}{a_{kk}},$$

то обратная величина ffp = 1/logfp = $f(\lambda)/f'(\lambda)$ – в точности нужная нам ньютоновская коррекция. Это наблюдение используется в следующей программе (см. [2]):

```
function ffp=deta(A,Ap)
% DETA compute determinant of A and derivative
% Given A=A(lambda) and Ap=A'(lambda), DETA(A,Ap)
% computes Newton correction ffp=f/f' where f=det(A).
n=length(A); logfp=0;
for j=1:n
    [amax,kmax]= max(abs(A(j:n,j)));        % partial pivoting
    if amax == 0,ffp=0; return, end
    kmax=kmax+j-1;
    if kmax ~= j                            % interchange rows
       h=Ap(kmax,:); Ap(kmax,:)=Ap(j,:); Ap(j,:)=h;
       h=A(j,:); A(j,:)=A(kmax,:); A(kmax,:)=h;
    end
    logfp=logfp + Ap(j,j)/A(j,j);
    Ap(j+1:n,j)=(Ap(j+1:n,j)*A(j,j)-A(j+1:n,j)*Ap(j,j))/A(j,j)^2;
    A(j+1:n,j)=A(j+1:n,j)/A(j,j);
    Ap(j+1:n,j+1:n)=Ap(j+1:n,j+1:n) - Ap(j+1:n,j)*A(j,j+1:n)- …
                      A(j+1:n,j)*Ap(j,j+1:n);
    A(j+1:n,j+1:n)=A(j+1:n,j+1:n) - A(j+1:n,j)*A(j,j+1:n);
end
ffp=1/logfp;
```

3. ПОДАВЛЕНИЕ ВМЕСТО ДЕФЛЯЦИИ

С помощью функции `deta` мы можем вычислить решение уравнения $\det A(\lambda) = 0$ по методу Ньютона. Чтобы избежать перевычисления уже найденных нулей $\lambda_1, \ldots, \lambda_k$, мы подавим их, работая с функцией

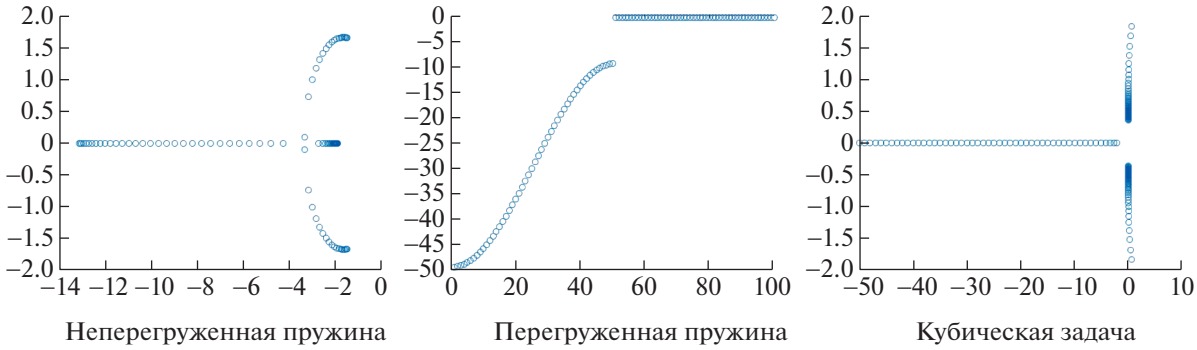
$$f_k(\lambda) := \frac{f(\lambda)}{p(\lambda)},$$

где $p(\lambda) = (\lambda - \lambda_1)\cdots(\lambda - \lambda_k)$. Тогда

$$p'(\lambda) = \sum_{j=1}^{k}\prod_{\substack{i=1\\ i\neq j}}^{k}(\lambda - \lambda_i) = p(\lambda)s(\lambda), \quad \text{где} \quad s(\lambda) = \sum_{j=1}^{k}\frac{1}{\lambda - \lambda_j}.$$

Производные для f_k имеют вид (мы опускаем аргумент λ)

$$f_k' = \frac{pf' - psf}{p^2} = \frac{f' - sf}{p}.$$



Фиг. 1. Примеры применения итераций Ньютона.

Ньютонова коррекция f_k/f_k', выраженная через f и f', приобретает вид

$$\frac{f_k}{f_k'} = \frac{f/p}{(f' - sf)/p} = \frac{f}{f' - sf} = \frac{f}{f'}\frac{1}{1 - \dfrac{f}{f'}s}. \tag{1}$$

В итоге получается итерация

$$\lambda_{j+1} = \lambda_j - \frac{f_k(\lambda_j)}{f_k'(\lambda_j)} = \lambda_j - \frac{f(\lambda_j)}{f'(\lambda_j)}\frac{1}{1 - \dfrac{f(\lambda_j)}{f'(\lambda_j)}\displaystyle\sum_{j=1}^{k}\frac{1}{\lambda - \lambda_j}},$$

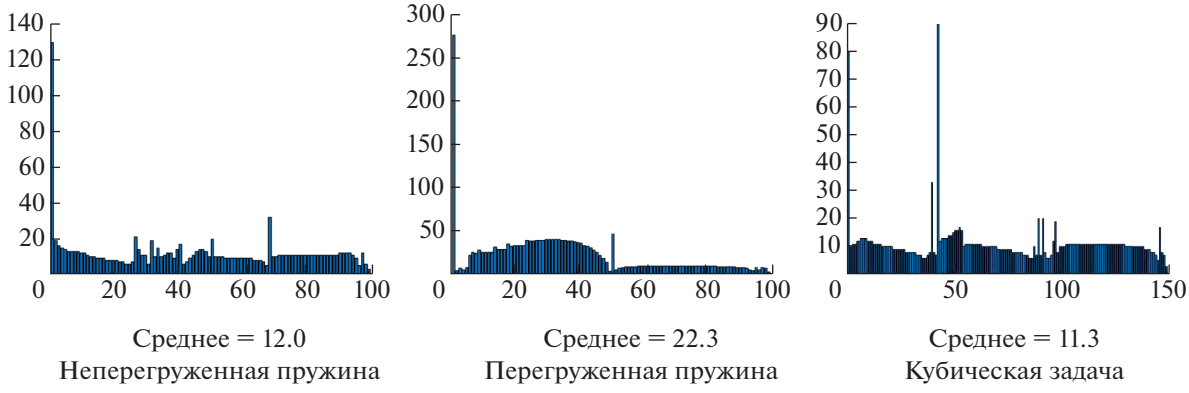
называемая итерацией *Ньютона–Мехли (Newton–Maehly)* (см. [4]).

В [2] мы привели расчеты для двух примеров масс с пружинами из [5] и для кубического примера из [1]. Вот MATLAB-программа для первого примера масс с пружинами:

```
n=50, tau=3, kappa=5,                    % nonoverdamped
e=-ones(n-1,1);
C=(diag(e,-1)+diag(e,1)+3*eye(n)); K=kappa*C; C=tau*C;
lam=-0.5+0.1*i; lamb=[];                 % start
for k=1:2*n
    ffp=1;
    while abs(ffp)>1e-14
        Qp=2*lam*eye(n)+C; Q=lam*(lam*eye(n)+ C)+K;
        ffp=deta(Q,Qp);
        s=sum(1./(lam-lamb(1:k-1)));
        lam=lam-ffp/(1-ffp*s);           % Newton step
    end
    lamb(k)=lam;
    lam=lam*(1+0.01*i);                  % start for next eigenvalue
end
plot(lamb,'o')
```

Итерация для первого собственного значения начинается с выбора случайного комплексного числа, здесь $\lambda_0 = -0.5 + 0.1i$. В качестве начального значения для следующего собственного значения мы выбираем последнее из найденных и подавляем значение λ_k с помощью малого возмущения: $\lambda_0 = \lambda_k(1 + i/100)$.

Аналогично проводятся вычисления для перегруженной (overdamped) пружины и кубической проблемы собственных значений для $n = 50$. Найденные собственные значения изображены на фиг. 1.



Фиг. 2. Число итераций.

Интересно посмотреть, сколько итераций нужно для каждого собственного значения. Столбцовые графики и средние числа итераций показаны на фиг. 2. Начальное значение для первого собственного значения, очевидно, выбрано не очень удачно, так как для сходимости требуется большое число итераций. Для кубической задачи большое число итераций возникает также для некоторых промежуточных собственных значений.

4. УЛУЧШЕНИЕ СХОДИМОСТИ

Причиной большого числа итерационных шагов при вычислении собственных значений в последних трех примерах является довольно плохая глобальная сходимость метода Ньютона. Локально метод Ньютона сходится к простому корню квадратично, обычно результат с машинной точностью получается за 3–4 итерации.

Пусть $f(z) = 0$ и λ_k – приближение к z. Ньютонова итерация заменяет f в окрестности λ_k на *линейную функцию* g такую, что $f(\lambda_k) = g(\lambda_k)$, $f'(\lambda_k) = g'(\lambda_k)$. Таким образом, g совпадает с отрезком ряда Тейлора $g(\lambda) = f(\lambda_k) + f'(\lambda_k)(\lambda - \lambda_k)$, а следующее приближение λ_{k+1} является нулем функции g.

Итерация Хэлли (Halley)

$$\lambda_{k+1} = \lambda_k - \frac{f(\lambda_k)}{f'(\lambda_k)} \frac{1}{1 - \frac{1}{2} \frac{f(\lambda_k) f''(\lambda_k)}{f'(\lambda_k)^2}} \tag{2}$$

заменяет f локально на *гиперболическую функцию*

$$g(\lambda) = \frac{a}{\lambda + b} + c$$

такую, что $f(\lambda_k) = g(\lambda_k)$, $f'(\lambda_k) = g'(\lambda_k)$ и $f''(\lambda_k) = g''(\lambda_k)$, следующее приближение λ_{k+1} является нулем функции g. Итерация Хэлли – метод третьего порядка, и значит, он сходится к простому корню кубически (см. [6]). Можно ожидать, что гиперболическая аппроксимация для f лучше с точки зрения глобальной сходимости.

5. РЕАЛИЗАЦИЯ ИТЕРАЦИИ ХЭЛЛИ

Нам нужна *вторая производная* определителя, более точно, нам нужно вычислять функцию

$$t(\lambda) = \frac{f(\lambda) f''(\lambda)}{f'^2(\lambda)}.$$

Заметим, что производная ньютоновской коррекции имеет вид

$$\frac{d}{dx}\left(\frac{f}{f'}\right) = \frac{f'^2 - ff''}{f'^2} = 1 - \frac{ff''}{f'^2}.$$

Отсюда

$$t = \frac{ff''}{f'^2} = 1 - \frac{d}{dx}\left(\frac{f}{f'}\right),$$

и чтобы получить $t(\lambda)$, нужно вычислять производную ньютоновской коррекции для нашей функции `deta`. Это делается алгоритмическим дифференцированием функции `deta`. Далее, функция `det2p` получает на вход матрицы A, A' и A'' и вычисляет ньютоновскую коррекцию f/f' и ее производную:

```
function [ffp,dffp] = det2p(A,Ap,App)
% DET2P computes Newton correction ffp = f/f'
%       and its derivative dffp = (f/f')'
n=length(A);
logfpp=0;                                % logfpp = log(f)''
logfp=0;                                 % log(f)'
for k=1:n
   [amax,kmax]=max(abs(A(k:n,k)));       % partial pivoting
   if amax==0                            % matrix singular
      ffp=0; dffp=0;return
   end
   kmax=kmax+k-1;
   if kmax~=k                            % interchange rows
     h=App(k,:); App(k,:)=App(kmax,:); App(kmax,:)=h;
     h=Ap(k,:); Ap(k,:)=Ap(kmax,:); Ap(kmax,:)=h;
     h=A(k,:); A(k,:)=A(kmax,:); A(kmax,:)=h;
   end
   logfpp=logfpp+(A(k,k)*App(k,k)-Ap(k,k)^2)/A(k,k)^2;
   logfp=logfp+Ap(k,k)/A(k,k);
   App(k+1:n,k)=(A(k,k)*App(k+1:n,k)-Ap(k+1:n,k)*Ap(k,k))/A(k,k)^2-…
        (Ap(k+1:n,k)*Ap(k,k)/A(k,k)^2+ …
        A(k+1:n,k)*App(k,k)/A(k,k)^2-…
        2* A(k+1:n,k)*Ap(k,k)^2/A(k,k)^3);

   Ap(k+1:n,k)=Ap(k+1:n,k)/A(k,k)-A(k+1:n,k)*Ap(k,k)/A(k,k)^2;
   A(k+1:n,k)=A(k+1:n,k)/A(k,k);         % elimination step

   App(k+1:n,k+1:n)=App(k+1:n,k+1:n) -…
        (App(k+1:n,k)*A(k,k+1:n) + Ap(k+1:n,k)*Ap(k,k+1:n) ) - …
        (Ap(k+1:n,k)*Ap(k,k+1:n) + A(k+1:n,k)*App(k,k+1:n));

   Ap(k+1:n,k+1:n)=Ap(k+1:n,k+1:n) - Ap(k+1:n,k)*A(k,k+1:n)-…
        A(k+1:n,k)*Ap(k,k+1:n);
   A(k+1:n,k+1:n)=A(k+1:n,k+1:n) - A(k+1:n,k)*A(k,k+1:n);
end
dffp=-logfpp/logfp^2; ffp=1/logfp;
```

6. ХЭЛЛИ–МЕХЛИ (HALLEY–MAEHLY)

Как и раньше, мы хотим подавить уже вычисленные собственные значения и снова рассматриваем

$$f_k(\lambda) : \frac{f(\lambda)}{p(\lambda)}, \quad p(\lambda) = (\lambda - \lambda_1) \cdots (\lambda - \lambda_k).$$

Теперь мы применяем итерации Хэлли к f_k:

$$\lambda_{\text{new}} = \lambda - \frac{f_k}{f_k'} \frac{1}{1 - \frac{1}{2} \frac{f_k f_k}{f_k'^2}},$$

и записываем итерацию в терминах f, f' и f''. Для ньютоновской коррекции f_k/f_k' используем уравнение (1). Для f_k''/f_k' выражаем сначала

$$f_k'' = \frac{d}{d\lambda}\left(\frac{f' - sf}{p}\right) = \frac{p(f'' - s'f - sf') - p'(f' - sf)}{p^2} =$$
$$= \frac{f'' - s'f - sf' - sf' + s^2 f}{p}, \quad p' = ps, \quad s = \sum_{j=1}^{k} \frac{1}{\lambda - \lambda_j}.$$

Затем, деля на f_k', получаем

$$\frac{f_k''}{f_k'} = \frac{f'' - s'f - 2sf' + s^2 f}{f' - sf} = \frac{\frac{f''}{f'} - s'\frac{f}{f'} - 2s + s^2 \frac{f}{f'}}{1 - s\frac{f}{f'}},$$

и после умножения на f_k/f_k' находим

$$t = \frac{f_k f_k''}{f_k'^2} = \frac{\frac{ff''}{f'^2} + (s^2 - s')\left(\frac{f}{f'}\right)^2 - 2s\frac{f}{f'}}{\left(1 - s\frac{f}{f'}\right)^2}. \tag{3}$$

Суммируя, реализуем итерацию Хэлли–Мехли, для этого необходимо следующее.

1. Вычислить ньютоновскую коррекцию для f_k:

$$\frac{f_k}{f_k'} = \frac{f}{f'} \frac{1}{1 - \frac{f}{f'}s}.$$

2. Вычислить $t(\lambda)$ для f_k в соответствии с уравнением (3).
3. Итерировать

$$\lambda_{\text{new}} = \lambda - \frac{f_k}{f_k'} \frac{1}{1 - \frac{1}{2}t}.$$

Мы решаем три нелинейные проблемы собственных значений с помощью метода Хэлли и сравниваем результаты с итерациями Ньютона (см. табл. 1). Глобальная сходимость и в самом деле улучшилась, число итераций уменьшилось.

7. ЛАГЕР И ОСТРОВСКИЙ

Еще один метод третьего порядка для вычисления нулей многочлена – это *метод Лагера*. В качестве аппроксимации многочлена f степени n он использует многочлен $g(\lambda) = a(\lambda - \lambda_1)(\lambda - \lambda_2)^{n-1}$. Параметры a, λ_1 и λ_2 определяются таким образом, что g интерполирует f вместе с производны-

Таблица 1. Сравнение методов Ньютона и Хэлли

Число итераций	Неперегруженная пружина	Перегруженная пружина	Кубическая задача
Ньютон:			
максимальное	128	275	90
среднее	11.4	20.9	11.3
Хэлли:			
максимальное	67	140	46
среднее	7	12.1	7.1

Таблица 2. Сравнение методов Хэлли, Лагера и Островского

Число итераций	Неперегруженная пружина	Перегруженная пружина	Кубическая задача
Хэлли:			
максимальное	67	140	46
среднее	7	12.1	7.1
Лагер:			
максимальное	18	36	16
среднее	5.3	6.6	5.2
Островский:			
максимальное	23	43	18
среднее	5.5	7.1	5.2

ми $f(\lambda_k) = g(\lambda_k)$, $f'(\lambda_k) = g'(\lambda_k)$, $f''(\lambda_k) = g''(\lambda_k)$. Следующее приближение – это нуль для g, ближайший к λ_k:

$$\lambda_{k+1} = \lambda_k - \frac{f(\lambda_k)}{f'(\lambda_k)} \frac{n}{1 + \sqrt{(n-1)^2 - n(n-1)\dfrac{f(\lambda_k)f''(\lambda_k)}{f'(\lambda_k)^2}}}. \tag{4}$$

Степень n является параметром метода Лагера. Пусть в (4) $n \to \infty$. Тогда мы получаем итерацию

$$\lambda_{k+1} = \lambda_k - \frac{f(\lambda_k)}{f'(\lambda_k)} \frac{1}{\sqrt{1 - \dfrac{f(\lambda_k)f''(\lambda_k)}{f'(\lambda_k)^2}}}, \tag{5}$$

которая называется *итерацией Островского с квадратным корнем*.

Заметим, что в итерационных методах Лагера и Островского так же, как и в методе Хэлли, нам нужно всего лишь два выражения:

$$\frac{f}{f'} \quad \text{и} \quad t = \frac{ff''}{f'^2}.$$

Поскольку метод Лагера задумывался как метод вычисления нулей многочлена, можно ожидать, что он будет хорошо работать для трех наших примеров. Сравнение трех методов в табл. 2 показывает, что это действительно так.

8. МЕТОДЫ ТРЕТЬЕГО ПОРЯДКА

Методы Хэлли, Лагера и Островского реализуют специальные случаи следующей теоремы.

Теорема 1 (методы третьего порядка, см. [6]). *Пусть s – простой нуль функции f, а G – любая функция такая, что*

$$G(0) = 1, \quad G'(0) = \frac{1}{2}, \quad |G''(0)| < \infty.$$

Тогда итерационный метод

$$x_{\text{new}} = x - \frac{f(x)}{f'(x)} G(t(x)), \quad t(x) = \frac{f(x)f''(x)}{f'(x)^2},$$

сходится к s по крайней мере кубически.

Примеры:

- формула Хэлли (Halley)

$$G(t) = \frac{1}{1 - \frac{1}{2}t} = 1 + \frac{1}{2}t + \frac{1}{4}t^2 + \frac{1}{8}t^3 + \ldots;$$

- формула Эйлера (Euler)

$$G(t) = \frac{2}{1 + \sqrt{1 - 2t}} = 1 + \frac{1}{2}t + \frac{1}{2}t^2 + \frac{5}{8}t^3 + \ldots;$$

- квадратичная обратная интерполяция

$$G(t) = 1 + \frac{1}{2}t;$$

- итерация Островского с квадратным корнем

$$G(t) = \frac{1}{\sqrt{1 - t}} = 1 + \frac{1}{2}t + \frac{3}{8}t^2 + \ldots;$$

- Лагер (Laguerre)

$$G(t) = \frac{n}{1 + \sqrt{(n-1)^2 - n(n-1)t}} 1 + \frac{1}{2}t + \frac{1}{8}\frac{3n-2}{n-1}t^2 + \ldots;$$

- семейство формул Хансен–Патрик (Hansen–Patrick) (см. [7])

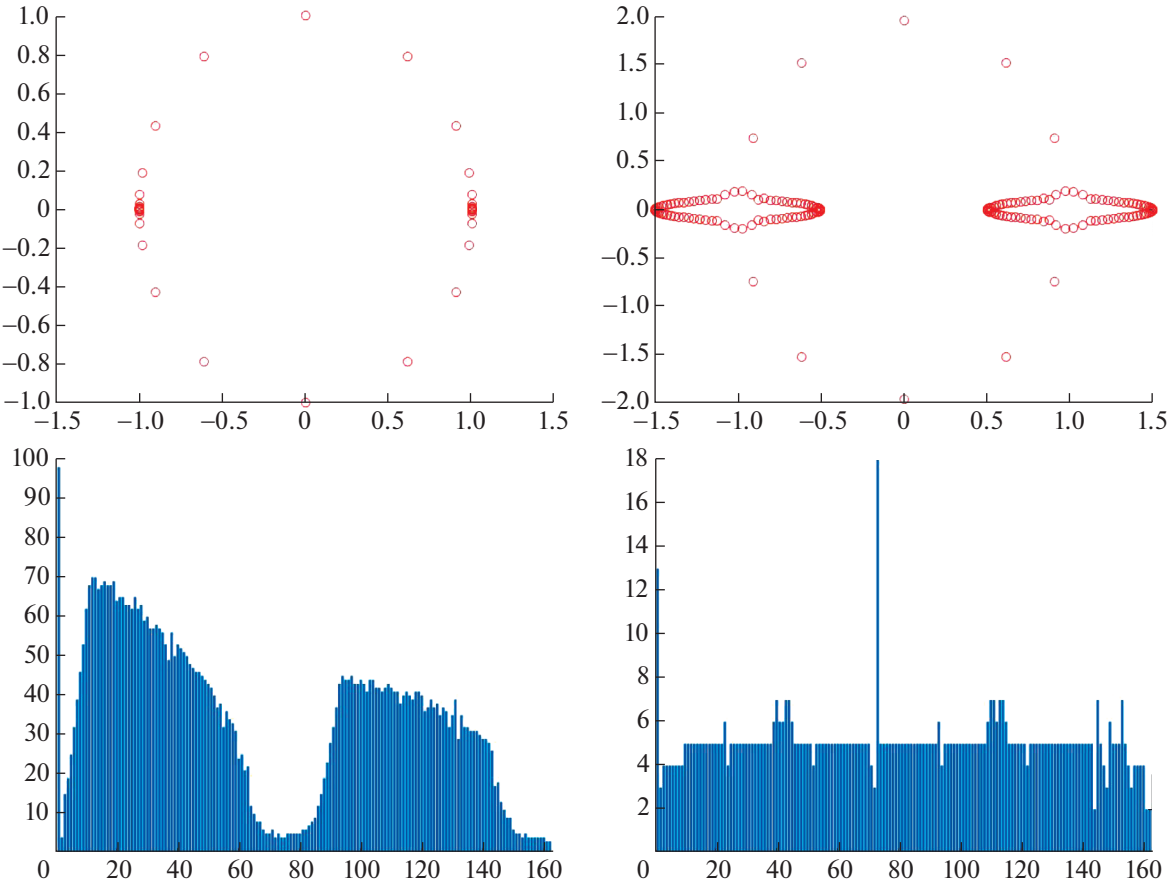
$$G(t) = \frac{\alpha + 1}{\alpha + \sqrt{1 - (\alpha + 1)t}} = 1 + \frac{1}{2}t + \frac{\alpha + 3}{8}t^2 + \ldots.$$

Заметим, что для применения этих итераций нам нужно вычислять лишь ньютоновскую коррекцию и $t = ff''/f'^2$.

9. NLEVP – КОЛЛЕКЦИЯ НЕЛИНЕЙНЫХ ПРОБЛЕМ СОБСТВЕННЫХ ЗНАЧЕНИЙ

В замечательной коллекции NLEVP нелинейных проблем собственных значений есть примеры всех типов матриц (см. [8] и [9]). (http://www.maths.manchester.ac.uk/our-research/research-groups/numerical-analysis-and-scientific-computing/numerical-analysis/software/nlevp/)

Используя метод Лагера, мы решили две квадратичные задачи `sign1` и `sign2` (плотные матрицы, $n = 81$). Результаты приведены на фиг. 3. Задача `sign1` имеет $2n = 162$ собственных значений на единичной окружности с двумя кластерами в точках ± 1. Из графиков следует, что сходимость к собственным значениям из этих двух кластеров медленная. Сходимость в задаче `sign2` намного лучше, так как ее собственные значения лучше разделены. MATLAB-инструменты для замера времени `tic`, `toc` показали, что на использованном автором ноутбуке задача `sign1` решалась 63.92 с, а задача `sign2` – 10.82 с.



Фиг. 3. Слева – `sign1`, справа – `sign2`.

10. НЕПОЛИНОМИАЛЬНАЯ ПРОБЛЕМА СОБСТВЕННЫХ ЗНАЧЕНИЙ

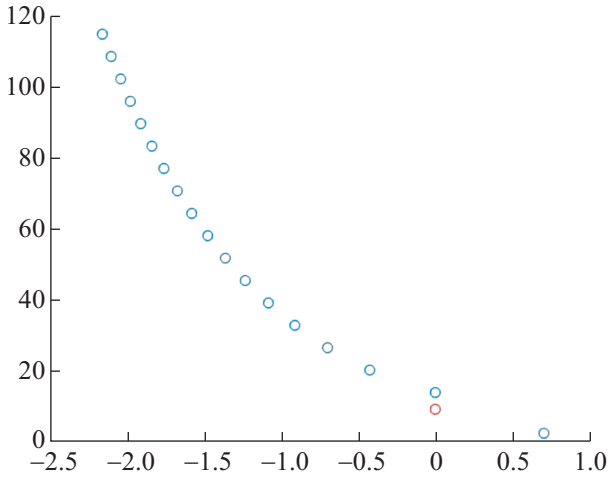
`TimeDelay` представляет собой неполиномиальную нелинейную проблему собственных значений из NLEVP-коллекции с 3×3-матрицей $A(\lambda)$:

$$A(\lambda) = -\lambda I + A_0 + A_1 e^{-\lambda}.$$

Об этой задаче пишут (см. [8]): "...характеристическое уравнение системы с запаздыванием по времени с единственной задержкой и постоянными коэффициентами. Задача имеет двойное непростое собственное значение $\lambda = 3\pi i$".

Нелинейное уравнение $\det A(\lambda) = 0$ имеет бесконечно много решений. Используя итерации Островского, мы можем, например, вычислить первые 20 решений и получить график фиг. 4. На мнимой оси получаем упомянутые выше двойные собственные значения (λ_2 и λ_3, фиг. 4) и также простое собственное значение $\lambda_4 = 4.5\pi i$. Собственное значение $\lambda_1 = 0.705244109106679 + 2.741466762205487i$ имеет положительную вещественную часть, остальные собственные значения имеют отрицательные вещественные части. Двойное собственное значение вычисляется с точностью, характерной для стандарта IEEE арифметики с плавающей точкой:

$$\lambda_1 = 0.705244109106679 + 2.741466762205487i,$$
$$\lambda_2 = 0.000000005149180 + 9.424777943433675i,$$
$$\lambda_3 = -0.000000007679198 + 9.424777969836999i,$$
$$\lambda_4 = -0.000000000000001 + 14.137166941154069i,$$
$$\lambda_5 = -0.422996397305027 + 20.485362607960255i,$$
$$\lambda_6 = -0.693701244038287 + 26.758000106609209i.$$



Фиг. 4. Нелинейная задача: пример с запаздыванием по времени.

11. ИСКЛЮЧЕНИЕ ГАУССА ДЛЯ ЛЕНТОЧНЫХ МАТРИЦ

Многие задачи из NLEVP-коллекции имеют ленточные матрицы (например, `beamsensitivity` с 7-ю или `pdde-stabiity` с 32-мя диагоналями). Определители таких матриц разумно вычислять по специальному алгоритму. MATLAB-функция для исключения Гаусса для ленточных матриц с частичным выбором описана в [3]. Если A имеет q нижних и p верхних диагоналей, то мы храним их как столбцы матрицы B. Для реализации частичного выбора добавляем q нулевых столбцов к матрице B:

$$A = \begin{bmatrix} x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & x & 0 & 0 & 0 & 0 & 0 \\ x & x & x & x & 0 & 0 & 0 & 0 \\ 0 & x & x & x & x & 0 & 0 & 0 \\ 0 & 0 & x & x & x & x & 0 & 0 \\ 0 & 0 & 0 & x & x & x & x & 0 \\ 0 & 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & 0 & x & x & x \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & x & x & 0 & 0 \\ 0 & x & x & x & 0 & 0 \\ x & x & x & x & 0 & 0 \\ x & x & x & x & 0 & 0 \\ x & x & x & x & 0 & 0 \\ x & x & x & x & 0 & 0 \\ x & x & x & x & 0 & 0 \\ x & x & x & 0 & 0 & 0 \end{bmatrix}.$$

Адаптируя функцию `det2p` к этой ленточной стуктуре, получаем функцию `det2pband`:

```
function [ffp,dffp]=det2pband(p,q,B,Bp,Bpp);
% DET2PBAND computes Newton-correction and derivative for a banded matrix
n=length(B); logfpp=0; logfp=0;
Bpp=[Bpp,zeros(n,q)];
Bp=[Bp,zeros(n,q)]; B=[B,zeros(n,q)];    % augment B with q columns
normb=norm(B,1);
for j=1:n
   maximum=0; kmax=j;                    % search pivot
   for k=j:min(j+q,n)
      if abs(B(k,j-k+q+1))>maximum,
         kmax=k; maximum=abs(B(k,j-k+q+1));
      end
   end
   if maximum<1e-14*normb;               % only small pivots
      ffp=0; dffp=0; return              % consider det=0
   end
```

```
   if j~=kmax                                  % interchange rows
      ind1=j-kmax+q+1:min(n,j+2*q+p-kmax+1);
      ind2=q+1:min(n,2*q+p+1);
      h=Bpp(kmax,ind1); Bpp(kmax,ind1)=Bpp(j,ind2); Bpp(j,ind2)=h;
      h=Bp(kmax,ind1); Bp(kmax,ind1)=Bp(j,ind2); Bp(j,ind2)=h;
      h=B(kmax,ind1); B(kmax,ind1)=B(j,ind2); B(j,ind2)=h;
   end

   logfpp=logfpp+(B(j,q+1)*Bpp(j,q+1)-Bp(j,q+1)^2)/B(j,q+1)^2;
   logfp=logfp+Bp(j,q+1)/B(j,q+1);
   for k=j+1:min(n,j+q)                        % elimination step
      ind3=j-k+q+1;
      Bpp(k,ind3)=(Bpp(k,ind3)*B(j,q+1)-Bp(j,q+1)*Bp(k,ind3))/B(j,q+1)^2 …
                 -(Bp(k,ind3)*Bp(j,q+1)+B(k,ind3)*Bpp(j,q+1))/B(j,q+1)^2 …
                 +2*Bp(j,q+1)^2*B(k,ind3)/B(j,q+1)^3;
      Bp(k,ind3)=(B(j,q+1)*Bp(k,ind3)-B(k,ind3)*Bp(j,q+1))/B(j,q+1)^2;;
      B(k,ind3)=B(k,ind3)/B(j,q+1);
   end
   for k=j+1:min(n,j+q)
      for l=j+1:min(n,j+p+q)
         ind4=l-k+q+1; ind5=j-k+q+1; ind6=l-j+q+1;
         Bpp(k,ind4)=Bpp(k,ind4)-Bpp(k,ind5)*B(j,ind6)
                     -Bp(k,ind5)*Bp(j,ind6)…
                    -Bp(k,ind5)*Bp(j,ind6)-B(k,ind5)*Bpp(j,ind6);
         Bp(k,ind4)=Bp(k,ind4)-Bp(k,ind5)*B(j,ind6)-B(k,ind5)*Bp(j,ind6);
         B(k,ind4)=B(k,ind4)-B(k,ind5)*B(j,ind6);
      end
   end
end
dffp=-logfpp/logfp^2; ffp=1/logfp;
```

Пример `beamsensitivity` дает квадратичную проблему собственных значений с 7-диагональной матрицей. Для $n = 200$ вычисляются 400 собственных значений по методу Лагера. Время измеряется MATLAB-функциями `tic`, `toc`. Если применить метод для плотной матрицы, то потребуется 83.85 с. Алгоритм с использованием ленточной структуры снижает время до 10.39 с.

12. ПРИМЕР СТРУНЫ В ВЯЗКОЙ СРЕДЕ

Хайям и др. пишут (см. [10]): “Стандартный подход к численному решению квадратичной задачи переводит $Q(\lambda) = \lambda^2 M + \lambda D + K$ в линейный полином $L(\lambda) = \lambda X + Y$ с матрицей удвоенного размера с сохранением спектра. Уравнение $L(\lambda)z = 0$ обычно решается QZ-алгоритмом для задач умеренного размера и крыловскими методами для больших разреженных задач. Обычный выбор L на практике – это первая сопровождающая форма вида

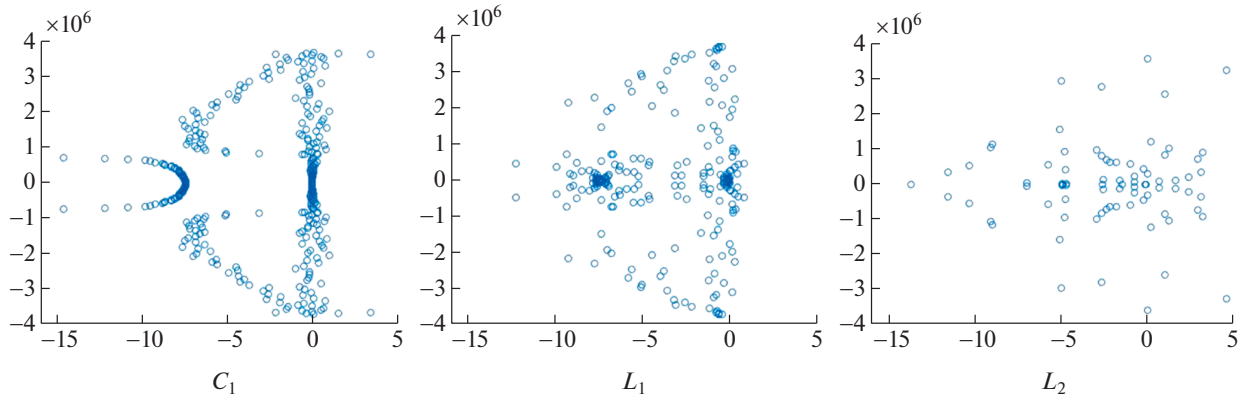
$$C_1(\lambda) = \lambda\begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} D & K \\ -I & 0 \end{bmatrix}.$$

Когда K и M невырожденные, соответственно, два пучка:

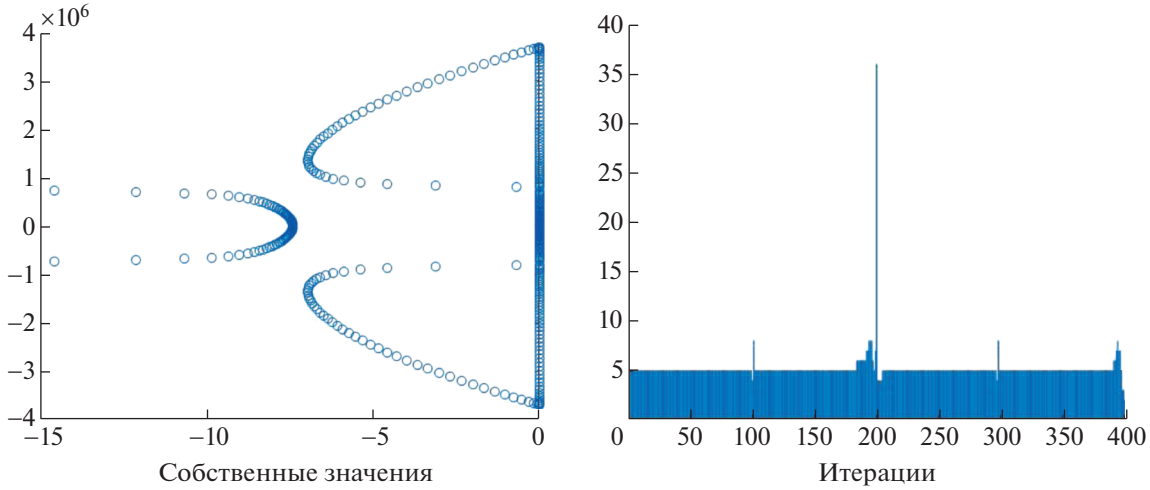
$$L_1(\lambda) = \lambda\begin{bmatrix} M & 0 \\ 0 & -K \end{bmatrix} + \begin{bmatrix} D & K \\ K & 0 \end{bmatrix}, \quad L_2(\lambda) = \lambda\begin{bmatrix} 0 & M \\ M & D \end{bmatrix} + \begin{bmatrix} -M & 0 \\ 0 & K \end{bmatrix},$$

являются другими возможными линеаризациями”.

При использовании этих линеаризаций (см. [10]) результаты решения обобщенной проблемы собственных значений с помощью MATLAB-функции `eig` довольно удручающие (фиг. 5).



Фиг. 5. Линеаризация без масштабирования.



Фиг. 6. Итерации Лагера для струны в вязкой среде.

Другие авторы (Fan, Lin, Van Dooren) показали в [11], как плохая обусловленность линеаризованных задач может быть исправлена масштабированием. В [10] объясняется, почему преобразованные системы без масштабирования являются настолько плохо обусловленными.

Ситуация напоминает мне старый пример Джима Уилкинсона, показывающий, что сведение проблемы собственных значений к задаче вычисления корней характеристического многочлена является не лучшим подходом к решению задачи, потому что изменяет ее обусловленность радикальным образом.

Однако при прямом решении уравнения $\det A(\lambda) = 0$ с помощью одного из наших методов, например, итераций Лагера, получаем корректные результаты без необходимости применять масштабирование (фиг. 6).

13. ВЫВОДЫ

Мы показали, как реализуются итерационные методы третьего порядка для решения $f(\lambda) = \det A(\lambda) = 0$ с использованием автоматического (алгоритмического) дифференцирования. Эта техника дает точные производные, так как при вычислении определителей методом Гаусса используются лишь четыре арифметические операции.

Поскольку мы работаем непосредственно с исходной задачей, нет преобразований, которые могли бы изменить обусловленность задачи.

Кубическую сходимость можно получить с помощью многошаговых итераций, которые обходятся без вторых производных. Однако, используя только точечные итерации, мы получаем алгоритм `det2p`, в котором вычисляются ньютоновская коррекция и величина $t = ff''/f'^2$, содержащая нужные нам производные.

Вычисление вторых производных дорого. Для полной $n \times n$-матрицы нужно $\sim n^3$ операций на одну итерацию. Тем не менее, благодаря мощным процессорам наших компьютеров, мы можем решать нелинейные проблемы собственных значений умеренных размеров. Ситуация намного более благоприятна в случае ленточных матриц, для которых одна итерация требует лишь $\sim n$ операций.

Автор выражает благодарность Zhong-Zhi Bai и Yu-Mei Huang – организаторам Мемориального семинара в Ланжоу, посвященного Джину Голубу (Lanzhou, 2019). Приглашение участвовать в этой конференции дало особый стимул для разработки алгоритмов, обсуждаемых в этой статье.

СПИСОК ЛИТЕРАТУРЫ

1. *Arbenz P., Gander W.* Solving nonlinear eigenvalue problems by algorithmic differentiation // Computing. 1986. V. 36. P. 205–215.
2. *Gander W.* Zeros of determinants of λ-matrices // Matrix Methods: Theory, Algorithms and Applications. Dedicated to the Memory of Gene Golub. 2010. P. 238–246.
3. *Gander W., Gander Martin J., Kwok F.* Scientific Computing, an Introduction Using MAPLE and MATLAB. Switzerland: Springer, 2014.
4. *Maehly H.J.* Zur iterativen Auflösung algebraischer Gleichunge // Zeitschrift für angewandte Mathematik und Physik. 1954. P. 260–263.
5. *Tisseur F., Meerbergen K.* The quadratic eigenvalue problem // SIAM Rev. 2001. V. 43. P. 234–286.
6. *Gander W.* On Halley's iteration method // Am. Math. Month. 1985. V. 92. № 2. P. 131–134.
7. *Hansen E., Patrick M.* A family of root finding methods // Numer. Math. 1977. V. 27. P. 257–269.
8. *Betcke T., Higham N.J., Mehrmann V., Schröder C., Tisseur F.* NLEVP: A collection of nonlinear eigenvalue problems // ACM Trans. Math. Softw. 2013. V. 39. № 2. P. 1–28.
9. *Betcke T., Higham N.J., Mehrmann V., Schröder C., Tisseur F.* A collection of nonlinear eigenvalue problems. Users' guide // MIMS EPrint 2011.117, Manchester Inst. Math. Sci., Univer. of Manchester, UK, 2011.
10. *Higham N.J., Mackey D.S., Tisseur F., Garvey S.D.* Scaling, sensitivity and stability in the numerical solution of quadratic eigenvalue problems // Int. J. Numer. Meth. Engng. 2008. V. 73. P. 344–360.
11. *Fan H.-Y., Lin W.-W., Van Dooren P.* Normwise scaling of second order polynomial matrices // SIAM J. Matrix Anal. Appl. 2004. V. 26. P. 252–256.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 800–812

ОБЩИЕ ЧИСЛЕННЫЕ МЕТОДЫ

УДК 519.65

МАЛОРАНГОВОЕ ПРЕДСТАВЛЕНИЕ НЕЙРОННЫХ СЕТЕЙ[1)]

© 2021 г. Ю. В. Гусак[1,*], Т. К. Даулбаев[1,], И. В. Оселедец[1,2], Е. С. Пономарев[1], А. С. Чихоцкий[1]**

[1] *121205 Москва, Большой бульвар, 30, стр. 1, Сколковский институт науки и технологий, Россия*

[2] *119333 Москва, ул. Губкина, 8, Институт вычислительной математики им. Г.И. Марчука Российской академии наук, Россия*

**e-mail: y.gusak@skoltech.ru*

***e-mail: t.daulbaev@skoltech.ru*

Поступила в редакцию 24.12.2020 г.
Переработанный вариант 24.12.2020 г.
Принята к публикации 14.01.2021 г.

Представлен новый метод ускорения глубоких нейронных сетей, который использует основные идеи сокращения размерности для решения уравнений в динамических системах. В основе предложенного метода лежит алгоритм поиска подматрицы максимального объема (MaxVol). Эффективность разработанного метода продемонстрирована на задаче ускорения предобученных нейронных сетей на задаче классификации изображений для трех разных наборов данных. Показано, что во многих практических задачах возможно эффективно заменить сверточные слои на полносвязные с малым числом параметров и меньшей вычислительной сложностью без существенной потери точности. Библ. 39. Фиг. 3. Табл. 4.

Ключевые слова: ускорение нейронных сетей, MaxVol, машинное обучение, анализ компонент.

DOI: 10.31857/S0044466921050100

1. ВВЕДЕНИЕ

Связь между глубокими нейронными сетями и системами обыкновенных дифференциальных уравнений (ОДУ) была показана в [1]–[4]. В упомянутых работах выход слоя нейронной сети при прямом проходе был представлен в виде состояния динамической системы в определенное время. Один из эффективных методов ускорения решения динамических систем – конструирование упрощенных моделей (см. [5]). Классический подход для построения таких моделей – метод дискретной эмпирической интерполяции (DEIM, см. [6]). Идея данного метода заключается в построении приближения вектора состояния вектором малой размерности в комбинации с эффективным пересчетом коэффициентов в полученном пространстве малой размерности с помощью нахождения подматрицы большого объема (т.е. с большим по значению определителем).

В настоящей работе мы используем связь нейросетевых алгоритмов и ОДУ для построения малорангового представления для предобученных сверточных и полносвязных нейронных сетей. Полученное малоранговое представление является полносвязной сетью с существенно меньшим числом нейронов в каждом слое, что значительно ускоряет работу модели.

Следуя подходу алгоритмов для ОДУ, предполагаем, что выходные тензоры для части слоев нейронной сети лежат в пространстве малой размерности. Мы называем это допущение *предположением о малоранговости*. Далее в работе показываем подкрепляющие его экспериментальные данные.

[1)]Работа выполнена при финансовой поддержке РФФИ (коды проектов 19-31-90172, 20-31-90127) и Минобрнауки РФ, проект 14.756.31.0001.

Итак, пусть $\mathbf{x}$ – объект из набора данных (например, картинка). Пусть $\mathbf{z}_k = \mathbf{z}_k(\mathbf{x})$ – векторизованный выходной тензор k-го слоя. Предположим, существует матрица $V_k \in \mathbb{R}^{D_k \times R_k}$ ($D_k \gg R_k$) такая, что

$$\mathbf{z}_k \cong V_k \mathbf{c}_k, \tag{1}$$

где $\mathbf{c}_k = \mathbf{c}_k(\mathbf{x})$ – вектора малой размерности, которые мы будем называть *векторами-вложениями*. Матрица V_k одинакова для всех $\mathbf{x}$.

Само линейное представление не снижает вычислительную сложность нейронной сети, потому что за каждой линейной операцией следует поэлементная нелинейная функция активации. Однако мы предлагаем способ приближенного пересчета векторов-вложений, при котором каждый следующий вектор малой размерности вычисляется на основании предыдущего малоразмерного вектора и фиксированных матриц. Данный метод мы назвали методом построения сетей меньшего порядка (Reduced-Order Network, или RON). В предположении о малоранговости наш метод может приблизить большинство сверточных нейронных сетей полносвязной сетью с существенно меньшим числом параметров и вычислительной сложностью. (Подразумеваем сети, состоящие из сверток, полносвязных слоев, неубывающих функций активации, нормализаций, максимум-пулингов, а также “остаточные” сверточные слои с непоследовательной связью слоев (блоки ResNet).)

Другими словами, вместо работы с большими по значению выходами слоев мы предлагаем проецировать вход всей нейронной сети в пространство малой размерности и работать с низкопараметрическим представлением во всех слоях. Выход последнего слоя нейронной сети переводится из малоразмерного пространства обратно в исходное с помощью линейного преобразования. В результате такого подхода вычислительная сложность всей глубокой нейронной сети существенно снижается.

На практике, даже если *предположение о малоранговости* выполняется не на всех слоях или только приближенно, получается восстановить точность модели с помощью нескольких итераций дообучения.

Эмпирически показываем, что наш алгоритм может быть эффективно использован как дополнение к методу ускорения нейронных сетей на основе прунинга каналов.

Основные результаты настоящей работы следующие.

• Предложен новый метод ускорения глубоких нейронных сетей, основанный на малоранговой аппроксимации и не требующий повторного обучения модели.

• Показано, как эффективно применять метод поиска прямоугольной матрицы максимального объема (rectangular maximum volume) для уменьшения размерности слоев, оценена ошибка такого приближения.

• Предложенный метод экспериментально проверен на серии вычислительных экспериментов по ускорению предобученных глубоких сверточных нейронных сетей (VGG and ResNet) на задаче классификации изображений для наборов данных CIFAR10, CIFAR100 и SVHN и полносвязных сетей LeNet на MNIST. В ряде случаев удалось существенно ускорить модель без потери качества.

• Продемонстрировано, что метод эффективен для ускорения сетей после процедуры прунинга, что позволило существенно ускорить уже ускоренную модель.

2. ОБЗОР ЛИТЕРАТУРЫ

В последние годы было предложено множество способов ускорить исполнение сверточных нейронных сетей (CNNs) (см. [25]). В этом разделе мы рассмотрим основные идеи различных семейств методов и выделим различия между ними и нашим подходом.

Многие различные методы имеют дело с предварительно обученной сетью, которую мы называем *сеть-Учитель*, и ускоренной сетью, называемую *сеть-Ученик*. Эта терминология взята из методов *дистилляции знаний* (knowledge distillation) (см. [26]–[29]), в которых выходы после softmax-слоя сети-Учителя используются как целевые метки для сети-Ученика.

В п. 5.4 мы сравнили наш подход с различными алгоритмами *прореживания каналов* (channel pruning). Эти методы нацелены на удаление избыточных каналов в весах различных слоев нейронной сети, тем самым ускоряя и сжимая ее. Каналы выбираются исходя из специального ин-

формационного критерия. Например, этим критерием может быть сумма абсолютных значений весов (см. [30]) или средняя доля нулей (см. [31]).

Есть два доминирующих подхода в прунинге.

Первый имеет дело с отдельно взятой сетью, которая тренируется с нуля с добавлением регуляризатора, усиливающего разреженность весов. Затем некоторые каналы признаются избыточными и удаляются (см. [18], [32]). Обычно это итерационный, вычислительно сложный процесс, особенно в случае глубоких сетей.

Второй подход задействует сеть-Учителя и сеть-Ученика. Ученик обучается минимизировать ошибку воспроизведения промежуточных выходов слоев сети-Учителя (см. [23], [31], [22]).

В [23] выбор каналов осуществляется с помощью LASSO регрессии, а реконструкция сети происходит в смысле среднеквадратичного отклонения. В работе ThiNet [22] стратегия выбора каналов зависит от статистики на следующих слоях.

В [18] было предложено умножать каждый канал на уникальное значение, получаемое в процессе обучения сети. Такой подход позволил произвести разреженную регуляризацию с помощью данных скалярных параметров. Прунинг может быть объединен с процессом поиска архитектуры, как в [13], где стратегия прунинга определяется с помощью LSTM сети, обучаемой алгоритмами обучения с подкреплением.

Наконец, в статье Discrimination-aware Channel Pruning (DCP) [12], с которой мы сравниваем свой алгоритм, к предобученной сети применена многоступенчатая схема прореживания. На каждом шаге алгоритма DCP сеть с предыдущего шага тренируется со специальным классификатором и дискриминационной функцией потерь (discriminative loss). Наименее информативные каналы либо сокращаются с фиксированной скоростью (DCP метод), либо выбираются с использованием жадного алгоритма (DCP-Adapt метод).

Другое семейство подходов к ускорению моделей – *малоранговые методы*, использующие матричное или тензорное разложение для выбора информативных параметров. В большом числе случаев существенное уменьшение вычислительной сложности достигается путем замены одного сверточного слоя на набор меньших сверточных слоев, что показано в [33]–[36], [20]. Отметим, что в большинстве малоранговых методов тензорные разложения применяются к весовым тензорам, а не к выходам слоев (см. [21]).

Наконец, стоит упомянуть методы снижения вычислительной стоимости модели путем дискретизации значений или *квантизации* (quantization) (см. [37], [38]). Такие методы могут значительно ускорить сети, но они обычно требуют специального оборудования для достижения теоретического ускорения на практике.

3. ВСПОМОГАТЕЛЬНЫЕ МЕТОДЫ И ОПРЕДЕЛЕНИЯ

Рассмотрим алгоритм выбора прямоугольной матрицы максимального объема (rectangular maximum volume) в п. 3.1 и метод нахождения подпространства вложений (embedding) малой размерности в п. 3.2. Обе эти операции играют ключевую роль в нашем методе.

3.1. Алгоритм поиска подматрицы максимального объема (MaxVol) и скетч-матрица

Прямоугольный алгоритм MaxVol – жадный алгоритм, который ищет в исходной матрице подматрицу из ее строк, имеющую максимальный объем. Объем для матрицы A определяется следующим образом:

$$\mathrm{vol}(A) = \det(A^{\top}A). \tag{2}$$

MaxVol имеет ряд практических применений (см. [7], [8]). В данной работе мы используем его для снижения размерности переопределенных систем.

Положим, $A \in \mathbb{R}^{D\times R}$ – матрица, где число строк намного превышает число столбцов (высокая матрица, $D \gg R$). Нам требуется решить следующую систему линейных уравнений:

$$A\mathbf{x} = \mathbf{b} \tag{3}$$

при фиксированной матрице A для любой правой части $\mathbf{b} \in \mathbb{R}^{D}$. Решением такой системы является

$$\mathbf{x} = A^{\dagger}\mathbf{b}, \tag{4}$$

где $A^\dagger = (A^\top A)^{-1}A^\top$ – псевдообращение Мура–Пенроуза матрицы A. Существует проблема, связанная с тем, что произведение матрицы на вектор с матрицей $A^\dagger$ размера $R \times D$ вычислительно сложно. Кроме того, для плохо обусловленных матриц поиск решения является нестабильным.

Вместо использования всех D уравнений мы выберем наиболее "репрезентативные". Для этого применим упомянутый алгоритм поиска подматрицы максимального объема (https://bitbucket.org/muxas/maxvolpy) для матрицы A. Результатом работы алгоритма является P индексов строк ($R \le P \ll D$), которые соответствуют избранным уравнениям в данной системе. Они используются для дальнейших вычислений. В нашей работе мы предполагаем, что значение параметра P лежит в интервале $[R, 2R]$.

Подматрица из P строк может быть представлена в виде матричного произведения SA, где $S \in \{0,1\}^{P\times D}$. Назовем *$S$ матрицей выбора строк*. Для удобства будем считать, что алгоритм поиска прямоугольной матрицы максимального объема возвращает матрицу выбора строк S. Тогда решением исходной системы уравнений (3) будет

$$\mathbf{x} = (SA)^\dagger(S\mathbf{b}). \tag{5}$$

Взятие строк по индексам в $\mathbf{b}$ – простая операция, поэтому итоговая стоимость подсчета $S\mathbf{b}$ равна $O(P)$. Если $(SA)^\dagger$ предварительно посчитана, то для любой новой правой части требуется посчитать лишь одно произведение матрицы на вектор с матрицей размера $R \times P$.

3.2. Вычисление вложений малой размерности

Пусть $Z \in \mathbb{R}^{N\times D}$ – выходная матрица для заданного слоя нейронной сети. Каждая строка этой матрицы соответствует одному элементу обучающей выборки, прошедшему через часть нейронной сети от начала и до рассматриваемого слоя включительно. Усеченное сингулярное разложение ранга R для $Z^\top \in \mathbb{R}^{D\times N}$ может быть посчитано следующим образом:

$$Z^\top \cong \underbrace{V}_{D\times R}\ \underbrace{\Sigma\mathbb{U}^\top}_{R\times N}. \tag{6}$$

Здесь матрица V соответствует линейному отображению вектора в подпространство вложений малой размерности.

4. ОСНОВНОЙ МЕТОД

Цель метода – построить аппроксимацию исходной нейронной сети-Учителя более быстрой нейронной сетью-Учеником.

Мы детально описываем метод на примере многослойной полносвязной сети в п. 4.1. Приложение к сверточным нейронным сетям (CNN) рассмотрено в п. 4.2, способ применения к "остаточным" сверточным нейронным сетям (семейство сетей ResNet) – в п. 4.3.

4.1. Многослойная полносвязная сеть

Рассмотрим работу алгоритма на примере многослойной полносвязной нейронной сети, называемой также многослойным персептроном (MLP).

Обозначим через ψ_k ($k = 1, 2, \ldots, K$) множество неубывающих поэлементных функций активаций. В это множество входят такие известные функции, как ReLU, ELU, Leaky ReLU и т.д. Для простоты изложения полагаем, что мы хотим ускорить всю сеть-Учителя, однако хотим подчеркнуть, что наш метод применим и для ускорения части исходной нейронной сети. Также без ограничения общности предположим, что свободные коэффициенты всех линейных слоев равны нулю.

Пусть $\mathbf{z}_0$ – входной тензор нейронной сети, который должен пройти через K слоев сети-Учителя. Происходят следующие операции:

$$\mathbf{z}_1 = \psi_1(W_1\mathbf{z}_0), \quad \mathbf{z}_2 = \psi_2(W_2\mathbf{z}_1), \quad \ldots, \quad \mathbf{z}_K = W_K\mathbf{z}_{K-1}, \tag{7}$$

где $W_k \in \mathbb{R}^{D_k\times D_{k-1}}$ – это матрица весов для k-го слоя.

Обозначим через $\mathbf{c}_1, \ldots, \mathbf{c}_K$ векторы-вложения, соответствующие промежуточным признакам (выходам скрытых слоев нейронной сети) $\mathbf{z}_1, \ldots, \mathbf{z}_K$. Обозначим линейное отображение векторов $\mathbf{z}_k$ в векторы $\mathbf{c}_k$ через $V_k \in \mathbb{R}^{D_k \times R_K}$. Это линейное отображение получается с помощью сингулярного разложения (SVD) и известно заранее. Важно отметить, что размерность k-го вложения R_K намного меньше размерности исходного тензора признаков D_k.

Предположение о малоранговости первого слоя приводит нас к следующему выражению:

$$\mathbf{z}_1 \cong \boxed{V_1\mathbf{c}_1 \cong \psi_1(W_1\mathbf{z}_0)}. \tag{8}$$

Выделенное выражение – это переопределенная линейная система с матрицами $V_1 \in \mathbb{R}^{D_1 \times R_1}$, вектором $\psi_1(W_1\mathbf{z}_0)$ и вектором неизвестных $\mathbf{c}_1$. Если $S_1 \in \mathbb{R}^{P_1 \times D_1}$ является матрицей выбора строк (см. п. 3.1) для матрицы V_1, то мы можем выписать выражение для вектора-вложения $\mathbf{c}_1$:

$$\mathbf{c}_1 \cong (S_1V_1)^\dagger S_1\psi_1(W_1z_0) = \underbrace{(S_1V_1)^\dagger}_{R_1 \times P_1}\psi_1(\underbrace{S_1W_1}_{P_1 \times D_1}\mathbf{z}_0). \tag{9}$$

Мы переставили матрицу выбора строк S_1 и поэлементную активационную функцию ψ, потому что их перестановка не меняет выражение.

Используя описанный выше подход, в том же предположении о малоранговости запишем выражение для подсчета $\mathbf{c}_2$ через вложение $\mathbf{c}_1$:

$$\mathbf{z}_2 \cong \psi_2(W_2\mathbf{z}_1) \cong \psi_2(W_2V_1\mathbf{c}_1) \cong V_2\mathbf{c}_2. \tag{10}$$

Получим линейную систему

$$\boxed{V_2\mathbf{c}_2 \cong \psi_2(W_2V_1\mathbf{c}_1)}. \tag{11}$$

Теперь применим прямоугольный алгоритм MaxVol. Если $S_2 \in \mathbb{R}^{P_2 \times D_2}$ – матрица выбора строк для матрицы V_2, то вектор $\mathbf{c}_2$ вычисляется следующим образом:

$$\mathbf{c}_2 \cong \underbrace{(S_2V_2)^\dagger}_{R_2 \times P_2}\psi_2(\underbrace{S_2W_2V_1}_{P_2 \times R_1}\mathbf{c}_1). \tag{12}$$

Продолжая процесс для всех последующих слоев, получим выражение для выхода нейронной сети-Ученика, а именно, $\mathbf{z}_k \cong V_K\mathbf{c}_K$:

$$\begin{gathered}
\mathbf{c}_1 \cong \underbrace{(S_1V_1)^\dagger}_{R_1 \times P_1}\psi_1(\underbrace{S_1W_1}_{P_1 \times D_1}\mathbf{z}_0), \\
\ldots \\
\mathbf{c}_k \cong \underbrace{(S_kV_k)^\dagger}_{R_K \times P_k}\psi_2(\underbrace{S_kW_kV_{k-1}}_{P_k \times R_{k-1}}c_{k-1}), \quad k = 1, 2, \ldots, K, \\
\mathbf{z}_K \cong V_K\mathbf{c}_K.
\end{gathered} \tag{13}$$

Предположим, что s_k – выход функции ψ_k. Тогда система уравнений (13) приобретет вид

$$\begin{gathered}
\mathbf{s}_1 \cong \psi_1(\underbrace{S_1W_1}_{P_1 \times D_1}\mathbf{z}_0), \\
\mathbf{s}_2 \cong \psi_2(\underbrace{S_2W_2V_1(S_1V_1)^\dagger}_{P_2 \times P_1}\mathbf{s}_1), \\
\ldots \\
\mathbf{s}_K \cong \psi_K(\underbrace{S_KW_KV_{K-1}(S_{K-1}V_{K-1})^\dagger}_{P_K \times P_{K-1}}\mathbf{s}_{K-1}), \\
\mathbf{z}_K \cong \underbrace{V_K(S_KV_K)^\dagger}_{D_k \times R_K}\mathbf{s}_K.
\end{gathered} \tag{14}$$

В результате вместо сети с K слоями размера $D_k \times D_{k+1}$ (см. (7)) мы получим намного более компактную сеть из $K+1$ слоя (см. (14)).

Наш подход можно представить в виде алгоритма (см. алгоритм 1):

Algorithm 1: Инициализация сети-Ученика

Input: Веса слоев сети-Учителя $\{W_1, \ldots, W_K\}$; список поэлементных функций активации $\{\psi_1, \ldots, \psi_K\}$; подвыборка из тренировочной выборки Z, размер которой есть (число элементов подвыборки) $\times$ (размер входного вектора); $\{R_1, \ldots, R_K\}$ – размеры вложений;

Output: Веса сети-Ученика $\{\widetilde{W}_0, \widetilde{W}_1, \ldots, \widetilde{W}_K\}$;

▷ Для упрощения реализации алгоритма мы храним все веса $\{V_k\}_{k=1}^K$, но на самом деле достаточно хранить лишь два.

for $k \leftarrow 1$ **to** K **do**

  $Z \leftarrow$ проход данных Z через k-й слой

  $\mathbb{U}, \Sigma, V_k \leftarrow \text{truncated\_svd}(Z^\top, R_K)$

  ▷ На практике мы не храним всю Z, и используем потоковый рандомизированный алгоритм сингулярного разложения.

  $S_k \leftarrow \text{rect\_max\_vol}(V_k)$

end

$\widetilde{W}_0 \leftarrow S_1 W_1$

for $k \leftarrow 1$ **to** $K-1$ **do**

  $\widetilde{W}_k \leftarrow S_k W_k V_{k-1} (S_{k-1} V_{k-1})^\dagger$

end

$\widetilde{W} \leftarrow V_K (S_K V_K)^\dagger$

return $\{\widetilde{W}_0, \widetilde{W}_1, \ldots, \widetilde{W}_K\}$

4.2. Сверточные нейронные сети

Свертка – линейное преобразование. Мы рассматриваем его как произведение матрицы на вектор и преобразуем сверточные слои в полносвязные. Обсудим два важных момента.

Сначала мы векторизуем все выходные данные. Теряем ли мы геометрическую структуру карты объектов? Только частично, потому что структурная информация также учтена в исходной весовой матрице.

Второй момент касается того, что число переметров в сверточной матрице больше, чем в соответствующем ей ядре. Однако эти числа сопоставимы после сжатия, если количество каналов в сверточных слоях не достаточно велико. В результате сеть-Ученик может быть не только быстрее, но и меньше, чем сеть-Учитель.

Пакетная нормализация (Batch normalization), как частно используемый слой нейронной сети, может быть объединен с полносвязным слоем. Таким образом, в сети-Студенте мы избавляемся от слоев пакетной нормализации, но сохраняем свойство нормализации.

Операция подвыборки (Maximum pooling), применяемая к выходам скрытых слоев, является локальной операцией, которая обычно отображает участок размера 2×2 в одно значение: максимальное значение в данной области. Наш метод позволяет сжимать сети с таким слоем: в процессе сэмплирования берется в 4 раза больше индексов и после применяется операция подвыборки (Maximum pooling).

4.3. "Остаточные" сверточные нейронные сети (Residual Networks)

Стандартные архитектуры с последовательными сверточными слоями (например, VGG) проигрывают в эффективности более современным архитектурам (см. [9]–[11]). Такие модели со-

держат несколько параллельных ветвей, выходы которых суммируются перед тем, как попасть на вход функции активации.

Аппроксимируем выход каждой ветви и весь результат следующим образом:

$$V\mathbf{c} \cong \psi(V_1\mathbf{c}_1 + \ldots + V_k\mathbf{c}_k). \tag{15}$$

Выражение (15) – переопределенная система линейных уравнений. Если S – матрица выбора строк для матрицы V, то вложение $\mathbf{c}$ вычисляется как

$$\mathbf{c} \cong (SV)^{\dagger}\psi(SV_1\mathbf{c}_1 + \ldots + SV_k\mathbf{c}_k). \tag{16}$$

Остальные шаги для "остаточной" сети вычисляются так же, как и для многослойного персептрона (см. п. 4.1).

4.4. Ошибка аппроксимации

Положим $\varepsilon_k = V_k\mathbf{c}_k - z_k$ – ошибка малоранговой аппроксимации, тогда

$$S_kV_k\mathbf{c}_k = (S_kV_k)^{\dagger}S_kz_k + (S_kV_k)^{\dagger}S_k\varepsilon_k \tag{17}$$

и ошибка нашего алгоритма равна $e_k := \left\|(S_kV_k)^{\dagger}S_k\varepsilon_k\right\|_2$. Учитывая информацию о нормированности $\left\|V_k^{\top}\right\|_2 = \|S_k\|_2 = 1$, получаем

$$\left\|(S_kV_k)^{\dagger}S_k\right\|_2 = \left\|V_k^{\top}V_k(S_kV_k)^{\dagger}S_k\right\|_2 \le \left\|V_k(S_kV_k)^{\dagger}\right\|_2. \tag{18}$$

Принимая во внимание лемму 4.3 и замечание 4.4 из статьи про прямоугольный алгоритм MaxVol [8], имеем

$$\left\|V_k(S_kV_k)^{\dagger}\right\|_2 \le \sqrt{1 + \frac{(D_k - P_K)r_k}{P_K + 1 - R_K}}. \tag{19}$$

(В этой статье исходная матрица обозначается через C.) Следовательно,

$$e_k \le \sqrt{1 + \frac{(D_k - P_K)R_K}{P_K + 1 - R_K}}\left\|\varepsilon_k\right\|_2. \tag{20}$$

Например, если $P_K = 1.5R_K$, то ошибка приближения e_k равна $O(\sqrt{D_k}\left\|\varepsilon_k\right\|_2)$ для $R_K = o(D_k)$.

5. ЭКСПЕРИМЕНТЫ

Представим результаты вычислительных экспериментов. Сначала приводится эмпирическое подтверждение предположения о малоранговости выходов скрытых слоев нейронной сети. Затем демонстрируется качество работы нашего алгоритма RON для ускорения полносвязных и сверточных нейронных сетей. В завершение проводится сравнение с существующими аналогами, результаты которых приведены в DCP [12] и [13].

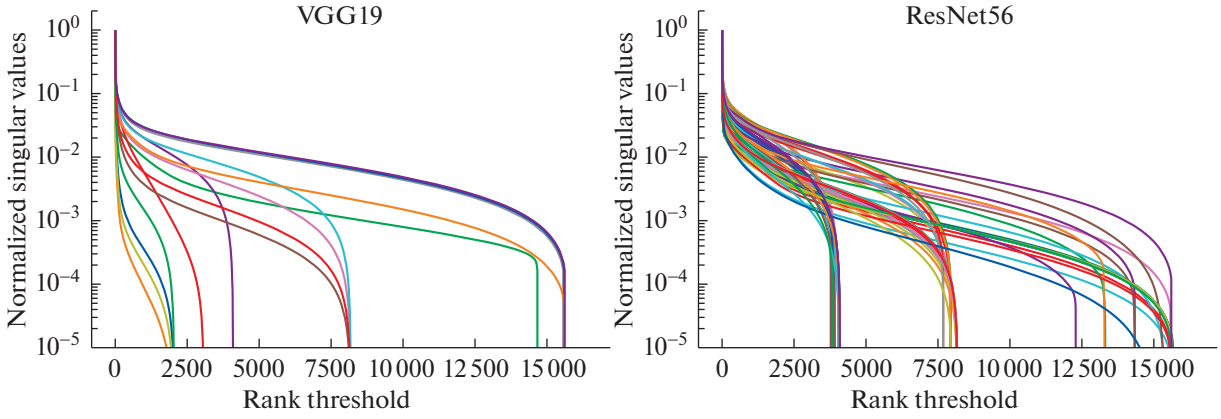
Наборы данных. Демонстрируем эффективность работы нашего алгоритма на четырех наборах данных: MNIST, CIFAR-10, CIFAR-100 и SVHN.

• MNIST – коллекция написанных от руки цифр, состоящая из 70000 картинок размера 28×28, включая 60 тысяч картинок обучающей выборки и 10 тысяч тестовой.

• CIFAR-10 – набор данных из 50000 обучающих и 10000 тестовых цветных картинок размера 32×32, на которых изображены объекты одного из 10 классов.

• CIFAR-100 содержит 100 классов, в каждом из которых по 500 обучающих и 100 тестовых изображений, параметры которых схожи с CIFAR-10.

• SVHN – датасет из фотографий с номерами уличных домов, содержащий 73257 обучающих и 26032 тестовых картинок размера 32×32.



Фиг. 1. Сингулярные числа для всех слоев для VGG19 и ResNet56, обученных на CIFAR-10. Значения нормированы на наибольшее сингулярное число для слоя. Видно, что большая их часть мала в относительном сравнении.

5.1. Сингулярные числа

Наш метод полагается на предположение о возможности эффективно отобразить выходы слоев в пространство малой размерности. Мы экспериментально проверяем это предположение для избранных архитектур на задаче классификации изображений. На фиг. 1 приведены графики для двух архитектур нейронной сети: VGG19 и ResNet56. На каждом из графиков изображены сингулярные значения промежуточных тензоров (выхода группы слоев или блока в случае ResNet). Заметим, что для части блоков сингулярные числа быстро убывают. Это означает, что их выходы могут быть хорошо приближены малоранговой аппроксимацией.

Для выбора ранга использовали два подхода: непараметрический алгоритм вариационной байесовской матричной факторизации (Variational Bayesian Matrix Factorization или VBMF из [14]) и выбор постоянного коэффициента снижения ранга.

Сингулярные числа вычисляются для матриц, содержащих данные из всей обучающей выборки. Мы использовали потоковой рандомизированный алгоритм вычисления сингулярного разложения (SVD) (см. [15], [16]), что позволило не хранить всю матрицу в памяти.

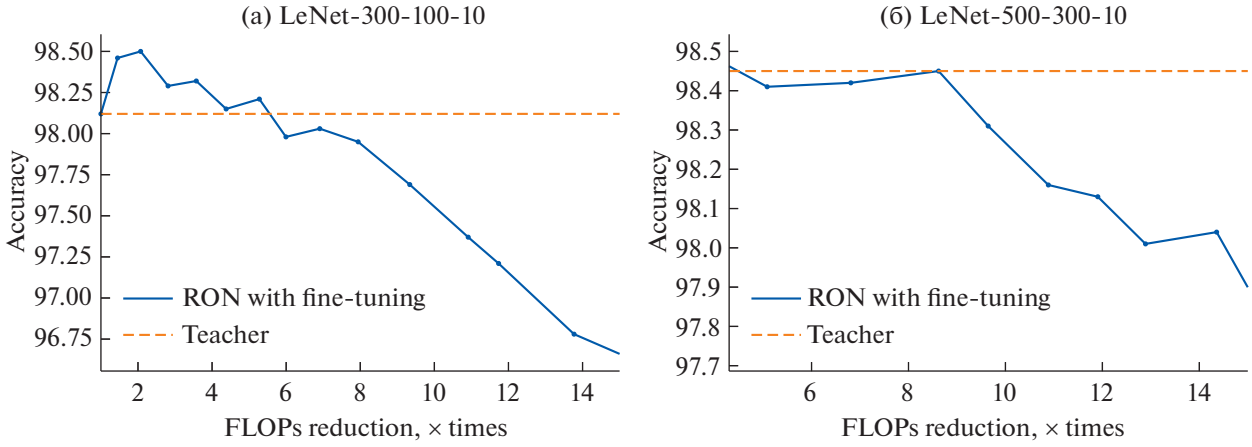
5.2. Ускорение полносвязных сетей

Для демонстрации работы метода на полносвязных сетях выбраны архитектуры LeNet-300-100 и LeNet-500-100 для классификации изображений из набора данных MNIST. LeNet-300-100 состоит из трех полносвязных слоев с матрицами размера 784×300, 300×100, 100×10 и функций активаций ReLU. В LeNet-500-100 на скрытых слоях содержится, соответственно, 500 и 100 нейронов. Осуществлено 15 итераций следующей процедуры. В начале мы обучили модель с шагом градиентного спуска `1e-3` в течение 25 эпох. Затем дообучили с меньшим шагом градиентного спуска (`5e-4`) также в течение 25 эпох. Затем применили наш метод (RON) с фиксированным уровнем снижения ранга: 0.7 и 0.75 соответственно. На фиг. 2 видно, что точность сетей убывает при более сильном ускорении.

В [13] модель LeNet-500-100 ускорена в ~7.85 раза без потери качества. Метод RON позволяет получить ускорение более чем в $\times 8$ раз (фиг. 2б).

5.3. Ускорение сверточных сетей

Мы применили наш метод к сверточным сетям архитектуры, схожей с VGG (см. [17]), для классификации на CIFAR-10, CIFAR-100 и SVHN. Во всех экспериментах лишь один раз использовали алгоритм RON, а затем дообучали получившуюся сеть, если это было необходимо. В процессе инициализации сети-Ученика (см. алгоритм 1) размеры вложений для CIFAR-10 выбирались с помощью VBMF, а для CIFAR-100 и SVHN – с помощью наперед



Фиг. 2. Метод RON для различных моделей LeNet.

заданного коэффициента сжатия. Мы использовали предобученные нейронные сети для максимальной чистоты эксперимента: для CIFAR-10 из репозитория Model-Zoo (https://github.com/SCUT-AILab/DCP/wiki/Model-Zoo). Данный репозиторий также содержит модель VGG-19 и ее прореженную методом DCP (см. [12]) версию. Для экспериментов на CIFAR-100 и SVHN использовались

- VGG-19 для CIFAR-100, (https://github.com/bearpaw/pytorch-classification)
- VGG-7 для SVHN. (https://github.com/aaron-xichen/pytorch-playground)

Затем мы применили RON к моделям вида VGG, ускорив несколько последних слоев. Например, в модели RON (8 to 16) были ускорены девять слоев с 8-го по 16 включительно.

RON без дообучения. Инициализируем сеть-Ученика способом, показанным в алгоритме 1, а затем измеряем достигнутое ускорение и качество полученной модели. Для VGG-19 на CIFAR-10 с помощью RON удалось построить модель, требующую в 1.53× меньшее число операций, чем исходная при улучшении качества на 0.09% без какого-либо дообучения (см. табл. 1). Мы сравнили применение нашего алгоритма к ускорению предобученной сети VGG-19 с алгоритмом из [13], чтобы наглядно показать, что без дообучения RON намного превосходит прунинг каналов до дообучения (см. [18], [13]) (см. модель с меткой “w/o fine-tuning”). Сверточные нейронные сети, которые были подвергнуты прунингу с параметром `pruning rate`, равным 0.1% (число FLOP в 1.23× меньше чем у исходной сети), демонстрируют падение качества более чем на 20% (см. фиг. 5 в [13]).

RON с дообучением. Если после ускорения модели алгоритмом RON провести ее дообучение, то зачастую можно добиться лучших показателей – качество восстановится. Так модель, ускоренная в 2.3×, показала качество на 0.28% лучше исходной сети для VGG-19 на CIFAR-10 (табл. 1). Процесс дообучения состоял из 250 эпох стохастического градиентного спуска с моментом 0.9 и размером батча 256. Шаг градиентного спуска изначально равнялся `1e-2` и сокращался в 2 раза каждые 10 эпох. При дообучении использовался дропаут (dropout).

Архитектуры VGG на CIFAR-100 (табл. 2) и SVHN (табл. 3) менее избыточны, их ускорение без потери качества меньше, чем для CIFAR-10.

Отметим, что шаги ускорения и дообучения могут быть применены пошагово, с постепенным увеличением числа сжатых слоев или степени их сжатия. Такой подход более вычислительно емкий, однако и для задачи прунинга каналов (см. [18], [12], [19], [13]) и для малоранговых методов (см. [20]) позволяет уменьшить падение качества при сжатии.

Применение RON после прунинга. Мотивация использования малорангового метода после процедуры прунинга заключается в следующем. Прунинг убирает наименее информативные каналы сверточного слоя, тем самым уменьшая и ускоряя сеть (например, в DCP из [12]). Однако сверточная сеть, составленная из самых информативных слоев (после прунинга), все еще может иметь малоранговую структуру и, значит, может быть ускорена методом RON. При применении

Таблица 1. Точность и теоретическое ускорение (уменьшение числа FLOP) для моделей, ускоренных методом RON на наборе данных CIFAR-10

Модель	Измененные слои	Acc@1 без дообучения	Acc@1 с дообучением	Сокращение числа FLOP
сеть-Учитель	–	–	93.70	1.00×
RON	10 to 16	**93.79**	**94.10**	**1.53**×
RON	9 to 16	93.46	**94.15**	**1.68**×
RON	8 to 16	90.58	**94.24**	**1.93**×
RON	7 to 16	85.79	**93.98**	**2.30**×
RON	6 to 16	72.53	93.12	3.01×
RON	5 to 16	58.12	91.88	3.66×
DCP [12]	–	–	93.96	2.00×
DCP + RON	10 to 16	**93.98**	**94.24**	**3.06**×
DCP + RON	9 to 16	**93.90**	**94.27**	**3.37**×
DCP + RON	8 to 16	91.82	**94.01**	**3.78**×
DCP + RON	7 to 16	88.88	**93.97**	**4.48**×
DCP + RON	6 to 16	81.30	93.26	5.56×
DCP + RON	5 to 16	64.12	91.5	7.21×

Примечание. DCP – метод прунинга из [12].

Таблица 2. VGG на CIFAR-100

Модель	Измененные слои	Acc@1 без дообуч.	Acc@5 без дообуч.	Acc@1 с дообуч.	Acc@5 с дообуч.	Ускорение на CPU	Сокращение числа FLOP
сеть-Учитель	–	–	–	71.95	89.41	1.00×	1.00×
RON 10×	8 to 16	70.81	88.51	**72.09**	**90.12**	**1.95**×	**1.66**×
RON 20×	8 to 16	63.94	85.12	71.89	89.95	2.15×	1.71×
RON 10×	10 to 16	60.68	82.36	70.87	90.46	1.72×	1.84×
RON 20×	10 to 16	44.07	68.29	69.69	89.78	2.19×	2.19×
RON 10×	12 to 16	42.77	67.34	66.84	88.16	2.22×	2.58×

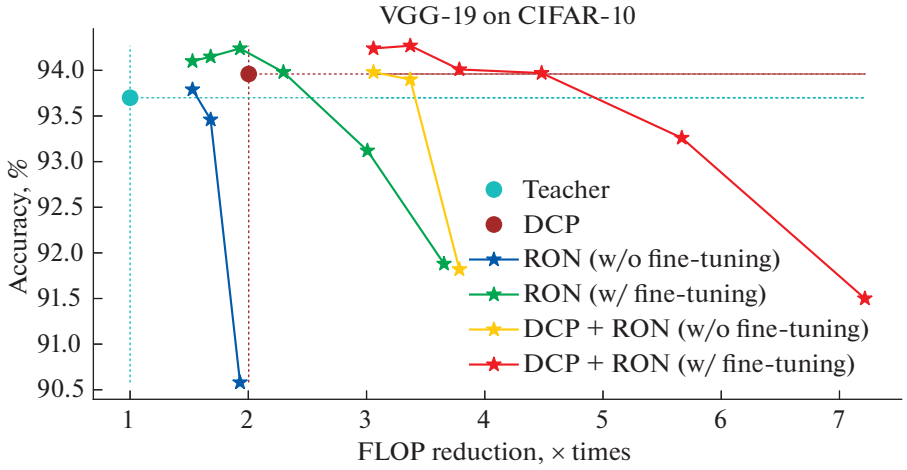
Примечание. RON $N\times$ означает ускоренную модель, у которой размерность последних слоев понижена в $N\times$ относительно исходной.

Таблица 3. VGG на SVHN

Модель	Измененные слои	Acc@1 без дообучения	Acc@1 с дообучением	Ускорение на CPU	Сокращение числа FLOP
сеть-Учитель	–	–	96.03	1.00×	1.00×
RON 10 ×	5 to 7	92.46	95.41	1.62×	1.30×
RON 20 ×	5 to 7	89.04	95.33	1.71×	1.53×
RON 20 ×	3 to 7	83.58	92.13	1.67×	1.65×

Примечание. RON $N\times$ означает ускоренную модель, у которой размерность последних слоев понижена в $N\times$ относительно исходной.

метода RON на модели VGG-19, уже прореженной методом DCP (см. [12]), при параметре `pruning rate` 0.3%, получилось добиться суммарного уменьшения числа операций в 4.48× раза при улучшении качества на 0.27% по сравнению с исходной сетью VGG-19 (фиг. 3).



Фиг. 3. Сравнение точности и теоретического ускорения (уменьшения числа FLOP) моделей, полученных методами DCP/RON/DCP + RON на наборе данных CIFAR-10. Если модель после применения RON дообучалась в течение нескольких эпох, то она имеет в скобках указание (w/ fine-tuning), в противном случае – (w/o fine-tuning).

5.4. Сравнение с другими методами

Достоинство нашего метода заключается в том, что его можно применять поверх других методов прореживания (channel pruning). Для демонстрации этого качества мы использовали уже прореженную нейронную сеть архитектуры VGG из [12] и ускорили ее. Полученное ускорение без потери качества составило 1.68× по сравнению с уже прореженной сетью или 3.36× по сравнению с исходной моделью.

Мы свели в одну табл. 4 результаты наших экспериментов и результаты из [12]. Также мы добавили к сравнению результаты из соответствующих работ для ThiNet [22], Channel pruning (CP) [23], Slimming [18] и метода width-multiplier [24].

6. ОБСУЖДЕНИЕ

Мы предложили метод, основанный на предположении о малоранговости матриц выходов слоев нейронной сети. Показали, что в ряде случаев полученная сеть-Ученик существенно быстрее исходной модели при сохранении той же точности даже без дообучения.

Минусом предложенного подхода является возможное увеличение числа параметров в результирующей полносвязной сети, когда исходной является сверточная сеть с широкими слоями. Однако наш подход эффективно работает поверх уже ускоренных методом прунинга моде-

Таблица 4. Сравнение на CIFAR-10

Модель	Сокращение числа FLOP	Падение качества, %
ThiNet [22]	2.00×	0.14
Network Sliming [18]	2.04×	0.19
Channel Pruning [23]	2.00×	0.32
Width-multiplier [24]	2.00×	0.38
Discrimination-aware Channel Pruning (DCP) [12]	2.00×	–0.17
DCP-Adapt [12]	2.86×	–0.58
RON (modified layers: 7 to 16) + fine-tuning	**2.30**×	**–0.18**
DCP + RON (modified layers: 9 to 16) + fine-tuning	**3.37**×	**–0.57**
DCP + RON (modified layers: 7 to 16) + fine-tuning	**4.48**×	**–0.27**

Примечание. VGG-19 (исходное качество 93.7%). Чем большее сокращение числа FLOP достигнуто при меньшем падении точности, тем лучше.

лей и в комбинации с ним может дать меньшее число параметров, чем было в исходной модели. В дальнейшем мы планируем использовать разреживание (sparsification) (см. [39]) и квантизацию (quantization) поверх нашего алгоритма для решения данной проблемы.

7. ЗАКЛЮЧЕНИЕ

Нами разработан метод ускорения нейронных сетей, основанный на проецировании выходов слоев в малопараметрическое подпространство. Основой метода послужили алгоритмы сингулярного разложения и поиска прямоугольной подматрицы максимального объема. Вычислительные эксперименты показали, что наш подход позволяет найти хорошее приближение исходной модели в полученном пространстве параметров новой, более быстрой, нейронной сети. А именно, в задаче классификации изображений для наборов данных CIFAR10 и CIFAR100, полученные нашим методом модели существенно ускоряют исходные и сопоставимы с ними по качеству даже без дополнительного дообучения. С дообучением исходную модель получилось ускорить в 4.48× раза без потери качества. Кроме обширных экспериментальных результатов в работе также приведена теоретическая оценка верхней границы ошибки приближения.

СПИСОК ЛИТЕРАТУРЫ

1. *Chen T.Q., Rubanova Y., Bettencourt J., Duvenaud D.K.* Neural ordinary differential equations // Adv. Neural Informat. Proc. Syst. 2018. P. 6572–6583.
2. *Grathwohl W., Chen R.T., Betterncourt J., Sutskever I., Duvenaud D.* Ffjord: Free-form continuous dynamics for scalable reversible generative models // Proc. Inter Conf. Learn. Represent. 2019.
3. *Gusak J., Markeeva L., Daulbaev T., Katrutsa A., Cichocki A., Oseledets I.* Towards Understanding Normalization in Neural ODEs// Inter. Conf. Learn. Represent. (ICLR) Workshop on Integration of Deep Neural Models and Differential Equations.
https://openreview.net/forum?id=mllQ3QNNr9d. 2020.
4. *Daulbaev T., Katrutsa A., Gusak J., Markeeva L., Cichocki A., Oseledets I.* Interpolation Technique to Speed Up Gradients Propagation in Neural ODEs. arXiv preprint arXiv:2003.05271. 2020.
5. *Quarteroni A., Rozza G., et al.* Reduced order methods for modeling and computational reduction // V. 9. Springer, 2014. № 5. С. 477–512.
6. *Chaturantabut S., Sorensen D.C.* Nonlinear model reduction via discrete empirical interpolation // SIAM J. Sci. Comput. 2010. V. 32. P. 2737–2764.
7. *Fonarev A., Mikhalev A., Serdyukov P., Gusev G., Oseledets I.* Efficient rectangular maximal-volume algorithm for rating elicitation in collaborative filtering // 2016 IEEE 16th Inter. Conf. on Data Mining (ICDM), IEEE. 2016. V. 1. P. 141–150.
8. *Mikhalev A., Oseledets I.V.* Rectangular maximum-volume submatrices and their applications // Linear Algebra and its Appl. 2018. V. 538. P. 187–211.
9. *He K., Zhang X., Ren S., Sun J.* Deep residual learning for image recognition // Proc. of the IEEE Conf. Comput. Vis. Pattern Recognit. 2016. P. 770–778.
10. *Zagoruyko S., Komodakis N.* Wide residual networks // Proc. British Machine Vis. Conf. (BMVC). 2016. P. 87.1–87.12.
11. *Huang G., Liu Z., Van Der Maaten L., Weinberger K.Q.* Densely connected convolutional networks // Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2017. P. 4700–4708.
12. *Zhuang Z., Tan M., Zhuang B., Liu J., Guo Y., Wu Q., Huang J., Zhu J.* Discrimination-aware channel pruning for deep neural networks // Adv. Neural Inform. Proc. Systems. 2018. V. 31. P. 881–892.
13. *Zhong J., Ding G., Guo Y., Han J., Wang B.* Where to prune: Using lstm to guide end-to-end pruning // Inter. Joint Conf. Artific. Intelligence. 2018. P. 3205–3211.
14. *Nakajima S., Sugiyama M., Babacan S.D., Tomioka R.* Global analytic solution of fully-observed variational bayesian matrix factorization // J. Machine Learn. Res. 2013. V. 14. P. 1–37.
15. *Woodruff D.P.* Sketching as a tool for numerical linear algebra // Foundat. and Trends in Theoret. Comput. Sci. 2014. V. 10. P. 1–157.
16. *Tsitsulin A., Munhkoeva M., Mottin D., Karras P., Oseledets I., Muller E.* Frede: Linear-space anytime graph embeddings // arXiv:2006.04746, 2020, url: https://arxiv.org/abs/2006.04746
17. *Simonyan K., Zisserman A.* Very deep convolutional networks for large-scale image recognition // arXiv:1409.1556, 2014, url: https://arxiv.org/abs/1409.1556
18. *Liu Z., Li J., Shen Z., Huang G., Yan S., Zhang C.* Learning efficient convolutional networks through network slimming // 2017 IEEE Inter. Conf. Comput. Vis. (ICCV). 2017.
19. *Gao X., Zhao Y., Dudzyak L., Mullins R., Xu C.zhong* Dynamic channel pruning: Feature boosting and suppression // Inter. Conf. Learn. Representat. 2019.

20. *Gusak J., Kholiavchenko M., Ponomarev E., Markeeva L., Blagoveschensky P., Cichocki A., Oseledets I.* Automated multi-stage compression of neural networks // IEEE/CVF Inter. Conf. Comput. Vis. Workshop (ICCVW). 2019.
21. *Cui C., Zhang K., Daulbaev T., Gusak J., Oseledets I., Zhang Z.* Active Subspace of Neural Networks: Structural Analysis and Universal Attacks // arXiv preprint arXiv:1910.13025. 2019.
22. *Luo J., Zhang H., Zhou H., Xie C., Wu J., Lin W.* Thinet: Pruning cnn filters for a thinner net // IEEE Transact. Pattern Anal. Mach. Intelligence. 2018. V. 41. Iss. 10. P. 2525–2538.
23. *He Y., Zhang X., Sun J.* Channel pruning for accelerating very deep neural networks // 2017 IEEE Inter. Conf. Comput. Vis. (ICCV). 2017.
24. *Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H.* MobileNets: Efficient Convolutional neural networks for mobile vision applications // arXiv:1704.04861, 2017, url: https://arxiv.org/abs/1704.04861
25. *Cheng Y., Wang D., Zhou P., Zhang T.* Model Compression and Acceleration for Deep Neural Networks: The Principles, Progress, and Challenges // IEEE Signal Proc. Magazine. 2018. V. 35. Iss. 1. P. 126–136.
26. *Bucilua C., Caruana R., Niculescu-Mizil A.* Model Compression // Proc. 12th ACM SIGKDD internat. Conf. on Knowledge Discovery and Data Mining. 2006. P. 535–541.
27. *Hinton G., Vinyals O., Dean J.* Distilling the Knowledge in a Neural Network // NIPS Deep Learn. and Represent. Learn. Workshop. 2015.
28. *Romero A., Ballas N., Kahou S.E., Chassang A., Gatta C., Bengio Y.* FitNets: Hints for Thin Deep Nets // Proc. Internat. Conf. Learn. Represent. 2015.
29. *Zagoruyko S., Komodakis N.* Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer // Proc. Inter. Conf. Learn. Represent. 2017.
30. *Li H., Kadav A., Durdanovic I., Samet H., Graf H.P.* Pruning filters for efficient convnets // Proc. Inter. Conf. Learn. Represent. 2017.
31. *Hu H., Peng R., Tai Y.-W., Tang C.-K.* Network trimming: A data-driven neuron pruning approach towards efficient deep architectures // arXiv:1607.03250, 2016, url: https://arxiv.org/abs/1607.03250
32. *Wen W., Wu C., Wang Y., Chen Y., Li H.* Learning structured sparsity in deep neural networks // Adv. Neural Inform. Proc. Systems. 2016. P. 2074–2082.
33. *Denton E., Zaremba W., Bruna J., LeCun Y., Fergus R.* Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation // Adv. Neural Infor. Proc. Systems. 2014. V. 2. P. 1269–1277.
34. *Jaderberg M., Vedaldi A., Zisserman A.* Speeding up Convolutional Neural Networks with Low Rank Expansions // Proc. British Mach. Vis. Conf. (BMVC). 2014.
35. *Lebedev V., Ganin Y., Rakhuba M., Oseledets I., Lempitsky V.* Speeding-up Convolutional Neural Networks Using Fine-tuned CP-Decomposition // Proc. 3rd Inter. Conf. Learn. Represent. 2015.
36. *Zhang X., Zou J., He K., Sun J.* Accelerating Very Deep Convolutional Networks for Classification and Detection // IEEE Transact. Pattern Anal. Mach. Intellig. 2016. V. 38. Iss. 10.
37. *Courbariaux M., Bengio Y., David J.-P.* Training deep neural networks with low precision multiplications // 3rd Inter. Conf. Learn. Represent. 2015.
38. *Gupta S., Agrawal A., Gopalakrishnan K., Narayanan P.* Deep Learning with Limited Numerical Precision // Proc. 32nd Inter. Conf. Inter. Conf. Mach. Learn. V. 37. 2015. P. 1737–1746.
39. *Molchanov D., Ashukha A., Vetrov D.* Variational dropout sparsifies deep neural networks // Proc. 34th Inter. Conf. Mach. Learn. 2017. V. 70. P. 2498–2507.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 813–826

ОБЩИЕ ЧИСЛЕННЫЕ МЕТОДЫ

УДК 512.64

О ТОЧНОСТИ КРЕСТОВЫХ И СТОЛБЦОВЫХ МАЛОРАНГОВЫХ MAXVOL-ПРИБЛИЖЕНИЙ В СРЕДНЕМ[1)]

© 2021 г. Н. Л. Замарашкин[1,*], А. И. Осинский[1,2,**]

[1] 119333 Москва, ул. Губкина, 8, ИВМ РАН, Россия

[2] 121205 Москва, Большой бульвар, 30, стр. 1, Сколтех, Россия

*e-mail: nikolai.zamarashkin@gmail.com

**e-mail: a.osinskiy@skoltech.ru

Поступила в редакцию 24.11.2020 г.
Переработанный вариант 24.11.2020 г.
Принята к публикации 14.01.2021 г.

В данной статье рассматривается проблема малорангового столбцового и крестового (*CGR*, *CUR*) приближения матриц по норме Фробениуса с точностью до фиксированного множителя $1+\varepsilon$. Доказывается, что для случайных матриц в среднем справедлива оценка вида $1+\varepsilon \le \frac{m+1}{m-r+1}\frac{n+1}{n-r+1}$, где m и n – число строк и столбцов крестового приближения. Таким образом, оказывается, что матрицы, для которых принцип максимального объема не позволяет гарантировать высокой точности, довольно редки. Также рассматривается связь полученных оценок с методами поиска подматрицы максимального объема и максимального проективного объема. Численные эксперименты показывают близость теоретических оценок и достижимых на практике результатов быстрой крестовой аппроксимации. Библ. 16. Фиг. 1.

Ключевые слова: малоранговое приближение матриц, крестовое/скелетное разложение, максимальный объем.

DOI: 10.31857/S0044466921050185

1. ВВЕДЕНИЕ

Пусть матрица $A \in \mathbb{C}^{M\times N}$, а A_r – ее наилучшее приближение ранга r по норме Фробениуса

$$\|A - A_r\|_F = \min_{\operatorname{rank}(B)\le r} \|A - B\|_F . \tag{1}$$

Построение приближения A_r предполагает получение сингулярного разложения для A, что слишком дорого для большого числа современных приложений, использующих малоранговые приближения. С другой стороны, как правило, не требуется искать именно A_r, а лишь такое приближение Ψ, $\operatorname{rank}(\Psi) \le r$, что

$$\|A - \Psi\|_F = (1+\varepsilon)\|A - A_r\|_F , \tag{2}$$

для некоторого заданного ε. Создание алгоритмов малой сложности для приближений матрицами малого ранга, удовлетворяющих (2), является интенсивной областью современных исследований [1]–[3].

В настоящей работе изучается точность так называемых *CGR* (иногда говорят *CUR*) малоранговых приближений, основанных на столбцах $C \in \mathbb{C}^{M\times n}$ и строках $R \in \mathbb{C}^{m\times N}$ приближаемой матрицы A. Поскольку выбранные строки и столбцы образуют в матрице A крест, то такие приближения принято называть *крестовыми*.

Известные *CGR* алгоритмы можно условно подразделить на 2 группы: детерминистические и рандомизированные. На данный момент теория рандомизированных алгоритмов развита намного полнее, чем теория детерминистических. Однако, на наш взгляд, на практике детермини-

[1)]Работа выполнена при финансовой поддержке Отделения Московского центра фундаментальной и прикладной математики в ИВМ РАН (соглашение № 075-15-2019-1624 с Минобрнауки РФ).

стические алгоритмы имеют ряд преимуществ. Во-первых, такие алгоритмы имеют меньшую сложность при получении приближений заданной точности. Во-вторых, что, возможно, даже более важно, наиболее востребованные алгоритмы используют для построения приближения лишь малую часть элементов приближаемой матрицы. Последнее делает детерминистические крестовые алгоритмы незаменимым инструментом при построении тензорных ТТ-приближений.

Настоящая работа является попыткой построения теории для детерминистических крестовых *CGR* приближений с выбором столбцов C и строк R на основе обобщенного принципа максимального объема.

Объемом прямоугольной матрицы $B \in \mathbb{C}^{m\times n}$ называется произведение всех ее сингулярных чисел

$$\mathrm{vol}(B) = \prod_{i=1}^{\min(m,n)} \sigma_i(B).$$

В соответствии с принципом максимального объема, малоранговое *CGR* приближение высокой точности получается, если на пересечении строк R и столбцов C оказывается подматрица $\hat{A}$, обладающая максимальным объемом среди всех подматриц данного размера. Теоретическое обоснование высокой поэлементной точности (в норме Чебышёва $\|A\|_C = \max_{i,j}|a_{i.j}|$) maxvol-приближений дано в [4] и обобщено в [5]. Этих оценок, тем не менее, недостаточно для доказательства существования приближений вида (2).

С другой стороны, рандомизированные алгоритмы крестовых приближений позволяют достичь высокой точности приближения в норме Фробениуса. Например, в [3] предложен алгоритм, гарантирующий точность

$$\|A - CUR\|_F \le (1+\varepsilon)\|A - A_r\|_F, \quad \varepsilon = \mathrm{const}\cdot\frac{r}{n-4r}, \tag{3}$$

с числом n строк R и столбцов C таким, что $n > 4r$. Как следует из (3), увеличивая n, можно делать точность приближения сколь угодно близкой к наилучшей. Оценка (3) является наилучшей известной оценкой крестовых аппроксимаций с точки зрения асимптотики по числу требуемых строк и столбцов n. Кроме того, нижние оценки в [6], [7] показывают, что асимптотическая оценка $\varepsilon = O(r/n)$ не может быть улучшена.

Несмотря на свойство асимптотической оптимальности, алгоритм из [3] обладает общими для рандомизированных алгоритмов недостатками. Во-первых, при построении приближения используются все элементы исходной матрицы. Кроме того, для достижения точности с $\varepsilon = 1$ реальный размер n превышает заданный ранг в десятки раз при использовании медленного метода, на основе процедуры дерандомизации, и даже тысячи раз при использовании более быстрого полностью рандомизированного алгоритма. Эти недостатки, как будет видно, отсутствуют у алгоритмов, основанных на принципе обобщенного максимального объема.

Определенные основания для предположения, что крестовые приближения на принципе максимального объема обладают высокой точностью в фробениусовой норме, было получено в [8]. Было доказано, что при выборе подматрицы $\hat{A} \in \mathbb{C}^{r\times r}$ с вероятностью, пропорциональной квадрату ее объема, матожидание ошибки во фробениусовой норме, отличается от ошибки наилучшего приближения не более чем в $r+1$ раз.

В настоящей работе мы определяем две вероятностные модели и рассматриваем подчиняющиеся им семейства случайных матриц. Мы доказываем, что средняя в норме Фробениуса ошибка *CGR* приближения на основе принципа максимального объема отличается от ошибки наилучшего приближения не более чем в $r+1$ раз. Более того, при использовании большего числа строк и столбцов можно получить среднюю погрешность, сколь угодно близкую к оптимальной, с числом строк и столбцов, существенно меньшим, чем в [3].

В работе также рассматривается средняя погрешность, даваемая так называемыми столбцовыми приближениями вида CW, где C – столбцы матрицы A, содержащие подматрицу максимального объема. Эти результаты аналогичны результатам из [7], [9]. Численные эксперименты показывают высокое совпадение наблюдаемых погрешностей с их теоретическим предсказанием.

2. ПОЧЕМУ НЕОБХОДИМ ВЕРОЯТНОСТНЫЙ ПОДХОД?

Напомним наилучшие известные результаты для точности *CGR* приближений. Все они относятся к поэлементной аппроксимации или, другими словами, к аппроксимации в C-норме.

Теорема 1 (см. [5]). *Пусть $\hat{A} \in \mathbb{C}^{n\times r}$, $n \geq r$, является подматрицей максимального объема матрицы A ранга не ниже r. Пусть C и R – столбцы и строки матрицы A, содержащие $\hat{A}$. Тогда*

$$\left\|A - C\hat{A}^{\dagger}R\right\|_C \leq \sqrt{r+1}\sqrt{\frac{n+1}{n-r+1}}\sigma_{r+1}(A). \tag{4}$$

Прямое применение неравенства (4) не позволяет получить оценку высокой точности для погрешности в норме Фробениуса. Действительно, суммируя квадрат ошибки для всех ненулевых элементов $A - C\hat{A}^{-1}R$, мы получим

$$\left\|A - C\hat{A}^{\dagger}R\right\|_F \leq \sqrt{(M-r)(N-r)}\left\|A - C\hat{A}^{-1}R\right\|_C,$$

что в итоге приводит к

$$\left\|A - C\hat{A}^{\dagger}R\right\|_F \leq \sqrt{(M-r)(N-r)}\sqrt{r+1}\sqrt{\frac{n+1}{n-r+1}}\left\|A - A_r\right\|_F.$$

От ошибки наилучшего приближения последняя оценка отличается на множитель, пропорциональный $\sqrt{(M-r)(N-r)}$. Последний зависит от размера приближаемой матрицы и делает оценку бесполезной для матриц больших размеров.

Более того, избежать такого множителя в наихудшем случае невозможно. Рассмотрим матрицы $A \in \mathbb{R}^{(r+1)\times N}$ вида

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & \ddots & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & \dfrac{1}{\sqrt{N-r+1}} & \dfrac{1}{\sqrt{N-r+1}} & \ldots & \dfrac{1}{\sqrt{N-r+1}} \\ 0 & 0 & 0 & -\dfrac{\varepsilon\sqrt{N-r}}{\sqrt{N-r+1}} & \dfrac{\varepsilon}{\sqrt{N-r}\sqrt{N-r+1}} & \ldots & \dfrac{\varepsilon}{\sqrt{N-r}\sqrt{N-r+1}} \end{bmatrix}. \tag{5}$$

Каким бы ни было значение ε, матрица максимального объема находится в первых r столбцах. Обозначим эти столбцы через C,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \dfrac{1}{\sqrt{N-r+1}} \\ 0 & 0 & 0 & -\dfrac{\varepsilon\sqrt{N-r}}{\sqrt{N-r+1}} \end{bmatrix} \in \mathbb{R}^{(r+1)\times r}.$$

Оценим ошибку наилучшего столбцового приближения CW для A. Заметим, что такое приближение достигается для матрицы $W = C^{\dagger}A$. Действительно, обозначив через b произвольный столбец в A, а через w соответствующий ему столбец W, найдем

$$\arg\min_w \left\|b - Cw\right\|_2 = C^{\dagger}b.$$

Объединив все столбцы в матрицу W, получим $W = C^{\dagger}A$. Поскольку ранг проектора $I - CC^{\dagger}$ равен 1, то матрица ошибки приближения $A - CC^{\dagger}A$ будет ранга 1, и

$$\left\|A - CC^{\dagger}A\right\|_2 = \left\|A - CC^{\dagger}\right\|_F.$$

Прямыми вычислениями получаем

$$\left\|A - CC^{\dagger}A\right\|_2 = \left\|A - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{1}{1+\varepsilon^2(N-r)} & \dfrac{\varepsilon\sqrt{N-r}}{1+\varepsilon^2(N-r)} \\ 0 & 0 & 0 & \dfrac{\varepsilon\sqrt{N-r}}{1+\varepsilon^2(N-r)} & \dfrac{\varepsilon^2(N-r)}{1+\varepsilon^2(N-r)} \end{bmatrix} A \right\|_2.$$

Учитывая то, что ошибка приближения в каждом из столбцов с номерами больше r одна и та же, и то, что для одного столбца верно равенство

$$\frac{\left\|A - CC^{\dagger}A\right\|_2}{\sqrt{N-r}} = \left\| \begin{bmatrix} 1 - \dfrac{1}{1+\varepsilon^2(N-r)} & \dfrac{\varepsilon\sqrt{N-r}}{1+\varepsilon^2(N-r)} \\ \dfrac{\varepsilon\sqrt{N-r}}{1+\varepsilon^2(N-r)} & 1 - \dfrac{\varepsilon^2(N-r)}{1+\varepsilon^2(N-r)} \end{bmatrix} \begin{bmatrix} \dfrac{1}{\sqrt{N-r+1}} \\ \dfrac{\varepsilon}{\sqrt{N-r}\sqrt{N-r+1}} \end{bmatrix} \right\|_2,$$

приходим к оценке

$$\left\|A - CC^{\dagger}A\right\|_2 \geq \sqrt{N-r} \left\| \begin{bmatrix} 0 \\ \dfrac{\varepsilon\left(\sqrt{N-r} - 1/\sqrt{N-r}\right)}{\left(1+\varepsilon^2(N-r)\right)\sqrt{N-r+1}} \end{bmatrix} \right\|_2 = \varepsilon\Omega(\sqrt{N-r}).$$

Осталось заметить, что крестовое приближение, являясь частным случаем столбцового ($CUR = CW$ для $W = UR$), не может давать оценку лучше.

Таким образом, при больших N для крестовых алгоритмов, основанных на принципе максимального объема, нельзя гарантировать высокую точность получаемых приближений. Тем не менее наблюдаемая на практике высокая эффективность таких алгоритмов говорит в пользу того, что примеры, подобные рассмотренному выше, встречаются редко. Обоснование данного наблюдения является целью настоящей работы.

3. ВЕРОЯТНОСТНАЯ МЕРА

Формализуем понятие редкости, определив RANDSVD ансамбль на матрицах с фиксированными сингулярными числами.

Определение 1. Будем говорить, что A является случайной и писать

$$A \sim RANDSVD(\Sigma),$$

если она выбирается из множества матриц вида

$$A = W_L \Sigma W_R,$$

где $\Sigma \in \mathbb{C}^{M\times N}$ – фиксированная матрица с неотрицательными элементами σ_i на диагонали, а $W_L \in \mathbb{C}^{M\times M}$ и $W_R \in \mathbb{C}^{N\times N}$ – независимые случайные унитарные матрицы с определенной для них инвариантной мерой Хаара. Без ограничения общности считаем, что диагональные элементы σ_i матрицы Σ упорядочены в порядке невозрастания $\sigma_i \geq \sigma_{i+1}$.

В соответствии с определением ансамбль RANDSVD(Σ) получает структуру вероятностного пространства и содержит все матрицы, имеющие одну и ту же матрицу сингулярных чисел Σ. Редкие события будут определяться множествами матриц, имеющих малую вероятностную меру.

Замечание 1. RANDSVD ансамбль является довольно известной конструкцией в вычислительной математике. Случайную RANDSVD матрицу для некоторых распределений сингулярных чисел можно получить в Matlab с помощью вызова функции `gallery('randsvd', ...)`. Кроме того, RANDSVD ансамбли используются при тестировании программного пакета LAPACK. LAPACK функция `zlange` позволяет получить матрицу из данного ансамбля.

4. СТОЛБЦОВЫЕ АППРОКСИМАЦИИ

Вероятностную меру на множестве матриц можно использовать для получения разнообразных оценок в среднем. Нас будут интересовать средние погрешности некоторых малоранговых приближений. Мы начнем исследование со случая так называемых столбцовых приближений.

Столбцовым приближением ранга r матрицы A называется выражение вида CW, где $C \in \mathbb{C}^{M\times n}$ – некоторые столбцы A, а $W \in \mathbb{C}^{n\times M}$ – произвольная матрица ранга не выше r, строки которой не обязаны принадлежать линейной оболочке строк A.

4.1. Модель RANDSVD шум

Пусть матрица A представляется в виде

$$A = Z + F = Z + W_L F_0 W_R,$$

с фиксированной матрицей Z, $\operatorname{rank} Z = r$, и случайной RANDSVD($F_0$) матрицей F. Таким образом, A является суммой постоянной матрицы малого ранга Z и случайной матрицы шума F.

Для матрицы Z запишем ее сингулярное разложение в виде

$$Z = U\Sigma V, \quad U \in \mathbb{C}^{M\times r}, \quad V \in \mathbb{C}^{r\times N}.$$

Сделаем важное наблюдение о том, что подматрица $\hat{Z} \in \mathbb{C}^{r\times n}$ максимального объема в Z располагается в столбцах с теми же номерами, что и подматрица $\hat{V} \in \mathbb{C}^{r\times n}$ максимального объема в матрице V.

В дальнейшем мы будем часто использовать свойство ограниченности фробениусовой нормы, псевдообратной для подматрицы максимального объема $\hat{V}$ в ортонормированных строках V.

Утверждение 1 (см. [5]). *Пусть $V \in \mathbb{C}^{r\times N}$ – матрица с ортонормированными строками: $VV^* = I_r$, а $\hat{V} \in \mathbb{C}^{r\times n}$ – подматрица матрицы V, обладающая наибольшим объемом среди всех подматриц такого размера. Тогда*

$$\left\|\hat{V}^{\dagger}\right\|_F \le \sqrt{r + \frac{r(N-n)}{n-r+1}}.$$

Кроме того, нам понадобится следующая простая

Лемма 1. *Для произвольных фиксированных матриц A и B выполняется следующее соотношение*:

$$\mathbb{E}_W \left\|AWB\right\|_F^2 = \frac{\left\|A\right\|_F^2 \left\|B\right\|_F^2}{N},$$

где математическое ожидание берется по множеству случайных унитарных матриц $W \in \mathbb{C}^{N\times N}$ с заданной на них инвариантной мерой Хаара.

Доказательство. Пусть $A = U_A\Sigma_A V_A$ и $B = U_B\Sigma_B V_B$ – сингулярные разложения матриц A и B. Тогда

$$\left\|AWB\right\|_F^2 = \left\|U_A\Sigma_A V_A W U_B \Sigma_B W_B\right\|_F^2 = \left\|\Sigma_A V_A W U_B \Sigma_B\right\|_F^2.$$

В силу унитарной инвариантности меры Хаара матрица $W' = V_A W U_B$ сама является случайной унитарной матрицей.

Представив квадрат нормы Фробениуса в виде суммы квадратов всех элементов $\Sigma_A W' \Sigma_B$, и воспользовавшись тем, что для каждого элемента случайной унитарной матрицы W' справедливо соотношение $\mathbb{E}[|w_{ij}|^2] = 1/N$, запишем:

$$\mathbb{E}_W \left\|AWB\right\|_F^2 = \sum_{ij} |v'_{ij}|^2 \sigma_i^2(A)\sigma_j^2(B) = \sum_{ij} \frac{1}{N}\sigma_i^2(A)\sigma_j^2(B) = \frac{\left\|A\right\|_F^2 \left\|B\right\|_F^2}{N}.$$

Последнее равенство доказывает утверждение леммы.

Теперь мы готовы сформулировать теорему о средней точности столбцовых приближений.

Теорема 2. *Пусть* $A = Z + F$, $F \in \mathrm{RANDSVD}\,(F_0)$, $\operatorname{rank} Z = r$, *и*

$$Z = U\Sigma V, \quad U \in \mathbb{C}^{M\times r}, \quad V \in \mathbb{C}^{r\times N},$$

есть сингулярное разложение Z. *Если столбцы* $C \in \mathbb{C}^{M\times n}$ *матрицы* A *выбираются как столбцы, соответствующие подматрице* $\hat{V} \in \mathbb{C}^{r\times n}$, *являющейся подматрицей* ***максимального объема*** *среди всех* $r \times n$ *подматриц матрицы* V, *то*

$$\mathbb{E}_{W_L,W_R}\left[\left\|A - C\hat{V}^{\dagger}V\right\|_F^2\right] \le \frac{n+1}{n-r+1}\|F\|_F^2.$$

Доказательство. Для доказательства теоремы прежде всего получим общее выражение погрешности.

Поскольку для столбцов C матрицы A выполняется соотношение

$$C = U\Sigma\hat{V} + F_C,$$

с матрицей F_C, составленной из столбцов матрицы F, соответствующих столбцам C, то для ошибки столбцового приближения с матрицей $W = \hat{V}^{\dagger}V$ справедливо представление

$$A - C\hat{V}^{\dagger}V = Z + F - (U\Sigma\hat{V} + F_C)\hat{V}^{\dagger}V = F - F_C\hat{V}^{\dagger}V = F - FP_C\hat{V}^{\dagger}V. \tag{6}$$

Матрица P_C составлена из столбцов единичной матрицы, для которых $AP_C = C$, $F_C = FP_C$ и, соответственно, $\hat{V} = VP_C$. Используя тождество $I = (I - V^*V) + V^*V$ и применяя равенства $\hat{V} = VP_C$ и $\hat{V}^{\dagger}\hat{V} = I_r$, преобразуем (6) к виду

$$\begin{aligned} A - C\hat{V}^{\dagger}V &= F - F(I - V^*V)P_C\hat{V}^{\dagger}V - FV^*VP_C\hat{V}^{\dagger}V = \\ &= F - F(I - V^*V)P_C\hat{V}^{\dagger}V - FV^*V. \end{aligned} \tag{7}$$

Объединяя первое и третье слагаемые в (7), представим ошибку как сумму двух ортогональных слагаемых

$$A - C\hat{V}^{\dagger}V = F(I - V^*V) - F(I - V^*V)P_C\hat{V}^{\dagger}V. \tag{8}$$

В силу столбцовой ортогональности матриц в правой части (8) справедливо равенство

$$\begin{aligned} \left\|A - C\hat{V}^{\dagger}V\right\|_F^2 &= \left\|F(I - V^*V)\right\|_F^2 + \left\|F(I - V^*V)P_C\hat{V}^{\dagger}V\right\|_F^2 = \\ &= \left\|F_0W_R(I - V^*V)\right\|_F^2 + \left\|F_0W_R(I - V^*V)P_C\hat{V}^{\dagger}\right\|_F^2, \end{aligned} \tag{9}$$

где мы учли, что фробениусова норма не меняется при умножении на ортонормированные строки V.

Применение леммы 1 к первому слагаемому в (9) приведет к оценке

$$\mathbb{E}_{W_R}\left[\left\|F_0W_R(I - V^*V)\right\|_F^2\right] = \frac{\|F\|_F^2\left\|I - V^*V\right\|_F^2}{N} = \|F\|_F^2\frac{N-r}{N} = \|F\|_F^2 - \frac{r}{N}\|F\|_F^2. \tag{10}$$

Аналогично для второго слагаемого с $B = (I - V^*V)P_C\hat{V}^{\dagger}$ справедливо неравенство

$$\mathbb{E}_{W_R}\left[\left\|F_0W_R(I - V^*V)P_C\hat{V}^{\dagger}\right\|_F^2\right] = \frac{\|F\|_F^2\left\|(I - V^*V)P_C\hat{V}^{\dagger}\right\|_F^2}{N} \le \|F\|_F^2\frac{\left\|\hat{V}^{\dagger}\right\|_F^2}{N}.$$

Так как $\hat{V}$ – подматрица максимального объема, то согласно утверждению 1 имеем

$$\left\|\hat{V}^{\dagger}\right\|_F^2 \le r + \frac{r(N-n)}{n-r+1}$$

и

$$\mathbb{E}_{W_R}\left[\left\|F_0W_R(I - V^*V)P_C\hat{V}^{\dagger}\right\|_F^2\right] \le \|F\|_F^2\left(\frac{r}{n-r+1} + \frac{r}{N}\right).$$

Суммируя последнее выражение с (10), получаем

$$\mathbb{E}_{W_R}\left[\left\|A - C\hat{V}^{\dagger}V\right\|_F^2\right] \le \|F\|_F^2 - \frac{r}{N}\|F\|_F^2 + \frac{r}{n-r+1}\|F\|_F^2 + \frac{r}{N}\|F\|_F^2 = \frac{n+1}{n-r+1}\|F\|_F^2.$$

Замечание 2. Оценка из теоремы 2 близка по виду к результату [7]

$$\mathbb{E}\left[\left\|A - CC^{\dagger}A\right\|_F^2\right] \le \frac{n+1}{n-r+1}\|A - A_r\|_F^2.$$

Однако смысл усреднений различен. В случае [7] матрица A фиксирована, и усреднение берется по группам столбцов. В случае теоремы 2 усреднение берется по ансамблю матриц $A = Z + F$.

Замечание 3. Часть приведенных выше рассуждений можно применить для доказательства небольшой величины ошибки столбцовой аппроксимации даже в том случае, когда матрица F не является случайной, но ее 2-норма сильно меньше нормы Фробениуса.

Действительно, из формулы (9) следует

$$\left\|A - C\hat{V}^{\dagger}V\right\|_F^2 = \left\|F\left(I - V^*V\right)\right\|_F^2 + \left\|F\left(I - V^*V\right)P_C\hat{V}^{\dagger}\right\|_F^2 \le \|F\|_F^2 + \|F\|_2^2\left\|\left(I - V^*V\right)P_C\hat{V}^{\dagger}\right\|_F^2 =$$
$$= \|F\|_F^2 + \|F\|_2^2\left(\left\|\hat{V}^{\dagger}\right\|_F^2 - \left\|V^*\hat{V}\hat{V}^{\dagger}\right\|_F^2\right) = \|F\|_F^2 + \|F\|_2^2\left(\left\|\hat{V}^{\dagger}\right\|_F^2 - r\right) \le \|F\|_F^2 + \frac{r(N-n)}{n-r+1}\|F\|_2^2.$$

Данная оценка гарантирует эффективность выбора приближения на основе подматрицы максимального объема, когда сингулярные числа погрешности одинаковы или почти одинаковы.

4.2. Модель RANDSVD матрица

Рассмотрим другую вероятностную модель. А именно, предположим, что сами матрицы A берутся из RANDSVD ансамбля. Анализ этого случая сложнее. Если в доказательстве теоремы 2 положение “хороших” столбцов C в матрице A не менялось от выбора случайной матрицы, то теперь это не так. Все столбцы в RANDSVD ансамбле равноправны, а положение столбцов, содержащих подматрицу большого объема, имеет случайный характер. Чтобы учесть это обстоятельство при вычислении средних погрешностей, нам понадобится обобщение леммы 1.

Лемма 2. *Для случайной унитарной матрицы $W \in \mathbb{C}^{N\times N}$ будем рассматривать ее блочное представление вида*

$$W = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix},$$

где $W_1 \in \mathbb{C}^{r\times N}$ – ее первые r строк, а $W_2 \in \mathbb{C}^{(N-r)\times N}$ – оставшиеся $N - r$ строк. Определим случайную матрицу F соотношением вида

$$F = F_2W_2$$

с фиксированной матрицей $F_2 \in \mathbb{C}^{M\times(N-r)}$.

Пусть $P_C \in \mathbb{C}^{M\times k}$ составлена из некоторых столбцов единичной матрицы, а F_C, как и ранее, определяется выражением $F_C = FP_C$. Пусть, наконец, $G \in \mathbb{C}^{k\times K}$ – произвольная матрица.

Тогда для условного матожидания по W при фиксированных строках W_1 справедливо неравенство

$$\mathbb{E}_W\left[\left\|F_CG\right\|_F^2 | W_1\right] \le \frac{\|F_2\|_F^2\|G\|_F^2}{N-r}\left(1 - \frac{\|W_1P_CG\|_F^2}{\|G\|_F^2}\right). \tag{11}$$

Доказательство. Пусть $F_2 = U_F\Sigma_FV_F$ – сингулярное разложение матрицы F_2. Определим матрицу Ψ в виде

$$\Psi = \begin{bmatrix} \Psi_1 \\ \Psi_2 \end{bmatrix} = \begin{bmatrix} I_{r\times r} & 0 \\ 0 & V_F \end{bmatrix}W = \begin{bmatrix} W_1 \\ V_FW_2 \end{bmatrix}.$$

В силу унитарной инвариантности меры Хаара матрицы Ψ имеют то же распределение, что и матрицы W. Теперь F_C можно представить в виде

$$F_C = U_F \Sigma_F \Psi_2 P_C. \tag{12}$$

Подставим (12) в $\|F_C G\|_F^2$ и воспользуемся унитарной инвариантностью фробениусовой нормы. Тогда

$$\|F_C G\|_F^2 = \|U_F \Sigma_F \Psi_2 P_C G\|_F^2 = \|\Sigma_F \Psi_2 P_C G\|_F^2.$$

Представим $\|\Sigma_F \Psi_2 P_C G\|_F^2$ как сумму квадратов 2-норм строк

$$\|F_C G\|_F^2 = \|\Sigma_F \Psi_2 P_C G\|_F^2 = \sum_{k=1}^{N-r} \sigma_k^2(F_2) \|(\Psi_2 P_C)_k G\|_2^2, \tag{13}$$

где обозначение $(\Psi_2 P_C)_k$ используется для строки матрицы $\Psi_2 P_C$ с номером k. Заметим, что строки $(\Psi_2 P_C)_k$ распределены одинаково (это очевидным образом следует из того, что строки матрицы W_2 распределены одинаково). Более того, одинаково распределенными являются и строки $(\Psi_2 P_C)_k G$. Следовательно, учитывая $\Psi_1 = W_1$, для произвольного k получаем

$$\begin{aligned} &\mathbb{E}_\Psi \left[\|(\Psi_2 P_C)_k G\|_2^2 \Psi_1 \right] = \frac{1}{N-r} \mathbb{E}_\Psi \left[\sum_{k=1}^{N-r} \|(\Psi_2 P_C)_k G\|_2^2 \Psi_1 \right] = \\ &= \frac{1}{N-r} \mathbb{E}_\Psi \left[\|\Psi_2 P_C G\|_F^2 \Psi_1 \right] = \frac{1}{N-r} \mathbb{E}_\Psi \left[\left(\|\Psi P_C G\|_F^2 - \|\Psi_1 P_C G\|_F^2 \right) \Psi_1 \right]. \end{aligned} \tag{14}$$

Поскольку второе слагаемое в (14) не меняется при усреднении, то имеем

$$\mathbb{E}_\Psi \left[\|(\Psi_2 P_C)_k G\|_2^2 | \Psi_1 \right] \le \frac{1}{N-r} \mathbb{E}_\Psi \left[\|\Psi P_C\|_2^2 \|G\|_F^2 | \Psi_1 \right] - \frac{\|\Psi_1 P_C G\|_F^2}{N-r} = \frac{\|G\|_F^2}{N-r} \left(1 - \frac{\|W_1 P_C G\|_F^2}{\|G\|_F^2} \right). \tag{15}$$

Усредняя (13) и подставляя в него (15), для $\mathbb{E}_W \left[\|F_C G\|_F^2 W_1 \right]$ получаем

$$\begin{aligned} &\mathbb{E}_W \left[\|F_C G\|_F^2 W_1 \right] = \mathbb{E}_\Psi \left[\sum_{k=1}^{N-r} \sigma_k^2(F_2) \|(\Psi_2 P_C)_k G\|_2^2 W_1 \right] = \\ &= \sum_{k=1}^{N-r} \sigma_k^2(F_2) \mathbb{E}_\Psi \left[\|(\Psi_2 P_C)_k G\|_2^2 W_1 \right] \le \frac{\|F_2\|_F^2 \|G\|_F^2}{N-r} - \frac{\|F_2\|_F^2 \|W_1 P_C G\|_F^2}{N-r}. \end{aligned}$$

Следствие 1. При $r = 0$ мы получаем $W_2 = W$, неравенства преобразуются в равенства, и оценка принимает вид

$$\mathbb{E}_W \left[\|F_C G\|_F^2 \right] = \frac{\|F_2\|_F^2 \|G\|_F^2}{N},$$

соответствующий утверждению леммы 1.

Замечание 4. Заметим, что оценка (11) леммы 2 не зависит от числа k столбцов, которые были выбраны у случайной матрицы F вида $F = F_2 W_2$ с помощью проектора P_C.

Теперь мы готовы доказать аналог теоремы 2 для случая, когда сама матрица A выбирается случайным образом из ансамбля RANDSVD

$$A = W_L (Z_0 + F_0) W_R = Z + F,$$

причем W_L и W_R – случайные унитарные матрицы, а Z_0, $\mathrm{rank} Z_0 = r$, и F_0 – фиксированные матрицы.

Теорема 3. *Пусть*

$$A = Z + F = W_L Z_0 W_R + W_L F_0 W_R,$$

$A \in RANDSVD\,(Z_0 + F_0)$, $\mathrm{rank} Z = r$, *и*

$$Z = U \Sigma V, \quad U \in \mathbb{C}^{M \times r}, \quad V \in \mathbb{C}^{r \times N},$$

есть сингулярное разложение Z. Если столбцы $C \in \mathbb{C}^{M\times n}$ матрицы A выбираются как столбцы, соответствующие подматрице $\hat{V} \in \mathbb{C}^{r\times n}$, являющейся подматрицей ***максимального объема*** *среди всех $r \times n$ подматриц матрицы V, то верно следующее*:

$$\mathbb{E}_{W_L,W_R}\left[\left\|A - C\hat{V}^{\dagger}V\right\|_F^2\right] \le \frac{n+1}{n-r+1}\|F\|_F^2.$$

Доказательство. Зафиксируем матрицу W_L. Как и ранее, справедливо равенство

$$A - C\hat{V}^{\dagger}V = F - F_C\hat{V}^{\dagger}V. \tag{16}$$

Повторяя преобразования из теоремы 2, будем иметь

$$\left\|A - C\hat{V}^{\dagger}V\right\|_F^2 = \left\|F(I - V^*V)\right\|_F^2 + \left\|F(I - V^*V)P_C\hat{V}^{\dagger}\right\|_F^2 \le \|F\|_F^2 + \left\|F_0W_R(I - V^*V)P_C\hat{V}^{\dagger}\right\|_F^2. \tag{17}$$

Займемся оценкой второго слагаемого в (17). Заметим, во-первых, что строки матрицы V являются одинаково распределенными. Действительно, если обозначить через $V_0 \in \mathbb{C}^{r\times N}$ правые сингулярные векторы Z_0, то $V = V_0W_R$. Кроме того,

$$\hat{V} = V_0W_RP_C$$

для некоторой матрицы $P_C \in \mathbb{C}^{N\times n}$, составленной из подмножества столбцов единичной матрицы. Подчеркнем, что P_C не является фиксированной матрицей, поскольку выбор позиций столбцов, в которых находится подматрица $\hat{V}$ максимального 2-объема в V, вообще говоря, зависит от случайной матрицы W_R. В дальнейшем для P_C будем указывать ее явную зависимость от W_R (или других матриц) и писать $P_C = P_C(W_R)$.

Обозначим матрицу во втором слагаемом в (17) через Δ и подставим туда полученные выражения для V и $\hat{V}$. Тогда имеем

$$\begin{aligned}\Delta = F_0W_R\left(I - V^*V\right)P_C(W_R)\hat{V}^{\dagger} &= F_0W_R\left(I - (V_0W_R)^*(V_0W_R)\right)P_C(W_R)\left(V_0W_RP_C(W_R)\right)^{\dagger} = \\ &= F_0(I - V_0^*V_0)W_RP_C(W_R)\left(V_0W_RP_C(W_R)\right)^{\dagger}.\end{aligned} \tag{18}$$

Рассмотрим матрицу $F' = F_0(I - V_0^*V_0)$ с сингулярным разложением $F' = U_{F'}\Sigma_{F'}V_{F'}$. Ранг F' не больше $N - r$, ее норма Фробениуса не превосходит $\|F_0\|_F$, а матрица правых сингулярных векторов $V_{F'}$ ортогональна V_0. Подставим ее в (18) и вычислим норму Фробениуса матрицы Δ:

$$\|\Delta\|_F^2 = \left\|\Sigma_F'V_F'W_RP_C(W_R)\left(V_0W_RP_C(W_R)\right)^{\dagger}\right\|_F^2. \tag{19}$$

Так как V_0 и V_F' ортогональны и образуют базис в $\mathbb{C}^N$, то соотношение

$$\Psi = \begin{bmatrix}\Psi_1\\ \Psi_2\end{bmatrix} = \begin{bmatrix}V_0\\ V_F'\end{bmatrix}W_R$$

определяет случайную унитарную матрицу Ψ. Более того, введенная ранее матрица P_C определяется только элементами Ψ_1. Действительно,

$$P_C = \arg\max_{P_C}\operatorname{vol}(V_0W_RP_C) = \arg\max_{P_C}\operatorname{vol}(V_0[V_0^* \;\; V_F^{*\prime}]\Psi P_C) = \arg\max_{P_C}\operatorname{vol}(\Psi_1P_C),$$

а потому далее будем писать $P_C = P_C(\Psi_1)$. После соответствующих замен в (19) получим

$$\|\Delta\|_F^2 \le \left\|\Sigma_F\Psi_2P_C(\Psi_1)\left(\Psi_1P_C(\Psi_1)\right)^{\dagger}\right\|_F^2. \tag{20}$$

Наконец, применим лемму 2 для оценки математического ожидания $\|\Delta\|_F^2$ по Ψ при условии фиксированной Ψ_1

$$\begin{aligned}\mathbb{E}_\Psi\left[\|\Delta\|_F^2\,\Psi_1\right] &\le \frac{\|\Sigma'_F\|_F^2}{N-r}\left\|(\Psi_1 P_C(\Psi_1))^\dagger\right\|_F^2 - \frac{\|\Sigma'_F\|_F^2}{N-r}\left\|\Psi_1 P_C(\Psi_1)(\Psi_1 P_C(\Psi_1))^\dagger\right\|_F^2 = \\ &= \frac{\|\Sigma'_F\|_F^2}{N-r}\left(\left\|(\Psi_1 P_C(\Psi_1))^\dagger\right\|_F^2 - r\right) \le \frac{\|F\|_F^2}{N-r}\left(\left\|(\Psi_1 P_C(\Psi_1))^\dagger\right\|_F^2 - r\right).\end{aligned} \tag{21}$$

Так как $\hat{V} = \Psi_1 P_C(\Psi_1)$ подматрица максимального объема в строках Ψ_1, то в силу утверждения 1 справедливо неравенство

$$\left\|(\Psi_1 P_C(\Psi_1))^\dagger\right\|_F^2 \le r + \frac{r(N-n)}{n-r+1}. \tag{22}$$

Откуда, после подстановки (22) в (21) и сокращений получаем

$$\mathbb{E}_\Psi\left[\|\Delta\|_F^2\,\Psi_1\right] \le \frac{r}{n-r+1}\|F\|_F^2. \tag{23}$$

Правая часть (23) не зависит от Ψ_1, поэтому эта же оценка верна и для безусловного среднего $\mathbb{E}_\Psi\left(\|\Delta\|_F^2\right)$. Окончательно имеем

$$\mathbb{E}_{W_R}\left[\left\|A - C\hat{V}^\dagger V\right\|_F^2\right] = \mathbb{E}_\Psi\left[\left\|A - C\hat{V}^\dagger V\right\|_F^2\right] \le \|F\|_F^2 + \frac{r}{n-r+1}\|F\|_F^2 = \left(\frac{n+1}{n-r+1}\right)\|F\|_F^2.$$

Поскольку оценка справедлива для любой фиксированной W_L, то усреднение по ней ничего не изменит, и утверждение теоремы доказано.

Замечание 5. Так как W_L с самого начала фиксировалась, теорема верна и для семейства матриц, где умножение на W_L не производится.

5. ОСНОВНОЙ РЕЗУЛЬТАТ ДЛЯ КРЕСТОВОЙ АППРОКСИМАЦИИ

Наконец, мы готовы перейти к доказательству основного результата о средней точности крестовых аппроксимаций, построенных на основе обобщенного принципа максимального проективного объема для матриц A из RANDSVD ансамбля

$$A = Z + F = W_L(Z_0 + F_0)W_R.$$

Напомним определение r-проективного объема.

Определение 2 (см. [5]). r-Проективным объемом матрицы X называется произведение ее первых (наибольших) r сингулярных чисел

$$\mathrm{vol}_r(X) = \prod_{i=1}^{r}\sigma_i(X).$$

Почему мы говорим об обобщенном принципе максимального объема? На это есть несколько причин, но основной является следующая.

С точки зрения теории крестовых приближений идеальный принцип максимального проективного объема звучит так: если для матрицы A требуется построить CGR приближение ранга не выше r, то в A выбираются подматрица $\hat{A}$ максимального проективного объема, столбцы C и строки R, на пересечении которых стоит $\hat{A}$, а приближение имеет вид

$$A \approx C\hat{A}_r^\dagger R, \quad G = \hat{A}_r^\dagger,$$

где $B_r^\dagger$ обозначает обобщенную обратную для первых r сингулярных чисел B.

Наиболее сложная часть данной конструкции состоит в анализе подматрицы максимального проективного объема и ее положения в исходной матрице. В работе мы предлагаем упрощенный подход, который и называем обобщенным принципом максимального проективного объема.

Пусть как и везде ранее

$$Z = U\Sigma V, \quad U \in \mathbb{C}^{M\times r}, \quad V \in \mathbb{C}^{r\times N}$$

есть сингулярное разложение для Z. Будем выбирать столбцы C, строки R и генератор G с помощью следующего алгоритма обобщенного принципа максимального проективного объема:

1) столбцы $C = AP_C$ соответствуют столбцам $Z_C = ZP_C$, содержащим подматрицу максимального проективного объема в Z;

2) строки R соответствуют подматрице максимального проективного объема в матрице $CZ_C^{\dagger}Z$;

3) если $\hat{A}$ обозначает подматрицу матрицы A, стоящую на пересечении столбцов C и строк R, то $G = \left(\hat{A}Z_C^{\dagger}Z_C\right)^{\dagger} = \left(A\mathcal{P}\right)^{\dagger}$, где $\mathcal{P} = Z_C^{\dagger}Z_C$ – ортопроектор на пространство размерности r.

Справедлива следующая

Теорема 4. *Пусть* $A \in$ RANDSVD $(Z_0 + F_0)$

$$A = Z + F = W_L Z_0 W_R + W_L F_0 W_R,$$

rank $Z = r$, *и*

$$Z = U\Sigma V, \quad U \in \mathbb{C}^{M\times r}, \quad V \in \mathbb{C}^{r\times N},$$

есть сингулярное разложение Z. *Пусть столбцы* $C \in \mathbb{C}^{M\times n}$, *строки* $R \in \mathbb{C}^{m\times N}$ *и генератор* $G \in \mathbb{C}^{n\times m}$ *выбираются в соответствии с обобщенным принципом максимального проективного объема. Тогда имеем*

$$\mathbb{E}_{W_L,W_R}\left[\|A - CGR\|_F^2\right] \le \frac{m+1}{m-r+1}\frac{n+1}{n-r+1}\|F\|_F^2.$$

Доказательство. Начнем доказательство с важного наблюдения. Рассмотрим произвольную матрицу $X \in \mathbb{C}^{M\times N}$ ранга r. Пусть $X = U_X\Sigma_X V_X$ – сингулярное разложение X с матрицами $U_X \in \mathbb{C}^{M\times r}$ и $V_X \in \mathbb{C}^{r\times N}$. Тогда подматрица $\hat{X} \in \mathbb{C}^{m\times n}$ максимального проективного объема в X удовлетворяет соотношению

$$\hat{X} = \hat{U}_X\Sigma_X\hat{V}_X, \quad \hat{U}_X \in \mathbb{C}^{m\times r}, \quad \hat{V}_X \in \mathbb{C}^{r\times n},$$

с подматрицами $\hat{U}_X$ и $\hat{V}_X$, имеющими максимальный объем в U_X и V_X соответственно.

Принимая во внимание выбор столбцов C в матрице A, можем записать

$$CZ_C^{\dagger}Z = C(U\Sigma\hat{V})^{\dagger}U\Sigma V = C(\Sigma\hat{V})^{\dagger}U^{*}U\Sigma V = C(\Sigma\hat{V})^{\dagger}\Sigma V = C(\hat{V})^{\dagger}\Sigma^{-1}\Sigma V = C(\hat{V})^{\dagger}V. \tag{24}$$

По замечанию к теореме 3, приближение ранга r вида $CZ_C^{\dagger}Z$ в среднем обладает хорошими аппроксимационными свойствами. А именно,

$$\mathbb{E}_{W_R}\left[\left\|A - CZ_C^{\dagger}Z\right\|_F^2\right] = \mathbb{E}_{W_R}\left[\left\|A - C(\hat{V})^{\dagger}V\right\|_F^2\right] \le \frac{n+1}{n-r+1}\|F\|_F^2. \tag{25}$$

Обозначим через Φ матрицу $\Phi = CZ_C^{\dagger}Z$. Очевидным образом ранг Φ не превосходит r. Применяя замечание к теореме 3 к Φ (только теперь рассматривается строчное приближение), запишем

$$\begin{aligned}\mathbb{E}_{W_L}\left\|A - \Phi\Phi_R^{\dagger}R\right\|_F^2 &= \mathbb{E}_{W_L}\left\|A - C(P_R^T CZ_C^{\dagger}Z_C)^{\dagger}R\right\|_F^2 = \\ &= \mathbb{E}_{W_L}\left\|A - C(\hat{A}Z_C^{\dagger}Z_C)^{\dagger}R\right\|_F^2 \le \frac{m+1}{m-r+1}\|A - \Phi\|_F^2,\end{aligned} \tag{26}$$

где по аналогии с P_C матрица P_R составлена из соответствующих столбцов единичной матрицы, а $\hat{A} = P_R^T A P_C$ – подматрица с большим проективным объемом. Комбинируя (25) и (26), заканчиваем доказательство теоремы

$$\mathbb{E}_{W_L,W_R}\|A - CGR\|_F^2 = \mathbb{E}_{W_R}\left(\mathbb{E}_{W_L}\left\|A - C(\hat{A}Z_C^{\dagger}Z_C)^{\dagger}R\right\|_F^2\right) \le \frac{m+1}{m-r+1}\mathbb{E}_{W_R}\left\|A - CZ_C^{\dagger}Z\right\|_F^2 \le$$
$$\le \frac{m+1}{m-r+1}\frac{n+1}{n-r+1}\left\|A - CZ_C^{\dagger}Z\right\|_F^2 = \frac{m+1}{m-r+1}\frac{n+1}{n-r+1}\|F\|_F^2.$$

Замечание 6. При $r = m = n$ коэффициент будет равен $(r+1)^2$. Интересно, что тот же коэффициент наблюдается в аппроксимации по норме Чебышёва [10], и при усреднении по подматрицам с вероятностью, пропорциональной квадрату их объема [8]. В этом случае можно в условиях теоремы использовать понятие объема вместо проективного объема.

Тот факт, что коэффициент при ошибке крестовой аппроксимации является произведением коэффициентов для столбцовой и строковой аппроксимации, встречается в различных работах. Например, при переходе от $r+1$ в столбцовой аппроксимации [9] к $(r+1)^2$ в крестовой [8]. Та же ситуация наблюдается и в случае известных оценок точности малоранговых приближений в спектральной норме [5], [11], [12]. Наконец, аналогичное произведение появляется при оценке ошибки неполного LU разложения [13], которое основано на алгоритме неполного QR разложения [14].

Замечание 7. Тот факт, что подматрица $\hat{A}$, определяемая алгоритмом для обобщенного принципа максимального проективного объема, действительно обладает большим проективным объемом, едва ли вызывает сомнения. Однако конструкция обобщенного принципа имеет еще одно существенное отличие от конструкции “идеального”. А именно, в идеальной конструкции $G = \hat{A}_r^{\dagger}$. В то же время для обобщенной конструкции $G = (\hat{A}\mathcal{P})^r = (\hat{A}\mathcal{P})_r^{\dagger}$, с проектором $\mathcal{P} = Z_C^{\dagger}Z_C = \hat{V}^{\dagger}\hat{V}$. Подробный теоретический анализ этого различия выходит за рамки данной статьи. Практика показывает, что отличие несущественное.

6. СВЯЗЬ С ЧИСЛЕННЫМИ АЛГОРИТМАМИ

Прежде всего следует указать на то, что все доказанные результаты остаются в силе в случае замены подматриц максимального объема на подматрицы локально максимального объема.

Определение 3. Говорят, что подматрица $\hat{A}$ матрицы A обладает локально максимальным объемом в матрице A, если объем любой другой подматрицы $\tilde{A}$ того же размера, и отличающейся от $\hat{A}$ не более чем в одной строке и в одном столбце

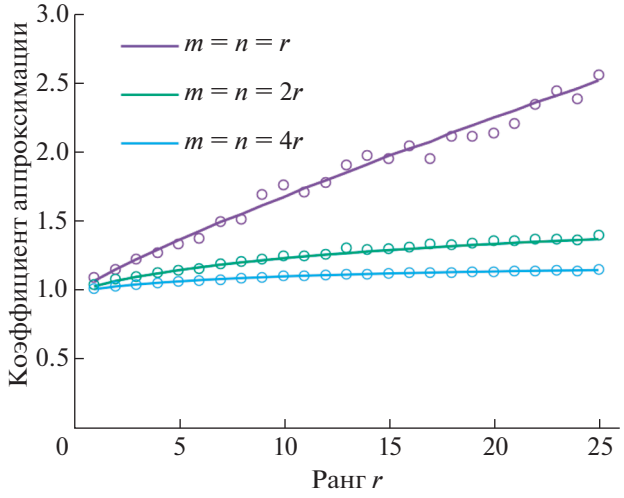
$$\mathrm{vol}\left(\tilde{A}\right) \le \mathrm{vol}\left(\hat{A}\right).$$

Алгоритмы maxvol [15] и Dominant-C [16] позволяют находить подматрицы локально максимального объема в предписанных строках и/или столбцах, а потому формально позволяют достичь доказанных ранее результатов, если только матрица Z известна.

На практике наилучшее приближение $Z = A_r$ является неизвестным. Задача состоит именно в поиске приближения, близкого к наилучшему. Для этого поиск ведется в самой матрице A, а вместо проектора P используется сокращенное сингулярное разложение подматрицы $\hat{A}$, что приводит к аппроксимации вида $C\hat{A}_r^{\dagger}R$. В связи с тем, что доказанные выше результаты уже не гарантируют оценки ошибки $\left\|A - C\hat{A}_r^{\dagger}R\right\|_F$, представляет интерес то, насколько ошибка на практике близка к той, что указана в теоремах. А именно, выполняется ли неравенство

$$\mathbb{E}\left\|A - C\hat{A}_r^{\dagger}R\right\|_F^2 \le \frac{m+1}{m-r+1}\frac{n+1}{n-r+1}\|A - A_r\|_F^2. \tag{27}$$

Для более точного сравнения уточним смысл доказанных результатов. А именно, вместо того, чтобы явно оценивать $\|\hat{V}\|_F^2$ сверху, заменим оценку на $\mathbb{E}_V\left\|\hat{V}\right\|_F^2$, где $\hat{V}$ ищется как подматрица с



Фиг. 1. Кружки обозначают значения $\left\|A - C\hat{A}_r^{\dagger}R\right\|_F / \|A - A_r\|_F$ для случайных $A \in \mathbb{R}^{N\times N}$, $N = 1000$, с сингулярными числами $\sigma_1 = \ldots = \sigma_r = 100\sigma_{r+1} = \ldots = 100\sigma_N$. Значения ошибки получены с помощью алгоритма maxvol-proj [16]. Линии показывают ожидаемое значение коэффициента аппроксимации для каждого ранга и размера. Данный коэффициент равен $1 + \frac{1}{N-r}\left(\mathop{\mathbb{E}}_{\hat{V}\in\mathbb{C}^{r\times m}}\left\|\hat{V}^{\dagger}\right\|_F^2 - r\right)$. Различные цвета показывают результаты для различных размеров подматрицы $\hat{A} \in \mathbb{C}^{m\times n}$: $m = n = r$, $m = n = 2r$ и $m = n = 4r$.

локально максимальным объемом. В этом случае коэффициенты в (27) изменятся следующим образом:

$$\mathbb{E}\left\|A - C\hat{A}_r^{\dagger}R\right\|_F^2 \approx \left(1 + \frac{\mathop{\mathbb{E}}_{\hat{V}\in\mathbb{C}^{r\times m}}\left\|\hat{V}^{\dagger}\right\|_F^2 - r}{M - r}\right)\left(1 + \frac{\mathop{\mathbb{E}}_{\hat{V}\in\mathbb{C}^{r\times m}}\left\|\hat{V}^{\dagger}\right\|_F^2 - r}{N - r}\right)\|A - A_r\|_F^2. \tag{28}$$

Выражение (28) является наиболее близкой гипотезой. Матожидания вида $\mathbb{E}_V\left\|\hat{V}\right\|_F^2$ можно получить путем отдельной (независимой от A) генерации матриц U и V, поиска в них подматриц локально максимального объема и последующего усреднения.

На фиг. 1 показаны численные значения величины $\left\|A - C\hat{A}_r^{\dagger}R\right\|_F^2$ на основе алгоритмов, не использующих знания матрицы Z, в сравнении с правой частью (28).

Как видим, численные значения ошибки близки к предсказанным теоретически и обладают малой дисперсией, особенно при числе строк и столбцов, большем r. Более подробные эксперименты из [16] также подтверждают гипотезу (28).

Использованные для тестирования процедуры поиска подматриц локально максимального объема и большого проективного объема доступны в GitHub:

https://github.com/RodniO/Projective-volume-low-rank

7. ЗАКЛЮЧЕНИЕ

Полученные результаты показывают, что в определенном смысле для большинства матриц принцип максимального объема позволяет строить столбцовые и крестовые аппроксимации высокой точности с небольшим числом дополнительных строк и столбцов. Полученные оценки имеют тот же коэффициент вида $\frac{n+1}{n-r+1}$, что и наилучшие известные оценки крестовой и столбцовой аппроксимации и имеют ту же асимптотическую зависимость от числа строк и столбцов, что и наилучшие оценки снизу.

СПИСОК ЛИТЕРАТУРЫ

1. *Halko N., Martinsson P., Tropp J.* Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions // SIAM Rev. 2011. V. 53. P. 217–288.
2. *Civril A., Magdon-Ismail M.* Column subset selection via sparse approximation of SVD // Theor. Comput. Sci. 2012. V. 412. P. 1–14.
3. *Boutsidis C., Woodruff D.P.* Optimal cur matrix decompositions // SIAM J. Comput. 2017. V. 46. P. 543–589.
4. *Goreinov S.A., Tyrtyshnikov E.E.* The maximal-volume concept in approximation by low-rank matrices // Contemp. Math. 2001. V. 268. P. 47–51.
5. *Osinsky A.I., Zamarashkin N.L.* Pseudo-skeleton approximations with better accuracy estimates // Linear Algebra Appl. 2018. V. 537. P. 221–249.
6. *Deshpande A., Vempala S.* Adaptive sampling and fast low-rank matrix approximation // Lect. Not. Comp. Sci. 2006. V. 1. P. 292–303.
7. *Guruswami V., Sinop A.K.* Optimal column-based low-rank matrix reconstruction // ArXiv e-prints. 2012. arXiv:1104.1732.
8. *Замарашкин Н.Л., Осинский А.И.* О существовании близкой к оптимальной скелетной аппроксимации матрицы во фробениусовой норме // Докл. АН. 2018. Т. 479. № 5. С. 489–492.
9. *Deshpande A., Rademacher L.* Efficient volume sampling for row/column subset selection // IEEE 51st Annual Symposium on Foundations of Computer Science. 2010. P. 329–338.
10. *Горейнов С.А., Тыртышников Е.Е.* Квазиоптимальность скелетного приближения матрицы в чебышёвской норме // Докл. АН. 2011. Т. 438. № 5. С. 593–594.
11. *Goreinov S.A., Tyrtyshnikov E.E., Zamarashkin N.L.* A theory of pseudo-skeleton approximations // Linear Algebra Appl. 1997. V. 261. P. 1–21.
12. *Michalev A.Y., Oseledets I.V.* Rectangular maximum-volume submatrices and their applications // Linear Algebra Appl. 2018. V. 538. P. 187–211.
13. *Pan C.T.* On the existence and computation of rank revealing LU factorizations // Linear Algebra Appl. 2000. V. 316. P. 199–222.
14. *Gu M., Eisenstat S.C.* Efficient algorithms for computing a strong rank-revealing qr factorization // SIAM J. Sci. Comput. 1996. V. 17. No. 4. P. 848–869.
15. *Goreinov S.A., Oseledets I.V., Savostyanov D.V., Tyrtyshnikov E.E., Zamarashkin N.L.* How to find a good submatrix // Matrix Methods: Theory, Algorithms, Applications / Ed. by V. Olshevsky, E. Tyrtyshnikov. World Scientific Publishing, 2010. P. 247–256.
16. *Osinsky A.I.* Rectangular maximum volume and projective volume search algorithms // ArXiv e-prints. 2018. arXiv:1809.02334.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 827–844

ОБЩИЕ ЧИСЛЕННЫЕ МЕТОДЫ

УДК 519.6

ПРИБЛИЖЕННЫЕ АЛГОРИТМЫ МАЛОРАНГОВОЙ АППРОКСИМАЦИИ В ЗАДАЧЕ ВОСПОЛНЕНИЯ МАТРИЦЫ НА СЛУЧАЙНОМ ШАБЛОНЕ[1)]

© 2021 г. О. С. Лебедева[1], А. И. Осинский[2,], С. В. Петров[1,*]**

[1] *119333 Москва, ул. Губкина, 8, ИВМ РАН им. Г.И. Марчука, Россия*

[2] *121205 Москва, Большой бульвар, 30, стр. 1, Сколтех, Россия*

**e-mail: spetrov.msk@gmail.com*

***e-mail: sasha_o@list.ru*

Поступила в редакцию 24.11.2020 г.
Переработанный вариант 24.11.2020 г.
Принята к публикации 14.01.2021 г.

Изучается возможность ускорения алгоритма проектирования на старшие сингулярные пространства в задаче "восполнения" матрицы малого ранга по небольшому числу ее элементов. Идея работы состоит в замене процедуры поиска наилучшего приближения во фробениусовой норме на быстрые приближенные алгоритмы. Рассматриваются два метода вычисления таких приближенний: (а) проектирование на случайные подпространства; (б) метод крестовой аппроксимации. Доказаны теоремы о геометрической сходимости алгоритмов с приближенными проекциями. Проведены численные эксперименты, показывающие эффективность обоих вариантов по сравнению с точной проекцией. Библ. 18. Фиг. 4.

Ключевые слова: матрицы малого ранга, восполнение матриц, Singular Value Projection, метод крестовой аппроксимации, случайные подпространства.

DOI: 10.31857/S0044466921050136

1. ВВЕДЕНИЕ

Пусть $X \in \mathbb{R}^{m\times n}$ – неизвестная матрица ранга k, причем $k \ll \min(m,n)$, а Ω – случайный набор пар индексов (i, j), где $i \in 1,2,\ldots,m$ и $j \in 1,2,\ldots,n$. Задача восстановления всей матрицы X лишь по элементам X_{ij} для $(i, j) \in \Omega$ называется задачей матричного восполнения (Matrix completion problem).

Современные приложения задачи матричного восполнения включают рекомендательные системы (см. [1]), обработку коррелирующих сигналов (см. [2]–[4]), машинное обучение (см. [5], [6]) и многое другое.

Одним из эффективных методов, применяемых для решения задачи восполнения, является алгоритм *проекции на главные сингулярные пространства* (Singular Value Projection method), предложенный в [7], [8] (далее для краткости мы будем называть этот метод SVP).

По своей структуре алгоритм SVP относится к классу методов проективного градиента. Каждая его итерация состоит из двух шагов: шага градиентного спуска и последующей проекции полученного приближения на многообразие матриц ранга k. Сложность одной итерации SVP практически полностью определяется сложностью построения наилучшего приближения с помощью сингулярного разложения (SVD), применяемого к $m \times n$-матрицам общего вида.

В работе рассматриваются способы ускорения метода SVP за счет замены медленного алгоритма SVD наилучшего приближения матрицами предписанного ранга k на известные быстрые методы построения малоранговых приближений. При этом не предполагается высокой точности получаемых приближений. Напротив, достаточно такой точности, при которой алгоритм восполнения сохраняет свойство глобальной сходимости к искомой матрице X, а возможное увели-

[1)]Работа поддержана Отделением Московского центра фундаментальной и прикладной математики в ИВМ РАН (Соглашение № 075-15-2019-1624 с Минобрнауки РФ).

чение общего числа итераций компенсируется скоростью построения приближениями малого ранга на каждой итерации. Как будет показано, применение такого подхода во многих случаях приводит к существенному сокращению времени работы всего алгоритма. Семейство алгоритмов, использующих алгоритмы приближенного проектирования будем называть методом ASVP (Approximate Singular Value Projection).

В работе рассматриваются два быстрых алгоритма малоранговых приближений:

- метод проектирования на случайные пространства (см. [9], [10]);
- метод псевдо-крестовой аппроксимации, основанной на принципе большого проективного объема (см. [11]–[13]).

Нашей целью является построение алгоритма ASVP, допускающего теоретическое обоснование, а также получение значимых оценок на его сложность.

Как было отмечено в [4], алгоритм SVP может проявлять нерегулярное поведение. По ходу работы алгоритма гипотезы, используемые в доказательстве его сходимости могут начать нарушаться, даже если они верны для искомой матрицы и исходного приближения; в таких случаях на практике алгоритм расходится. Для восстановления сходимости метода в [4] предложено ограничивать длину шага в градиентном спуске. Последнее замедляет сходимость и увеличивает сложность SVP. В случае метода ASVP нерегулярное поведение имеет даже большее значение. Более того, на практике нерегулярное поведение является типичным! В работе дается анализ причин нерегулярного поведения алгоритмов SVP и ASVP и предлагаются изменения, которые сохраняют сходимость алгоритма без уменьшения скорости сходимости.

2. ТЕОРИЯ СХОДИМОСТИ АЛГОРИТМА SVP

Следуя [7], мы рассматриваем теорию алгоритма SVP в несколько более общей постановке задачи восполнения.

А именно, пусть на множестве $m \times n$-матриц задан аффинный оператор $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^M$, где M обозначает число аффинных соотношений на элементах $m \times n$-матриц. Тогда для произвольного вектора $\mathbf{B} \in \mathbb{R}^M$ задачу малорангового восполнения можно поставить как задачу оптимизации

$$\psi_{\mathcal{A}}(X) = \frac{1}{2}\|\mathcal{A}(X) - \mathbf{B}\|_F^2 \to \inf, \quad \operatorname{rank}(X) \le k, \tag{1}$$

с квадратичным функционалом $\psi_{\mathcal{A}}(X)$. Следуя [7], заметим, что градиент $\nabla\psi_{\mathcal{A}}(Y)$ функционала (1) в произвольной точке $Y \in \mathbb{R}^{m\times n}$ может быть записан в виде

$$\nabla\psi_{\mathcal{A}}(Y) = \mathcal{A}^{\mathrm{T}}(\mathcal{A}(Y) - \mathbf{B}),$$

где оператор $\mathcal{A}^{\mathrm{T}} : \mathbb{R}^M \to \mathbb{R}^{m\times n}$ является транспонированным к $\mathcal{A}$ (везде предполагается, что в пространствах $\mathbb{R}^{m\times n}$ и $\mathbb{R}^M$ задано стандартное скалярное произведение).

Алгоритм SVP состоит из двух шагов: шага градиентного спуска и проекции нового приближения на многообразие матриц ранга не выше k. Другими словами, если X_t – приближение, полученное на шаге t алгоритма, то следующее приближение X_{t+1} получается в виде

$$X_{t+1} = \mathcal{P}_k\left(X_t - \tau\nabla\psi_{\mathcal{A}}(X_t)\right) = \mathcal{P}_k\left(X_t - \tau\mathcal{A}^{\mathrm{T}}\left(\mathcal{A}(X_t) - \mathbf{B}\right)\right), \tag{2}$$

где $\tau \in \mathbb{R}$ – некоторый шаг градиентного спуска, который мы определим позже, а $\mathcal{P}_k$ – наилучший во фробениусовой норме проектор на множество матриц ранга не выше k. Для сходимости SVP алгоритма в [7] доказана следующая теорема.

Теорема 1 (см. [7]). *Пусть X_* – решение* (1), *для которого $\psi_{\mathcal{A}}(X_*) = 0$, а оператор $\mathcal{A}$ удовлетворяет условию ограниченной изометрии вида*

$$(1 - \sigma)\|X\|_F^2 \le \|\mathcal{A}(X)\|_2^2 \le (1 + \sigma)\|X\|_F^2 \tag{3}$$

с параметром $0 < \sigma < 1$ для всех матриц X ранга не выше $2k$. Если дополнительно

$$\frac{2\sigma}{1 - \sigma} < 1,$$

то алгоритм SVP с постоянным шагом $\tau = 1/(1+\sigma)$ *сходится к решению* X_*, *а скорость сходимости определяется соотношением*

$$\psi_{\mathcal{A}}(X_{t+1}) \le \frac{2\sigma}{1-\sigma}\psi_{\mathcal{A}}(X_t).$$

Из (2) следует, что алгоритмическая сложность SVP определяется сложностью вычисления наилучшей проекции $\mathcal{P}_k$ на многообразие матриц ранга не выше k. Если $m \approx n$, то сложность такого вычисления, основанного на сингулярном разложении, будет порядка $\mathbb{O}(n^3)$. Цель настоящей работы состоит в исследовании возможности ускорения SVP путем замены оптимального проектора $\mathcal{P}_k$ на быстро вычисляемый приближенный $\hat{\mathcal{P}}_k$.

3. ТЕОРИЯ СХОДИМОСТИ АЛГОРИТМА SVP С ПРИБЛИЖЕННЫМ ВЫЧИСЛЕНИЕМ ПРОЕКЦИЙ (ASVP)

Формализуем понятие приближенного проектирования на многообразие матриц ранга не выше k. Пусть оператор $\hat{\mathcal{P}}_k : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$, заданный произвольным образом, всегда возвращает матрицу ранга не выше k. Будем называть $\hat{\mathcal{P}}_k$ *оператором приближенного проектирования*, если для любой матрицы Y выполнено

$$\left\|\hat{\mathcal{P}}_k(Y) - Y\right\|_F^2 \le (1+\varepsilon)\left\|\mathcal{P}_k(Y) - Y\right\|_F^2 \tag{4}$$

с некоторой константой $\varepsilon > 0$.

По аналогии с алгоритмом SVP определим итерацию приближенного алгоритма ASVP равенством

$$X_{t+1} = \hat{\mathcal{P}}_k(X_t - \tau\mathcal{A}^{\mathrm{T}}(\mathcal{A}(X) - \mathbf{B})).$$

Покажем, что для такого алгоритма ASVP справедлива теорема, аналогичная теореме 1.

Теорема 2. *Пусть на матрице* X_* *ранга не выше* k *достигается минимум функционала*

$$\psi_{\mathcal{A}}(X) = \frac{1}{2}\left\|\mathcal{A}(X) - \mathbf{B}\right\|_F^2 \to \inf, \quad \operatorname{rank}(X) \le k, \tag{5}$$

с оператором $\mathcal{A}$, *для которого условие ограниченной изометрии*

$$(1-\sigma)\|X\|_F^2 \le \left\|\mathcal{A}(X)\right\|_2^2 \le (1+\sigma)\|X\|_F^2 \tag{6}$$

выполняется на всех матрицах ранга не выше $2k$.

В этом случае для алгоритма ASVP с постоянным шагом $1/(1+\sigma)$ *и оператором* $\hat{\mathcal{P}}_k$ *приближенного проектирования, удовлетворяющим условию* (4) *с константой* ε, *справедливо неравенство*

$$\psi_{\mathcal{A}}(X_{t+1}) \le (1+\varepsilon)\psi_{\mathcal{A}}(X_*) - \varepsilon\psi_{\mathcal{A}}(X_t) +$$
$$+ (1+\varepsilon)\frac{\sigma}{1-\sigma}\left\|\mathcal{A}(X_* - X_t)\right\|_F^2 + \frac{\varepsilon\left\|\mathcal{A}^{\mathrm{T}}\right\|^2}{2(1+\sigma)}\left\|\mathcal{A}(X_t) - \mathbf{B}\right\|_F^2.$$

Доказательство. Используя арифметическое тождество

$$(b-a)^2 - (c-a)^2 = (b-c)^2 + 2(c-a)(b-c)$$

и символьную подстановку

$$a \leftrightarrow \mathbf{B}, \quad b \leftrightarrow \mathcal{A}(X), \quad c \leftrightarrow \mathcal{A}(X_t),$$

для произвольной матрицы X можно записать

$$\begin{aligned}\psi_{\mathcal{A}}(X) - \psi_{\mathcal{A}}(X_t) &= \frac{1}{2}\left\|\mathcal{A}(X) - \mathbf{B}\right\|_F^2 - \frac{1}{2}\left\|\mathcal{A}(X_t) - \mathbf{B}\right\|_F^2 = \\ &= (\mathcal{A}^{\mathrm{T}}(\mathcal{A}(X_t) - \mathbf{B}), X - X_t) + \frac{1}{2}\left\|\mathcal{A}(X - X_t)\right\|_F^2,\end{aligned} \tag{7}$$

где скалярное произведение (U,V) для матриц $U,V \in \mathbb{R}^{m\times n}$ определяется соотношением $(U,V) = \mathrm{tr}(V^{\mathrm{T}}U)$. Для произвольной матрицы X ранга не выше k воспользуемся во втором слагаемом (7) свойством ограниченной изометрии оператора $\mathcal{A}$ и установим верхнюю оценку для разности:

$$\begin{aligned}\psi_{\mathcal{A}}(X) - \psi_{\mathcal{A}}(X_t) &= (\mathcal{A}^{\mathrm{T}}(\mathcal{A}(X_t) - \mathbf{B}), X - X_t) + \frac{1}{2}\|\mathcal{A}(X - X_t)\|_F^2 \le \\ &\le (\mathcal{A}^{\mathrm{T}}(\mathcal{A}(X_t) - \mathbf{B}), X - X_t) + \frac{1+\sigma}{2}\|X - X_t\|_F^2.\end{aligned} \tag{8}$$

Правую часть (8) обозначим через $f_t(X)$. Таким образом,

$$f_t(X) = (\mathcal{A}^{\mathrm{T}}(\mathcal{A}(X_t) - \mathbf{B}), X - X_t) + \frac{(1+\sigma)}{2}\|X - X_t\|_F^2 \tag{9}$$

и

$$\psi_{\mathcal{A}}(X) - \psi_{\mathcal{A}}(X_t) \le f_t(X) \tag{10}$$

для всех матриц X, ранг которых не превосходит k.

Значения $f_t(X)$ дают оценку на падение функционала невязки на всем множестве матриц ранга не выше k, а вид $f_t(X)$ позволяет найти в явном виде матрицу ранга не выше k, на которой падение невязки является значительным.

Действительно, используем арифметическое тождество

$$\frac{1+\sigma}{2}(2pq + p^2) = \frac{1+\sigma}{2}[(p+q)^2 - q^2]$$

вместе с соответствием

$$p \leftrightarrow X - X_t, \quad q \leftrightarrow \frac{1}{1+\sigma}\mathcal{A}^{\mathrm{T}}(\mathcal{A}(X_t) - \mathbf{B}), \quad f_t(X) \leftrightarrow \frac{1+\sigma}{2}(2pq + p^2).$$

После замен $f_t(X)$ принимает вид

$$f_t(X) = \frac{1+\sigma}{2}\left\|X - \left(X_t - \frac{1}{1+\sigma}\mathcal{A}^{\mathrm{T}}(\mathcal{A}(X_t) - \mathbf{B})\right)\right\|_F^2 - \frac{1}{2(1+\sigma)}\left\|\mathcal{A}^{\mathrm{T}}(\mathcal{A}(X_t) - \mathbf{B})\right\|_F^2.$$

Определяя Y_{t+1} равенством

$$Y_{t+1} = X_t - \frac{1}{1+\sigma}\mathcal{A}^{\mathrm{T}}(\mathcal{A}(X_t) - \mathbf{B}),$$

представим $f_t(X)$ в виде суммы:

$$f_t(X) = \frac{1+\sigma}{2}\|X - Y_{t+1}\|_F^2 - \frac{1}{2(1+\sigma)}\left\|\mathcal{A}^{\mathrm{T}}(\mathcal{A}(X_t) - \mathbf{B})\right\|_F^2. \tag{11}$$

Так как второе слагаемое в (11) не зависит от X, то минимум $f_t(X)$ на множестве всех матриц ранга не выше k достигается на матрице $Z_{t+1} = \mathcal{P}_k(Y_{t+1})$. При этом сама матрица Y_{t+1} формально совпадает с матрицей, полученной градиентным спуском из точки X_t с величиной шага $\tau = 1/(1+\sigma)$, не зависящей от его номера t.

Видно, что Z_{t+1} совпадает с приближением на шаге $t+1$ в алгоритме SVP.

Так как Z_{t+1} дает минимум $f_t(X)$, то значение $f_t(Z_{t+1})$ можно оценить сверху значением $f_t(X_*)$:

$$f_t(Z_{t+1}) \le f_t(X_*) = (\mathcal{A}^{\mathrm{T}}(\mathcal{A}(X_t) - \mathbf{B}), X_* - X_t) + \frac{1+\sigma}{2}\|X_* - X_t\|_F^2.$$

Используя свойство ограниченной изометрии оператора $\mathcal{A}$ на матрице $(X_* - X_t)$, ранг которой очевидным образом не превосходит $2k$, преобразуем последнее выражение к виду

$$f_t(Z_{t+1}) \le (\mathcal{A}^{\mathrm{T}}(\mathcal{A}(X_t) - \mathbf{B}), X_* - X_t) + \frac{1+\sigma}{2(1-\sigma)}\|\mathcal{A}(X_* - X_t)\|_F^2.$$

Теперь воспользуемся формулой (7) для первого слагаемого, чтобы получить

$$f_t(Z_{t+1}) \le \psi_{\mathscr{A}}(X_*) - \psi_{\mathscr{A}}(X_t) + \frac{\sigma}{1-\sigma}\left\|\mathscr{A}(X_* - X_t)\right\|_F^2. \tag{12}$$

Положим $X_{t+1} = \hat{\mathscr{P}}_k(Y_{t+1})$. В силу определения $\hat{\mathscr{P}}_k$ выполняется неравенство

$$\left\|X_{t+1} - Y_{t+1}\right\|_F^2 \le (1+\varepsilon)\left\|Z_{t+1} - Y_{t+1}\right\|_F^2. \tag{13}$$

Рассмотрим разность $f_t(X_{t+1}) - f_t(Z_{t+1})$. Подставляя для $f_t(X_{t+1})$ и $f_t(Z_{t+1})$ их представление из (11), сокращая постоянное слагаемое и учитывая (13), получаем

$$f_t(X_{t+1}) - f_t(Z_{t+1}) = \frac{1+\sigma}{2}\left(\left\|X_{t+1} - Y_{t+1}\right\|_F^2 - \left\|Z_{t+1} - Y_{t+1}\right\|_F^2\right) \le \varepsilon\frac{1+\sigma}{2}\left\|Z_{t+1} - Y_{t+1}\right\|_F^2. \tag{14}$$

Еще раз применим (11) теперь к правой части (14). Тогда

$$f_t(X_{t+1}) - f_t(Z_{t+1}) \le \varepsilon f_t(Z_{t+1}) + \varepsilon\frac{1}{2(1+\sigma)}\left\|\mathscr{A}^{\mathrm{T}}(\mathscr{A}(X_t) - \mathbf{B})\right\|_F^2 \le$$

$$\le \varepsilon f_t(Z_{t+1}) + \varepsilon\frac{\left\|\mathscr{A}^{\mathrm{T}}\right\|^2}{2(1+\sigma)}\left\|\mathscr{A}(X_t) - \mathbf{B}\right\|_F^2,$$

где $\left\|\mathscr{A}^{\mathrm{T}}\right\|$ – операторная норма $\mathscr{A}^{\mathrm{T}}$, подчиненная второй норме в пространстве $\mathbb{R}^M$ и фробениусовой норме в пространстве $\mathbb{R}^{m\times n}$. Теперь мы готовы оценить разность $\psi(X_{t+1}) - \psi(X_t)$. Действительно, как следует из (10),

$$\psi_{\mathscr{A}}(X_{t+1}) - \psi_{\mathscr{A}}(X_t) \le f_t(X_{t+1}) = f_t(X_{t+1}) - \left(f_t(Z_{t+1}) - f_t(Z_{t+1})\right) =$$

$$= \left(f_t(X_{t+1}) - f_t(Z_{t+1})\right) + f_t(Z_{t+1}) \le (1+\varepsilon)f_t(Z_{t+1}) + \varepsilon\frac{\left\|\mathscr{A}^{\mathrm{T}}\right\|^2}{2(1+\sigma)}\left\|\mathscr{A}(X_t) - \mathbf{B}\right\|_F^2.$$

Заменяя $f_t(Z_{t+1})$ выражением (12), преобразуем последнее соотношение к окончательному виду

$$\psi_{\mathscr{A}}(X_{t+1}) - \psi_{\mathscr{A}}(X_t) \le (1+\varepsilon)\left(\psi_{\mathscr{A}}(X_*) - \psi_{\mathscr{A}}(X_t)\right) +$$

$$+ (1+\varepsilon)\frac{\sigma}{1-\sigma}\left\|\mathscr{A}(X_* - X_t)\right\|_F^2 + \frac{\varepsilon\left\|\mathscr{A}^{\mathrm{T}}\right\|^2}{2(1+\sigma)}\left\|\mathscr{A}(X_t) - \mathbf{B}\right\|_F^2.$$

Откуда после сокращения слагаемого $\psi(X_t)$ в левой и правой частях неравенства заканчиваем доказательство.

Следствие 1. *Пусть существует матрица X_* ранга не выше k, для которой $\psi_{\mathscr{A}}(X_*) = 0$ и выполняется условие ограниченной изометрии* (3). *Алгоритм ASVP с постоянным шагом $\tau = \frac{1}{1+\sigma}$ сходится к решению X_*, а скорость сходимости определяется соотношением*

$$\psi_{\mathscr{A}}(X_{t+1}) \le \psi_{\mathscr{A}}(X_t)\left(\frac{2\sigma}{1-\sigma} + \varepsilon\frac{\left\|\mathscr{A}^{\mathrm{T}}\right\|^2}{1+\sigma}\right). \tag{15}$$

Доказательство. Из предыдущей теоремы и в силу соотношений

$$\psi_{\mathscr{A}}(X_*) = 0,$$
$$\mathscr{A}(X_*) = \mathbf{B},$$
$$\psi_{\mathscr{A}}(X_t) = \frac{1}{2}\left\|\mathscr{A}(X_t - X_*)\right\|_F^2 = \frac{1}{2}\left\|\mathscr{A}(X_t) - \mathbf{B}\right\|_F^2$$

имеем

$$\psi_{\mathscr{A}}(X_{t+1}) \le \left(\frac{2\sigma(1+\varepsilon)}{1-\sigma} + \varepsilon\frac{\left\|\mathscr{A}^{\mathrm{T}}\right\|^2}{1+\sigma} - \varepsilon\right)\psi_{\mathscr{A}}(X_t) =$$
$$= \left[\frac{2\sigma}{1-\sigma} + \varepsilon\left(\frac{2\sigma}{1-\sigma} + \frac{\left\|\mathscr{A}^{\mathrm{T}}\right\|^2}{1+\sigma} - 1\right)\right]\psi_{\mathscr{A}}(X_t). \tag{16}$$

В предположении $\sigma < 1/3$, аналогичном тому, что используется для доказательства сходимости SVP в [7], видим, что ε можно выбрать достаточно малой константой, чтобы из неравенства (16) следовала геометрическая сходимость ASVP.

Как следует из формулы (15), использование приближенного проектора $\hat{\mathscr{P}}_k$ в алгоритме ASVP ухудшает коэффициент линейной сходимости, что, конечно, не является неожиданным. Однако, как будет показано далее, выигрыш, получаемый за счет упрощения $\mathscr{P}_k$, превосходит проигрыш, связанный с увеличением числа итераций в ASVP.

В следствии 1 предполагается существование решения X_*, для которого $\psi_{\mathscr{A}}(X_*) = 0$. Однако на практике для заданного вектора $\mathbf{B} \in \mathbb{R}^M$ минимум функционала может быть не равен нулю, например, в случае $\mathbf{B} = \mathscr{A}(X) + \Delta$, где X – матрица малого ранга, а Δ имеет малую норму. Оценки для такого случая представлены в следствии 2.

Следствие 2. Пусть на матрице X_* достигается минимум функционала $\psi_{\mathscr{A}}(X)$ среди всех матриц ранга не выше k, а $\Delta = \mathbf{B} - \mathscr{A}(X_*)$ имеет норму $\delta = \|\Delta\|_2$.

Если для приближения X_t выполняется неравенство $\psi_{\mathscr{A}}(X_t) \ge C^2\dfrac{\delta^2}{2}$ с константой $C > 0$, то для следующего приближения, получаемого в методе ASVP, справедливо соотношение

$$\psi_{\mathscr{A}}(X_{t+1}) < \kappa\psi_{\mathscr{A}}(X_t)$$

с константой κ, удовлетворяющей неравенству

$$\kappa \le D_0 + \varepsilon(D_0 + D_1), \tag{17}$$

где

$$D_0 = \frac{1}{C^2} + \frac{2\sigma}{1-\sigma}\left(1 + \frac{1}{C}\right)^2,$$

$$D_1 = \frac{\left\|\mathscr{A}^{\mathrm{T}}\right\|^2}{1+\sigma} - 1.$$

Доказательство. Из теоремы 2

$$\psi_{\mathscr{A}}(X_{t+1}) \le (1+\varepsilon)\frac{\delta^2}{2} + (1+\varepsilon)\frac{\sigma}{1-\sigma}\|\mathbf{B} - \mathscr{A}(X_t) - \Delta\|_F^2 + \varepsilon\left(\frac{\left\|\mathscr{A}^{\mathrm{T}}\right\|^2}{1+\sigma} - 1\right)\psi_{\mathscr{A}}(X_t).$$

Учитывая, что $\delta^2 \le \dfrac{2}{C^2}\psi(X_t)$, и раскрывая второе слагаемое, получаем

$$\psi_{\mathscr{A}}(X_{t+1}) \le \frac{(1+\varepsilon)}{C^2}\psi_{\mathscr{A}}(X_t) + (1+\varepsilon)\frac{2\sigma}{1-\sigma}\left(\psi_{\mathscr{A}}(X_t) + \frac{2}{C}\psi_{\mathscr{A}}(X_t) + \frac{1}{C^2}\psi_{\mathscr{A}}(X_t)\right) +$$
$$+ \varepsilon\left(\frac{\|\mathscr{A}\|^2}{(1+\sigma)} - 1\right)\psi_{\mathscr{A}}(X_t) \le \left[\frac{1}{C^2} + \frac{2\sigma}{1-\sigma}\left(1 + \frac{1}{C}\right)^2\right]\psi_{\mathscr{A}}(X_t) +$$
$$+ \varepsilon\left[\frac{1}{C^2} + \frac{2\sigma}{1-\sigma}\left(1 + \frac{1}{C}\right)^2 + \frac{\left\|\mathscr{A}^{\mathrm{T}}\right\|^2}{1+\sigma} - 1\right]\psi_{\mathscr{A}}(X_t).$$

Полагая

$$D_0 = \frac{1}{C^2} + \frac{2\sigma}{1-\sigma}\left(1+\frac{1}{C}\right)^2 \quad \text{и} \quad D_1 = \frac{\left\|\mathscr{A}^{\mathrm{T}}\right\|^2}{1+\sigma} - 1,$$

окончательно получаем

$$\psi_{\mathscr{A}}(X_{t+1}) \le [D_0 + \varepsilon(D_0 + D_1)]\psi_{\mathscr{A}}(X_t). \tag{18}$$

4. ПРИМЕНЕНИЕ АЛГОРИТМОВ SVP И ASVP К ЗАДАЧЕ ВОСПОЛНЕНИЯ МАТРИЦ МАЛОГО РАНГА

Рассмотрим применение алгоритма ASVP к задаче восполнения матриц малого ранга по случайному набору элементов (на случайном шаблоне). С точки зрения анализа сходимости алгоритмов SVP и ASVP задача восполнения сводится к специальному способу выбора аффинного преобразования $\mathscr{A}$.

Пусть $X_* \in \mathbb{R}^{m\times n}$ – неизвестная матрица ранга k, причем $k \ll \min(m,n)$, а Ω – случайный набор пар индексов (i,j), где $i \in 1,2,\ldots,m$ и $j \in 1,2,\ldots,n$, причем любая пара (i,j) может быть выбрана равновероятно среди всех mn пар с вероятностью q, которую мы определим позже. Обозначим мощность множества Ω как $M = |\Omega|$, будем считать, что пары $(i,j) \in \Omega$ линейно упорядочены; будем также для простоты считать, что выполнено в точности $M = qmn$, а q будем также называть плотностью известных элементов матрицы.

Определим линейный оператор $\mathscr{S} : \mathbb{R}^{m\times n} \to \mathbb{R}^M$ по правилу

$$(\mathscr{S}(X))_k = \frac{1}{\sqrt{q}} X_{ij}, \quad k = 1,2,\ldots,M, \tag{19}$$

где $(\mathscr{S}(X))_k$ – компонента вектора $\mathscr{S}(X)$, соответствующая паре (i,j) в выбранном линейном упорядочении. Определим B формулой $B = \mathscr{S}(X_*) \in \mathbb{R}^M$ и зададим функционал $\psi_{\mathscr{S}}(X) = \frac{1}{2}\|\mathscr{S}(X) - \mathbf{B}\|_2^2$. Задача восполнения матрицы X_* по части ее элементов записывается в виде

$$\psi_{\mathscr{S}}(X) = \frac{1}{2}\|\mathscr{S}(X) - \mathbf{B}\|_2^2 \to \inf, \quad \operatorname{rank}(X) \le k. \tag{20}$$

Для упрощения записи в дальнейшем индекс $\mathscr{S}$ в обозначении функционала ψ опускается.

В качестве оператора $\hat{\mathscr{P}}_k$ при этом можно использовать произвольный алгоритм построения малоранговой аппроксимации $\tilde{A} = \hat{\mathscr{P}}_k A$, достаточно близкой к проекции с использованием сингулярного разложения

$$\left\|A - \tilde{A}\right\|_F \le (1+\varepsilon)\|A - A_k\|_F.$$

Нижний индекс k у матрицы здесь и далее обозначает наилучшее приближение ранга k по норме Фробениуса, которое можно получить с помощью сингулярного разложения.

При $\sigma \approx 0$ из следствия 2 подстановкой $\mathscr{A} = \mathscr{S}$, используя тривиальную оценку $\left\|\mathscr{S}^{\mathrm{T}}\right\|^2 \le \frac{1}{q}$, мы получаем условие

$$\varepsilon\left\|\mathscr{S}^{\mathrm{T}}\right\|^2 < 1, \quad \frac{\varepsilon}{q} < 1,$$

а потому для сходимости метода восполнения матриц достаточно $\varepsilon = O(q)$.

Отметим, что для задачи восполнения матрицы малого ранга по значениям ее элементов на разреженном шаблоне Ω напрямую воспользоваться результатами разд. 2 и 3 нельзя.

Так, например, в случае произвольного шаблона Ω оператор $\mathscr{S}(X)$, определенный ранее, не удовлетворяет условию ограниченной изометрии, которое является ключевым при доказательстве сходимости метода. Тем не менее, как показано в [7], если Ω имеет случайный характер, то условие ограниченной изометрии для оператора $\mathscr{S}$ выполняется с вероятностью, неотличимой от 1, на почти всем множестве матриц ранга, не превосходящего $2k$. Чтобы сформулировать соответствующее утверждение, нам потребуется понятие μ-*некогерентных матриц*.

Определение 1. Пусть для матрицы $X \in \mathbb{R}^{m\times n}$ задано ее сингулярное разложение $X = U\Sigma V^{\mathrm{T}}$. Матрица X называется μ-*некогерентной*, если для матриц сингулярных векторов U и V справедливы неравенства

$$\max_{i,j}|U_{ij}| \le \frac{\sqrt{\mu}}{\sqrt{m}}, \quad \max_{i,j}|V_{ij}| \le \frac{\sqrt{\mu}}{\sqrt{n}}.$$

Используя определение 1, сформулируем следующий результат о выполнении свойства ограниченной изометрии для оператора $\mathcal{S}$.

Теорема 3 [Теорема 4.2 из [7]]. *Существует константа $C \ge 0$ такая, что для любого $0 < \sigma < 1$, любых $\mu \ge 1$ и $n \ge m \ge 3$, и для любого шаблона разреженности Ω, пары индексов которого выбираются случайно в соответствии с моделью Бернулли с параметром*

$$q \ge C\frac{\mu^2 k^2}{\sigma^2}\frac{\log(n)}{m},$$

определяющим вероятность для пары индексов (i, j) принадлежать маске Ω, оператор $\mathcal{S}$ удовлетворяет на множестве всех μ-некогерентных матриц свойству ограниченной изометрии вида

$$(1-\sigma)\|X\|_F^2 \le \|\mathcal{S}(X)\|_F^2 \le (1+\sigma)\|X\|_F^2$$

с вероятностью не меньше $1 - \exp(-n\log(n))$.

Из теорем 2 и 3 следует сходимость алгоритмов SVP и ASVP в случае, если на всех итерациях (т.е. для всех t) матрицы $X_{t+1} - X_t$ и $X_t - X_*$ обладают ограниченной константой некогерентности μ. Отметим, что для разности $X_{t+1} - X_t$ формальная проверка μ-некогерентности является возможной и вычислительно недорогой процедурой. В то же время для разности $X_t - X_*$ на практике оценить константу некогерентности невозможно.

Из сказанного выше следует, что для задачи восполнения матриц малого ранга, заданных на случайном шаблоне, обоснование алгоритмов SVP и ASVP не является безусловным. Формальные реализации этих методов могут и проявляют нерегулярное поведение, т.е. расходятся при выборе теоретически обоснованного шага градиентного спуска τ (см., например, [4]) или в том случае, когда сингулярные числа решения X_* быстро убывают.

На практике получение (быстро) сходящихся реализаций алгоритмов возможно только с использованием дополнительных приемов, среди которых мы выделяем два: последовательный набор ранга приближения и выбор шага τ в градиентном методе.

4.1. Последовательный набор ранга приближения

Этот прием необходим в ситуациях, когда сингулярные числа решения X_* задачи восполнения быстро убывают, или когда ранг X_* заранее неизвестен. Причиной нерегулярного поведения алгоритмов SVP и ASVP алгоритмов в этом случае является существование решений задачи восполнения с большим рангом и обладающих большой константой некогерентности.

Рассмотрим искусственный, но проясняющий ситуацию пример. Пусть $E \in \mathbb{R}^{m\times n}$ – матрица ранга 1, каждый элемент которой равен единице. Такая матрица является μ-некогерентной с параметром некогерентности $\mu = 1$. Зададим $M \ll \min(m, n)$ пар индексов (i, j) и поставим задачу восполнения матрицы E среди матриц ранга M. Среди возможных решений этой задачи существует разреженное решение ранга M, все исходно неизвестные компоненты которого равны нулю. Однако такое решение имеет константу некогерентности порядка $\max\left(\sqrt{m}, \sqrt{n}\right)$.

Чтобы избежать нерегулярного поведения ASVP в рассматриваемых ситуациях, предлагается использовать последовательный набор ранга приближений. Другими словами, вместо того, чтобы решать задачу восполнения сразу для предписанного ранга k, задачу восполнения предлагается последовательно решать для рангов 1, 2, ..., k, и для каждого последующего ранга решение, соответствующее предыдущему рангу, использовать в качестве начального приближения.

Наиболее примитивный подход к набору ранга решения состоит в том, чтобы заранее фиксировать время работы алгоритма и отвести константные отрезки времени на работу алгоритма с

каждым рангом приближения k. При этом можно, например, отвести первую половину всего времени работы алгоритма на стадию набора ранга, а вторую – на итерации с необходимым конечным рангом приближения.

Другой более обоснованный способ набора ранга может быть основан на проверке выполнения условия μ-некогерентности для разности $X_t - X_{t-1}$. Анализ численных экспериментов показал, что при нерегулярном поведении алгоритмов SVP и ASVP параметр μ-некогерентности последовательности матиц $X_{t+1} - X_t$ увеличивается от итерации к итерации и стремится к максимально возможному значению $\max\left(\sqrt{m}, \sqrt{n}\right)$. Таким образом, на каждой итерации можно вычислять текущее значение некогерентности $X_{t+1} - X_t$ и проводить увеличение ранга приближения на единицу только в случае, если на протяжении нескольких итераций эта величина оказывается достаточно малой; для определения допустимой величины некогерентности полезно иметь априорную оценку на эту величину для искомой матрицы.

Сложность вычисления μ для матрицы $X_t - X_{t-1}$ определяется сложностью вычисления сингулярного разложения объединенных левых и объединенных правых факторов матриц X_t, X_{t-1}, что в случае $m = n$ требует $O(nk^2) \ll n^3$ операций.

При этом точно указать необходимое число последовательных шагов алгоритма, при которых величина некогерентности $X_t - X_{t-1}$ остается ниже допустимой границы, затруднительно; на основе проведенных экспериментов установлено, что в случае матриц порядка 1000 и случайных факторов достаточно около пяти шагов. Это число последовательных шагов можно оценивать по числу предшествующих шагов, при которых величины некогерентности были, наоборот, большими, например, используя следующую процедуру.

Алгоритм 1

Входные данные: граничное значение μ_{crit}, по которому определяется, что некогерентность на текущей итерации достаточно мала. Константа минимального числа последовательных итераций с низкой некогерентностью s_{crit}. Вводятся переменные, сохраняющие число предшествующих итераций SVP (ASVP) с большими и малыми величинами μ: $s_{\text{high}} = 0$, $s_{\text{low}} = 0$.

Цель: ранг k увеличивается, если на протяжении нескольких итераций $\mu < \mu_{\text{crit}}$.

1: если $\mu(X_t - X_{t-1}) > \mu_{\text{crit}}$, то

2: $\quad s_{\text{high}} := s_{\text{high}} + 1$

3: $\quad s_{\text{crit}} := \max(s_{\text{high}}, s_{\text{crit}})$

4: иначе

5: $\quad s_{\text{low}} := s_{\text{low}} + 1$

6: если $s_{\text{low}} > s_{\text{crit}}$, то

7: $\quad k := k + 1$

8: $\quad s_{\text{low}} := 0$

9: $\quad s_{\text{high}} := 0$

Такой способ набора ранга оказался особенно эффективен для варианта ASVP, основанного на псевдоскелетном методе с использованием подматриц большого объема (см. далее в п. 5.2).

4.2. Выбор параметра шага градиентного метода

Численные эксперименты показывают, что нерегулярного поведения алгоритмов SVP, ASVP можно избежать путем понижения параметра шага градиентного метода τ. Так, теорема 2 гарантирует, что в условиях теоремы метод SVP сходится геометрически при $\tau = 1/(1 + \sigma) \approx 3/4$ с учетом того, что, согласно определению оператора задачи восполнения матриц $\mathcal{P}$, все элементы на маске Ω скалируются на константу $1/\sqrt{q}$, использование такого шага интуитивно неустойчиво при критически малых значениях плотности известных элементов q. На практике оказывается, что при таком шаге в случае малых q и быстром падении сингулярных чисел искомой матрицы алгоритмы SVP, ASVP могут расходиться, но при тех же условиях и малом шаге порядка q расхо-

димости не наблюдается. Экспериментально установлено, что при шаге τ в пределах от $[q, 2q]$ при достаточно малых q сходимость SVP, ASVP наблюдается всегда.

Случаи расходимости SVP метода с большим шагом были замечены уже в [4]. Для повышения устойчивости алгоритма ее авторы также предложили выбирать τ из интервала $[q, 2q]$. Не проводя формального доказательства сходимости алгоритма SVP с шагом $\tau = q$, приведем некоторые соображения в пользу его устойчивости.

Действительно, если X_t и X_{t+1} – приближения с номерами t и $t+1$ и $B = S(X_*)$, то шаг градиентного метода $\tau = q$ соответствует присваиванию каждому элементу текущего приближения, лежащему на маске известных элементов, соответствующего известного элемента. Это значит, что

$$\|X_t - X_*\|_{F,\Omega} = \|X_t - Y_{t+1}\|_F \geq \|Y_{t+1} - X_{t+1}\|_F \geq \|X_{t+1} - X_*\|_{F,\Omega}, \tag{21}$$

где $\|X\|_{F,\Omega} = \|\mathcal{P}(X)\|_F$. Неравенство (21) можно записать в эквивалентной форме:

$$\psi(X_{t+1}) \leq \psi(X_t),$$

откуда имеем, что невязка на итерациях SVP алгоритма с шагом $\tau = q$ монотонно не возрастает.

Тем не менее эксперименты показывают, что в случаях, когда при использовании шага $\tau \approx 3/4$ на практике не нарушаются гипотезы о некогерентности, сходимость алгоритмов SVP, ASVP с шагом $\tau \approx 3/4$ значительно быстрее, чем с шагом $\tau \approx q$, т.е. на практике использование большого шага является предпочтительным. Чтобы на практике использовать преимущество большого шага в скорости сходимости, но не допускать нерегулярного поведения алгоритмов, предлагается использовать адаптивные процедуры изменения шага τ по ходу алгоритма. Для этого предлагается рассматривать изменение невязки $\|\mathcal{P}(X_t - X_*)\|_F$ между итерациями SVP, ASVP. Если отношение невязок на итерациях $t+1$ и t

$$\alpha_{t+1,t} := \frac{\|\mathcal{P}(X_{t+1} - X_*)\|_2}{\|\mathcal{P}(X_t - X_*)\|_2} > \alpha_{\text{crit}} > 1,$$

то делается вывод, что алгоритм начал вести себя нерегулярно, в связи с чем шаг градиентного метода понижается, а последняя итерация пересчитывается с уменьшенным шагом. Формально после выполения итерации $t+1$ алгоритмов SVP, ASVP выполняется следующий

Алгоритм 2

Входные данные: пограничная величина $\alpha_{\text{crit}} > 1$; стартовое значение шага $\tau = 3/4$; константы увеличения и уменьшения шага $\beta_{\text{inc}} < 1$, $\beta_{\text{dec}} < 1$.

Цель: шаг τ изменяется в зависимости от роста или падения погрешности.

1: если $(\alpha_{t+1,t} > \alpha_{\text{crit}})$, то

2: $\quad \tau := \tau + \beta_{\text{dec}}(q - \tau)$

3: $\quad$ Итерация $t+1$ проводится заново с уменьшенным шагом.

4: иначе

5: $\quad \tau := \tau + \beta_{\text{inc}}\left(\frac{3}{4} - \tau\right)$.

Этот адаптивный алгоритм выбора шага оказался эффективен на практике для ASVP с обоими вариантами метода приближенного проектирования, которые будут рассмотрены далее. Однако вычисление невязки может быть асимптотически сложной процедурой, так как требует не менее $O(mnqk)$ операций, поэтому возможно использовать эту процедуру обновления величины шага не после каждой итерации, а например, после каждой десятой итерации ASVP.

5. ВОЗМОЖНЫЕ МЕТОДЫ ПРИБЛИЖЕННОГО ПРОЕКТИРОВАНИЯ ASVP

5.1. Проектирование на случайные подпространства

Рассмотрим один из возможных методов случайного проектирования, необходимых для построения алгоритма ASVP. Будем использовать способ приближенного вычисления частичного

сингулярного разложения на основе проектирования на случайные подпространства. Такой способ введен и исследован в [9]. Будем использовать вспомогательную матрицу

$$J \in \mathbb{R}^{n\times l}, \quad l = k + p, \quad p > 0,$$

все элементы которой выбираются независимо и случайно по стандартному нормальному распределению. Тогда в качестве проектора $\hat{\mathcal{P}}$ предлагается использовать

$$\hat{\mathcal{P}}_k(Y) = Q(Q^{\mathrm{T}}Y)_k, \quad Y \in \mathbb{R}^{m\times n},$$

где матрица Q определяется как ортогональный базис столбцов YJ: $YJ = QR$, а $(Q^{\mathrm{T}}Y)_k$ – наилучшее приближение матрицы $(Q^{\mathrm{T}}Y)$ матрицей ранга k, вычисляемое с помощью сингулярного разложения вытянутой матрицы. Из леммы 6.1 в [18], которая обобщает результаты из [9], следует, что для любой матрицы $Y \in \mathbb{R}^{m\times n}$ справедлива

Лемма 1 (см. [18]):

$$\mathbb{E}_J\left\|Y - Q(Q^{\mathrm{T}}Y)_k\right\|_F^2 \le \left(1 + \frac{k}{p-1}\right)\|Y - Y_k\|_F^2. \tag{22}$$

В [9], [18] приведены оценки на отклонение от матожидания, которые опустим в силу их громоздкости. Таким образом, в [9] предложен следующий алгоритм вычисления приближенного частичного сингулярного разложения матрицы Y.

1. Создается случайная матрица $J \in \mathbb{R}^{n\times l}$.
2. Вычисляется произведение YJ.
3. Для произведения YJ вычисляется QR-разложение: $YJ = QR$.
4. Вычисляется произведение $Q^{\mathrm{T}}Y$.
5. Вычисляется сокращенное сингулярное разложение для вытянутой матрицы

$$(Q^{\mathrm{T}}Y)_k = U\Sigma V^{\mathrm{T}}, \quad \text{где} \quad U \in \mathbb{R}^{l\times k}, \quad V^{\mathrm{T}} \in \mathbb{R}^{k\times n}.$$

6. Так как

$$(QU)\Sigma V^{\mathrm{T}} = Q(Q^{\mathrm{T}}Y)_k,$$

то матрицы QU и V^{T} являются искомым приближением сингулярных векторов матрицы Y.

Согласно оценке (22), введенный таким образом оператор $\hat{\mathcal{P}}$ удовлетворяет определению оператора приближенного проектирования (4) с точностью $\varepsilon = k/(p-1)$ в среднем. Оценим сложность одной итерации полученного приближенного алгоритма ASVP.

• Умножение на случайную матрицу: так как приближенное сингулярное разложение вычисляется для суммы матрицы малого ранга (матрица предыдущей итерации) и разреженной матрицы (градиент), имеем $O\big((n+m)kl + mnlq\big)$ операций.

• QR-разложение YJ: $O(ml^2)$ операций.

• Умножение на матрицу Q^* слева: аналогично первому пункту, $O\big((n+m)kl + mnlq\big)$ операций.

• Сингулярное разложение вытянутой матрицы: $O(nl^2)$ операций.

• Восстановление левого сингулярного базиса: $O(ml^2)$ операций.

Таким образом, имеем суммарную сложность одной итерации приближенного алгоритма SVP $O\big((m+n)l^2 + mnlq\big)$. С учетом оценки $\varepsilon = O\left(\frac{k}{p}\right)$ можно считать, что при малых ε выполнено $p \gg k$, $l \approx p$, а вычислительная сложность одной итерации соответственно оценивается величиной

$$O\big((m+n)p^2 + mnpq\big).$$

5.2. Псевдоскелетный метод на матрицах большого проективного объема

Опишем метод ASVP с использованием в качестве проектора $\hat{\mathcal{P}}_k$ аппроксимации на основе крестового (псевдоскелетного) CGR-приближения, в котором используются строки $R \in \mathbb{R}^{p\times n}$ и столбцы $C \in \mathbb{R}^{m\times p}$ приближаемой матрицы

$$Y = X_t - \tau\mathcal{S}^{\mathrm{T}}(\mathcal{S}(X_t) - \mathbf{B}).$$

Один из самых быстрых способов построения аппроксимации – построение Fast CGR из [17].

Построение аппроксимации состоит из следующих этапов.

1. Использование алгоритма `maxvol` (см. [16]) **для поиска подматрицы** $\hat{Y} \in \mathbb{R}^{2k\times 2k}$.

Увеличенный в 2 раза ранг позволяет позже использовать усеченное сингулярное разложение для увеличения точности аппроксимации (по сравнению с прямым поиском аппроксимации ранга k), а также позволяет алгоритму не “зацикливаться” на одной и той же подматрице.

Эксперименты в [17] показывают, что в случае прямого построения аппроксимации $\tilde{Y} = C\hat{Y}^{-1}R$, коэффициент ε растет линейно с ростом r, поэтому нельзя гарантировать произвольную точность приближения ε, ограничиваясь лишь алгоритмом `maxvol`.

При фиксированном числе шагов алгоритм требует $O\left((m+n)k^2\right)$ операций и знания $O(k)$ строк и столбцов матрицы.

2. Увеличение числа строк и столбцов до p.

Новые строки и столбцы выбираются с помощью ускоренного варианта алгоритма `maxvol2` (см. [15]). Расширение требует лишь знания уже найденных с помощью `maxvol` строк и столбцов и занимает $O((m+n)kp)$ операций.

Согласно гипотезе из [17], для аппроксимации ранга $2k$ с p столбцами справедлива оценка коэффициента

$$1 + \varepsilon \le 1 + \frac{k}{p-k+1}, \tag{23}$$

а потому достаточно набрать $p = O(k/\varepsilon)$ строк и столбцов. При этом в качестве ядра CGR-аппроксимации выбирается матрица $G = \hat{Y}_{2k}^{+}$, для чего потребуется сингулярное разложение матрицы $\hat{A}$, занимающее $O(p^3)$ операций.

3. Сингулярное разложение аппроксимации.

Найденная на предыдущем шаге CGR-аппроксимация ранга $2k$ подвергается сингулярному разложению с сохранением лишь k максимальных сингулярных чисел.

Сингулярное разложение произведения $C\hat{Y}_{2k}^{+}R$ требует $O((m+n)kp)$ операций и выполняется в следующем порядке.

(а) Факторы $\hat{U} \in \mathbb{R}^{p\times 2k}$ и $\hat{V}^{\mathrm{T}} \in \mathbb{R}^{2k\times p}$ сингулярного разложения матрицы $\hat{Y}_{2k}^{+}$ умножаются на C и R соответственно.

(б) Для матрицы $C\hat{U}$ вычисляется представление в виде произведения матрицы с ортогональными столбцами и верхней треугольной матрицы; для матрицы $\hat{V}^{\mathrm{T}}R$ вычисляется представление в виде произведения нижней треугольной матрицы и матрицы с ортогональными строками.

(в) Выполняется сингулярное разложение произведения двух полученных треугольных матриц, с помощью которого находится оптимальное приближение этого произведения матрицей ранга k.

Полученная таким образом аппроксимация ранга k найденной на предыдущем шаге матрицы CGR считается искомым приближением для Y_k.

Таким образом, полная сложность алгоритма аппроксимации составляет $O\left((m+n)kp + p^3\right) = O\left((m+n)p^2q + p^3\right)$. Данная аппроксимация может быть использована в качестве оператора приближенного проектирования $\hat{\mathcal{P}}_k$ для алгоритма ASVP.

Полученное CGR-разложение записывается в виде $CGR = X_{t+1} = U_{t+1}V_{t+1}^{\mathrm{T}}$ с $U_{t+1} \in \mathbb{R}^{m\times k}$, $V_{t+1}^{\mathrm{T}} \in \mathbb{R}^{k\times n}$.

Напомним, что приближенное проектирование (аппроксимация) проводится для матрицы вида

$$Y = X_t - \tau\mathcal{S}^{\mathrm{T}}(\mathcal{S}(X_t) - \mathbf{B}).$$

Всего нам требуется $O(p)$ ее строк и столбцов, которые легко найти на основе матрицы $X_t = U_tV_t^{\mathrm{T}}$ за $O\left((m+n)kp\right)$ операций, что не увеличивает асимптотическую сложность алгоритма.

Заметим, что на практике число шагов алгоритма `maxvol` можно считать константой, так как по ходу итераций алгоритма ASVP приближение каждой итерации меняется медленно, откуда имеем, что и объем подматриц приближения меняется слабо. Алгоритм tmaxvol основан на максимизации объема, а потому будет слабо реагировать на небольшие изменения входных данных.

Более того, на основе сингулярных чисел, которые отбрасываются на стадии сингулярного разложения аппроксимации CGR, можно судить о скорости сходимости алгоритма и иногда вообще не изменять выбранные ранее строки и солбцы.

В численных экспериментах использовались

$$p = 2k + \left\lceil 0.7\frac{k}{q} \right\rceil$$

строк и столбцов. Это число намеренно ниже оценки (23): численные эксперименты из [17] показали, что реальные значения погрешности гораздо ниже верхних оценок, хотя те и предсказывают верную асимптотическую зависимость.

6. СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ АЛГОРИТМОВ ASVP И SVP

Рассмотрим соотношения сложностей алгоритма SVP с использованием полного сингулярного разложения матрицы и алгоритма ASVP с использованием приближенного проектирования матрицы. Для анализа отметим следующие особенности алгоритмов.

1. Доказательство сходимости алгоритмов SVP и ASVP базируется на использовании свойства ограниченной изометрии (3) рассматриваемого оператора A.

2. В случае, если A – оператор задачи матричного восполнения, существует асимптотическая оценка на минимальное количество известных элементов матрицы, которого достаточно для полного восполнения. Эта оценка определяется размерами неизвестной матрицы, рангом неизвестной матрицы, и “некогерентностью” неизвестной матрицы, численной величиной, характеризующей разреженность сингулярных факторов матрицы.

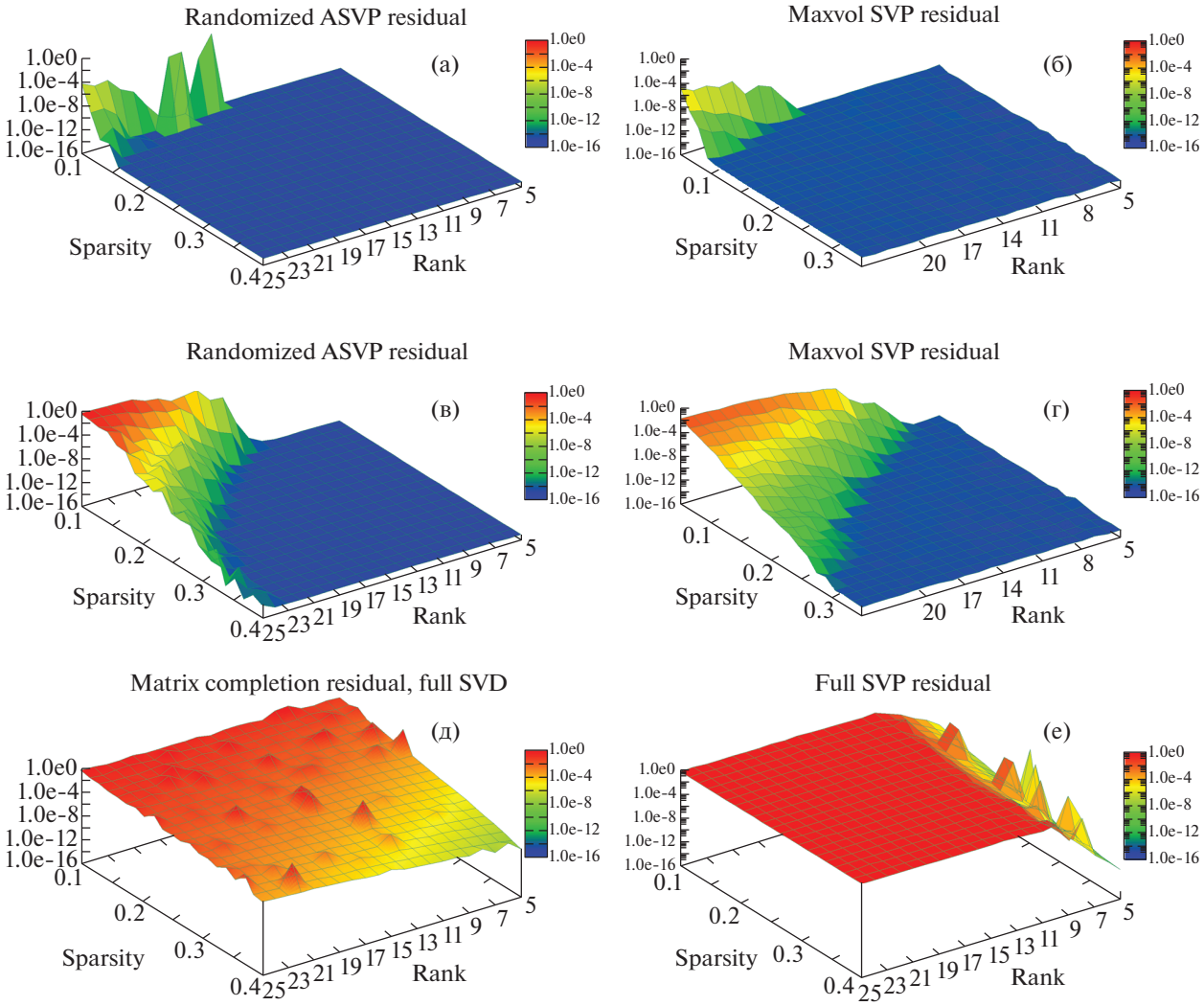
3. Не обязательно использовать именно такой порядок числа известных элементов матрицы, допустимо использовать и большие порядки.

4. В случае алгоритма ASVP гарантируется геометрическая сходимость алгоритма с порядком не менее $O\left(\frac{\epsilon}{q}\right)$, где q – плотность известных элементов матрицы, а ε – относительная ошибка приближенного проектирования. Таким образом, при уменьшении числа известных элементов матрицы требуется большая точность приближенного проектирования.

5. И в случае ASVP с использованием проектирования на случайные подпространства, и в случае ASVP с использованием псевдоскелетного метода, вводится дополнительный параметр p, определяющий некую вспомогательную размерность метода, причем с увеличением p увеличивается и точность, и вычислительная сложность приближенного проектирования, и наоборот.

6. Таким образом, с уменьшением плотности известных элементов матрицы требуется большая точность приближения, что приводит к увеличению вспомогательных размерностей рассматриваемых методов случайного проектирования, что может привести к увеличению сложности алгоритма ASVP.

Проведем формальное сравнение алгоритмов. Сложность SVP и ASVP удобно выразить через параметры m, n, k, q, p. Кроме того, $q \geq q_{\min}(m,n,k,\mu)$; также и для псевдоскелетного метода, и



Фиг. 1. Графики итоговой относительной невязки $\|\mathcal{P}(X_t - X_*)\|_2 / \|\mathcal{P}(X_*)\|_2$ восполнения матриц с сингулярными числами вида $\sigma_i = 1$: (а), (в) – ASVP со случайным проектированием, ограничение по времени в 50 и в 10 итераций SVP соответственно; (б), (г) – псевдоскелетный ASVP, ограничение по времени в 50 и в 10 итераций SVP соответственно; (д), (е) – полный SVP (50 итераций) с параметрами шага $\tau = q$ и $\tau = 3/4$ соответственно.

для случайного проектирования выполнено $\epsilon = O\left(\frac{k}{p}\right)$; считая множитель геометрической сходимости константой, чтобы обеспечить константный порядок числа итераций алгоритма, имеем $\epsilon = O(q)$, откуда $p = O\left(\frac{k}{q}\right)$.

Алгоритм SVP состоит из двух основных операций, оценим их сложность.

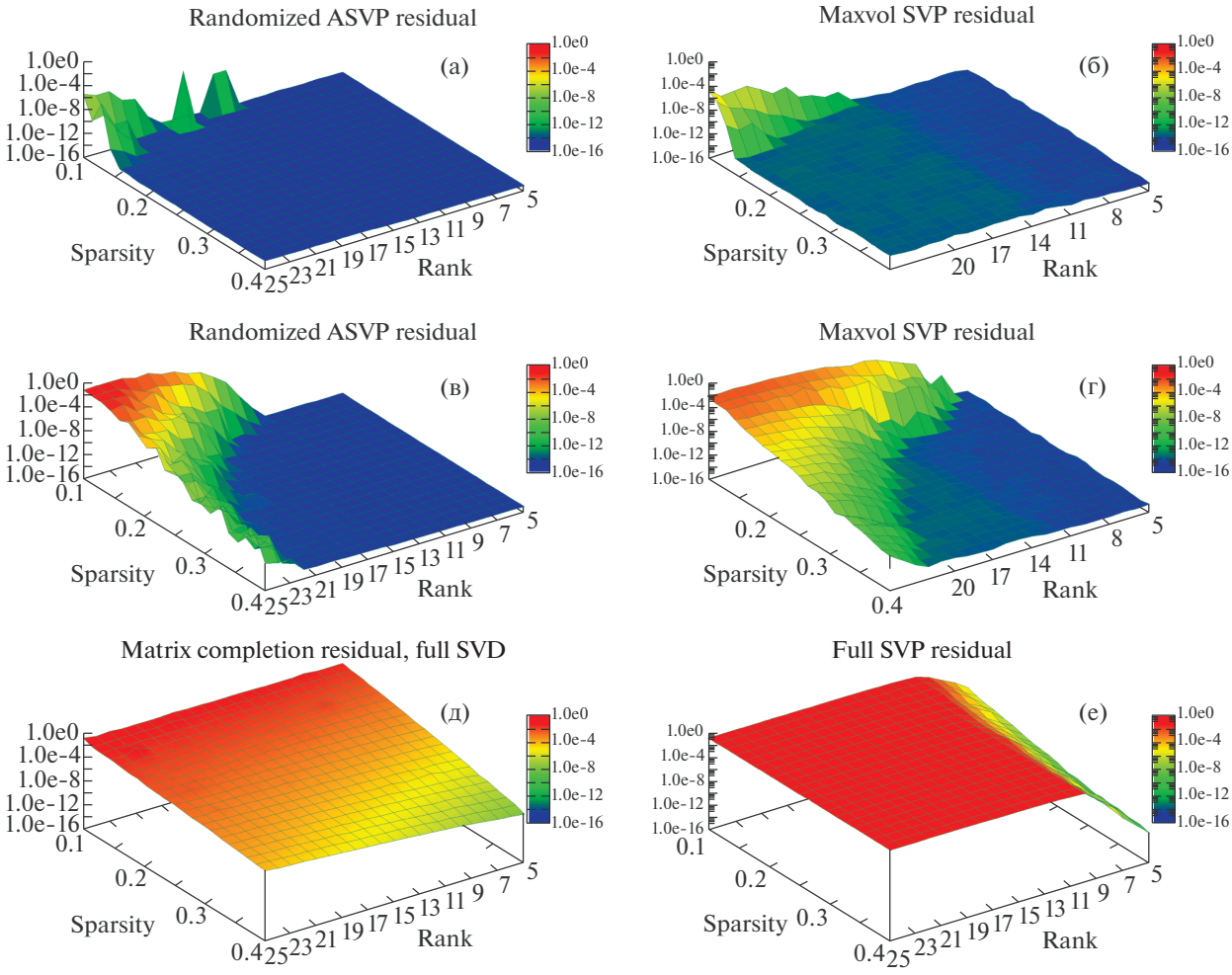
1. Вычисление невязки на маске известных элементов покомпонентно: $O(k)$ операций на каждый элемент, $O(mnkq)$ операций всего.

2. Вычисление сингулярного разложения для получения приближения следующей итерации: $O(mn\min(m,n))$.

Так как $q < 1$, $k < \min(m,n)$, имеем сложность одной итерации алгоритма $O(mn\min(m,n))$, кубическую в случае квадратной матрицы.

Рассмотрим полученные выше оценки на сложность итерации алгоритма ASVP.

- $O((m+n)p^2) + O(mnpq)$ для случайного проектирования,



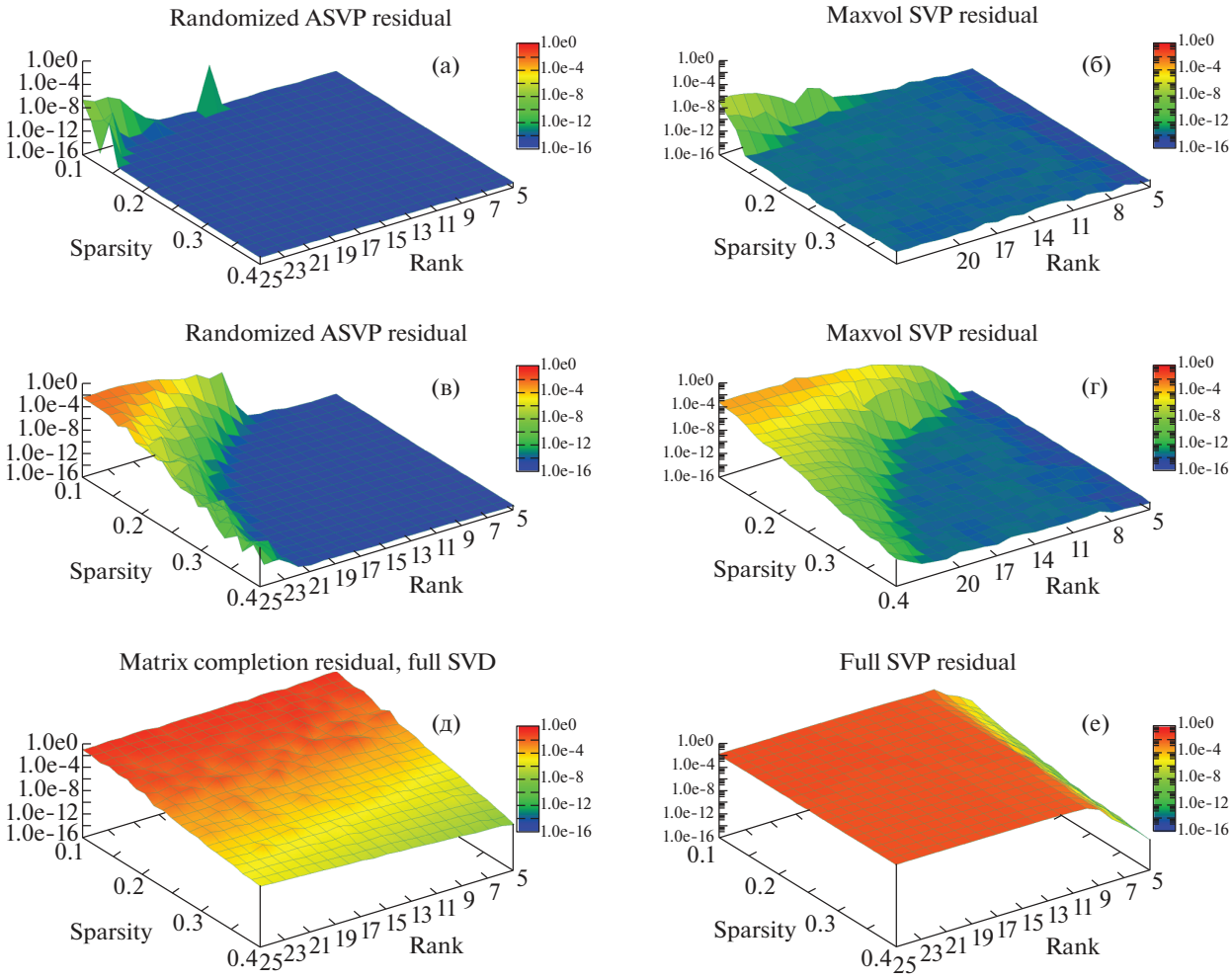
Фиг. 2. Графики итоговой относительной невязки $\|\mathcal{P}(X_t - X_*)\|_2 / \|\mathcal{P}(X_*)\|_2$ восполнения матриц с сингулярными числами вида $\sigma_i = \frac{1}{i}$: (а), (в) – ASVP со случайным проектированием, ограничение по времени в 50 и в 10 итераций SVP соответственно; (б), (г) – псевдоскелетный ASVP, ограничение по времени в 50 и в 10 итераций SVP соответственно; (д), (е) – полный SVP (50 итераций) с параметрами шага $\tau = q$ и $\tau = 3/4$ соответственно.

- $O((m+n)kp) + O(p^3)$ для псевдоскелетного метода.

С учетом $q = O(\epsilon) = O\left(\frac{k}{p}\right)$ для метода случайного проектирования имеем, что асимптотика сложности одной итерации псевдоскелетного метода в $O\left(\min(m,n)/p\right)$ раз меньше сложности одной итерации метода случайного проектирования.

Теперь рассмотрим сложности ASVP в зависимости от различных q. Пусть $n \geq m$. В случае минимально возможного $q = q_{\min} = O\left(\frac{k^2\mu^2 \log n}{m}\right)$ имеем сложность порядка

- $O\left(\frac{(m+n)m^2}{k^2\mu^4 \log^2 n}\right) + O(mnk)$ для случайного проектирования;
- $O\left(\frac{m^3}{\mu^6 k^3 \log^3 n}\right) + O\left(\frac{(m+n)m}{\mu^2 \log n}\right)$ для псевдоскелетного метода.



Фиг. 3. Графики итоговой относительной невязки $\|\mathcal{P}(X_t - X_*)\|_2 / \|\mathcal{P}(X_*)\|_2$ восполнения матриц с сингулярными числами вида $\sigma_i = \frac{1}{i^2}$: (а), (в) – ASVP со случайным проектированием, ограничение по времени в 50 и в 10 итераций SVP соответственно; (б), (г) – псевдоскелетный ASVP, ограничение по времени в 50 и в 10 итераций SVP соответственно; (д), (е) – полный SVP (50 итераций) с параметрами шага $\tau = q$ и $\tau = 3/4$ соответственно.

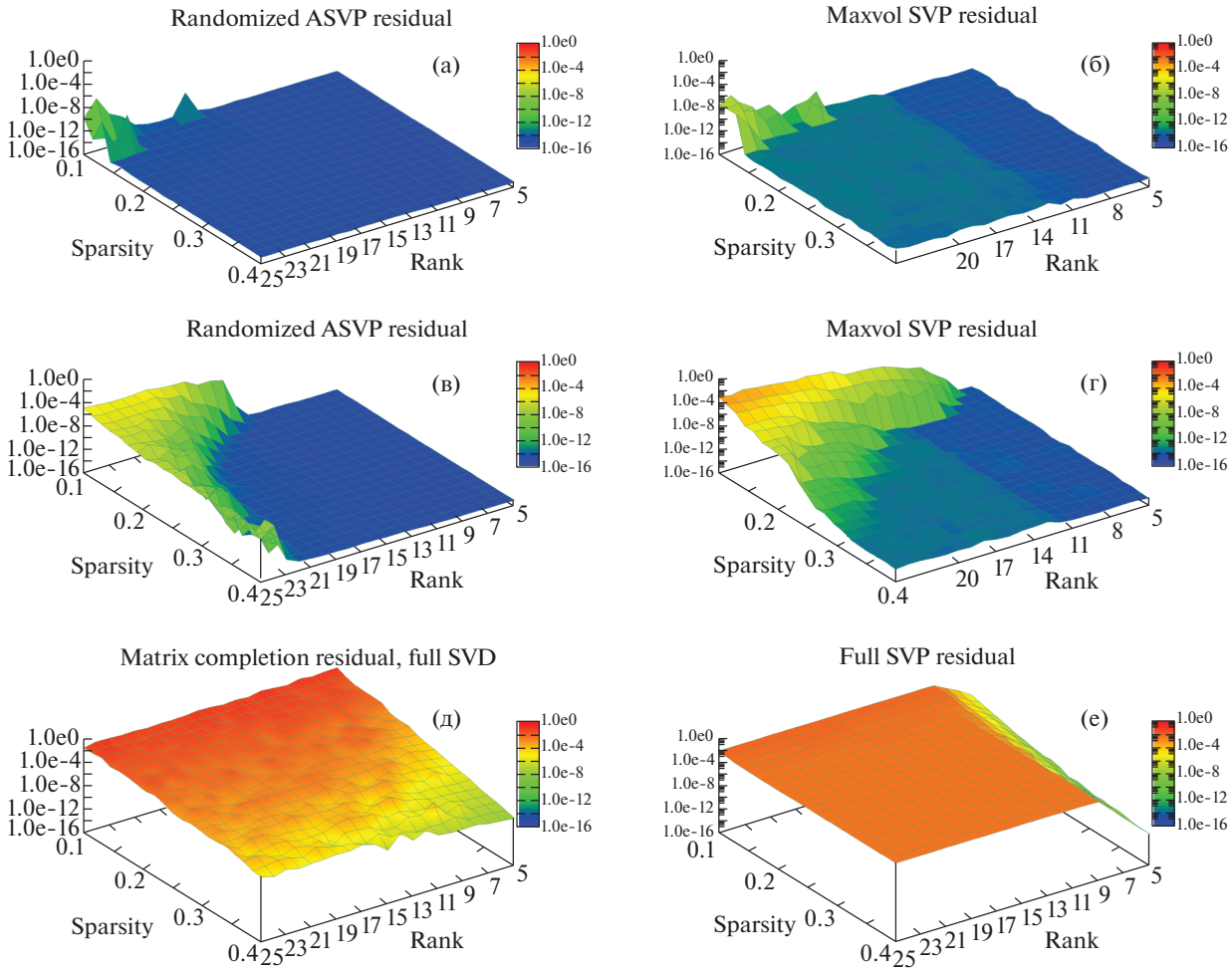
Покажем, что при больших порядках q можно достичь и более низкой сложности ASVP. Пусть $q = m^{-1/2} \geqslant \dfrac{k^2\mu^2 \log n}{m}$ (при малых порядках μ, близких к константе или к логарифму от размеров матрицы); тогда подстановкой имеем сложность

- $O((m+n)mk^2)$ для случайного проектирования;
- $O((m+n)m^{1/2}k^2) + O(m^{3/2}k^3)$ для псевдоскелетного метода.

7. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

Проведем экспериментальное сравнение алгоритмов SVP и ASVP с двумя рассмотренными вариантами методов приближенного проектирования.

- Рассматриваются квадратные матрицы размера $m = n = 1000$. Рассматриваются ранги матриц k от 5 до 25 и плотности распределения известных элементов q от 0.1 до 0.4.



Фиг. 4. Графики итоговой относительной невязки $\|\mathcal{S}(X_t - X_*)\|_2 / \|\mathcal{S}(X_*)\|_2$ восполнения матриц с сингулярными числами вида $\sigma_i = \frac{1}{2^i}$: (а), (в) – ASVP со случайным проектированием, ограничение по времени в 50 и в 10 итераций SVP соответственно; (б), (г) – псевдоскелетный ASVP, ограничение по времени в 50 и в 10 итераций SVP соответственно; (д), (е) – полный SVP (50 итераций) с параметрами шага $\tau = q$ и $\tau = 3/4$ соответственно.

• Матрицы формируются случайно следующим образом.

– Формируются случайные матрицы $\hat{U}_k \in \mathbb{R}^{m\times k}$ и $\hat{V}_k \in \mathbb{R}^{n\times k}$, каждый элемент которых распределен случайно по стандартному нормальному распределению.

– Вычисляются ортогональные базисы U_k и V_k в пространствах столбцов $\hat{U}_k$ и $\hat{V}_k$ соответственно, например, с помощью QR-разложения. Экспериментально установлено, что некогерентность факторов U_k, V_k, выбранных таким образом, растет логарифмически от m, n.

– Искомая матрица X_* задается в виде $X_* = U_k\Sigma_k V_k^{\mathrm{T}}$ с матрицей Σ_k сингулярных значений $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots$. Экспериментально установлено, что сходимость методов SVP, ASVP зависит от характера убывания сингулярных чисел σ_j; в экспериментах рассмотрены следующие законы σ_j: $\sigma_i = 1$ (фиг. 1), $\sigma_i = 1/i$ (фиг. 2), $\sigma_i = 1/i^2$ (фиг. 3), $\sigma_i = 1/2^i$ (фиг. 4).

• Сравнение проводится с ограничением по времени практической работы алгоритмов. Ограничение задается константой, которая определяется временем выполнения десяти или пятидесяти итераций алгоритма SVP с полным SVD; предполагается, что за это время приближенный алгоритм ASVP выполнит большее число итераций.

Ниже приведены графики зависимостей невязки от q и k при фиксированном размере матрицы в случае алгоритма SVP и двух рассмотренных вариантов алгоритма ASVP при различных законах падения сингулярных чисел неизвестной матрицы. Значения полных относительных ошибок по всей матрице на практике во всех экспериментах отличаются от относительных невязок на маске известных элементов не более, чем в два раза.

Графики сходимости в случае полного SVP приведены в двух вариантах: с шагом q (фиг. 1д–4д) и с шагом 3/4 (фиг. 1е–4е). Так как рассматривалось ограничение в 50 итераций SVP, вариант SVP с большим шагом на большой части графиков не достиг сходимости, так как такого числа итераций оказалось недостаточно для завершения последовательного набора ранга.

8. ЗАКЛЮЧЕНИЕ

В данной статье рассмотрено применение методов приближенного вычисления частичного сингулярного разложения к существующему итерационному алгоритму восполнения матриц малого ранга, сходящемуся геометрически, основной операцией которого является сингулярное разложение матрицы с последующим отсечением всех неглавных компонент (SVD-проекция) на каждой итерации. Для этого алгоритма получен теоретический результат, при определенных предположениях гарантирующий сохранение геометрической сходимости даже в случае замены точного вычисления SVD-проекции на приближенное. Сформулировано условие приближения, выполнения которого достаточно для геометрической сходимости при произвольном методе получения самой приближенной проекции. Рассмотрены два возможных варианта таких методов, удовлетворяющих этому условию в среднем или с высокой вероятностью. Для обоих этих методов проведены численные эксперименты и получены похожие результаты, показывающие большую вычислительную эффективность использования приближенных SVD-проекций.

СПИСОК ЛИТЕРАТУРЫ

1. *Kang Z., Peng C., Cheng Q.* Top-N Recommender System via Matrix Completion // arXiv preprint arXiv:1601.04800v1, 01/2016.
2. *Ahmed A., Romberg J.* Compressive multeplexing of correlated signals // IEEE Trans. Inform. Theory. 2015. V. 61. № 1. P. 479–498.
3. *Davies M., Eldar Y.* Rank awareness in joint sparse recovery // IEEE Trans. Inform. Theory. 2012. V. 58. № 2. P. 1135–1146.
4. *Hu R., Tong J., Xi J., Guo Q., Yu Y.* Low-Complexity and Basis-Free Channel Estimation for Switch-Based mmWave MIMO Systems via Matrix Completion. arXiv preprint arXiv:1609.05693v2[cs.IT], 11/2016.
5. *Argyriou A., Evgeniou T., Pontil M.* Convex multi-tasks feature learning // Machine Learn. 2008. V. 73. № 3. P. 243–272.
6. *Blei D.* Probabilistic topics models // Comm. ACM. 2012. V. 55. № 4. P. 77–84.
7. *Inderjit S. Dhillon Paghu Meka, Prateek Jain.* Guaranteed rank minimization via singular value projection // arXiv preprint arXiv:0909.5457, 09/2009.
8. *Tanner J., Wei K.* Normalized iterative hard thresholding for matrix completion // SIAM J. Sci. Comput. 2013. V. 35. № 5. P. 104–125.
9. *Tropp J.A., Halko N., Martinsson P.G.* Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions // SIAM Rev. 2011. V. 53. № 2. P. 217–288.
10. *Drienas P., Mahoney M.W.* Lectures on Randomized Numerical Linear Algebra // arXiv preprint arXiv:1712.08880v1 [cs.DS], 12/2017.
11. *Goreinov S.A., Tyrtyshnikov E.E., Zamarashkin N.L.* A theory of pseudoskeleton approximations // Linear Algebra and its Appl. 1996. V. 261. № 1–3. P. 1–21.
12. *Osinsky A.I., Zamarashkin N.L.* Pseudo-skeleton approximations with better accuracy estimates // Linear Algebra and its Appl. 2018. V. 537. P. 221–249.
13. *Osinsky A.I., Zamarashkin N.L.* New accuracy estimates for pseudoskeleton approximations of matrices // Doklady Math. 2016. V. 94. № 3. P. 643–645.
14. *Simon H.D., Zha H.* Low-rank matrix approximation using the Lanczos Bidiagonalization process with applications // SIAM J. Sci. Comput. 2000. V. 21. № 6. P. 2257–2274.
15. *Mikhalev A.Y., Oseledets I.V.* Rectangular submatrices of maximum volume and their computation // Doklady Math. 2015. V. 91. № 3. P. 267–268.
16. *Goreinov S.A., Oseledets I.V., Savostyanov D.V. et al.* How to find a good submatrix / Matrix Methods: Theory, Algorithms, Applications. Ed. by V. Olshevsky, E. Tyrtyshnikov. World Sci. Publ., 2010. P. 247–256.
17. *Osinsky A.I.* Rectangular maximum volume and projective volume search algorithms // arXiv 1809.02334 (Submitted on 7 Sep 2018)
18. *Boutsidis C., Drineas P., Magdon-Ismail M.* Near-optimal column-based matrix reconstruction // SIAM J. Comput. 2014. V. 43. № 2. P. 687–717.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 845–864

ОБЩИЕ ЧИСЛЕННЫЕ МЕТОДЫ

УДК 519.6

МОДЕЛИРОВАНИЕ СТРУКТУРЫ ДАННЫХ С ПОМОЩЬЮ БЛОЧНОГО ТЕНЗОРНОГО РАЗЛОЖЕНИЯ: РАЗЛОЖЕНИЕ ОБЪЕДИНЕННЫХ ТЕНЗОРОВ И ВАРИАЦИОННОЕ БЛОЧНОЕ ТЕНЗОРНОЕ РАЗЛОЖЕНИЕ КАК ПАРАМЕТРИЗОВАННАЯ МОДЕЛЬ СМЕСЕЙ[1)]

© 2021 г. И. В. Оселедец[1,2,*], П. В. Харюк[1,2,3,]**

[1] *121205 Москва, Большой бульвар, 30, стр. 1, Сколковский институт науки и технологий, Россия*

[2] *119333 Москва, ул. Губкина, 8, Институт вычислительной математики им. Г.И. Марчука РАН, Россия*

[3] *119991 Москва, Ленинские горы, 1, стр. 52, МГУ им. М.В. Ломоносова, Факультет вычислительной математики и кибернетики, Россия*

**e-mail: ivan.oseledets@gmail.com*

***e-mail: kharyuk.pavel@gmail.com*

Поступила в редакцию 24.12.2020 г.
Переработанный вариант 24.12.2020 г.
Принята к публикации 14.01.2021 г.

Развивается идея использования тензорных разложений в качестве параметрической модели группового анализа данных. Представлены две модели на основе блочного тензорного разложения: детерминистическая и вероятностная, с использованием различных форматов слагаемых. Установлена связь между блочным тензорным разложением и смесями непрерывных латентных вероятностных моделей: на основе блочного тензорного разложения построена модель смеси распределений со структурированным представлением. Модели были протестированы в задаче кластеризации набора цветных изображений и данных электрической активности мозга. Результаты показывают, что предложенные подходы способны к выделению релевантной индивидуальной составляющей данных. Библ. 54. Фиг. 4. Табл. 5.

Ключевые слова: групповой анализ данных, блочное тензорное разложение, машинное обучение, анализ компонент, модель смеси распределений.

DOI: 10.31857/S004446692105015X

1. ВВЕДЕНИЕ

В процессе поиска зависимостей в данных важно учитывать их природу. Ряд собранных наборов данных может быть представлен в тензорном виде (как многомерные массивы), и учет их размерности представляется важным не только для эффективной обработки, но и для потенциальной интерпретации найденных параметров в конкретном прикладном исследовании. Более того, подобные данные часто являются избыточными и могут быть приближены меньшим числом параметров. Тензорные разложения предоставляют возможность построения подобных приближений. Различные тензорные форматы, такие как канонический (полилинейный, CPD) или формат Таккера (см. [1], [2]), нашли успешное применение в задачах анализа данных (см. [3], [4]). В данной работе в качестве основы для моделей мы использовали так называемое *блочное тензорное разложение* (BTD), которое является одним из способов обобщения классических тензорных форматов и было предложено в [5]–[7].

В условиях ряда исследований требуется извлекать общую и индивидуальную информацию из данных. Например, подобные исследования возникают в когнитивных, медицинских и иных задачах изучения мозга по группе из нескольких человек (см., например, [8]–[10]); другим примером может служить химическая экспертиза набора образцов (см. [11]–[13]). Естественно ожидать от подобных наборов данных наличие общих (групповых) и индивидуальных частей, и реконструкцию соответствующих частей из данных мы будем называть *групповым анализом данных*.

[1)]Работа выполнена при финансовой поддержке РФФИ (проект № 16-31-00494-мол_а).

Предложенные в данной работе модели структуры данных основаны на BTD со слагаемыми различных типов, что позволяет моделировать групповые и индивидуальные части различными форматами.

Предложенные тензорные модели структуры данных основаны на концепте “связанного многомерного анализа компонент” (см. [14], [15]), реализованного в виде BTD с дополнительными условиями; кроме того, реализована модель смеси вероятностных распределений, параметризованная через BTD и названная *вариационным блочным тензорным разложением* (VBTD). В предыдущих работах по групповому анализу данных (см. [16], [17]) специальный случай первой (детерминистической) модели (а именно, условное BTD со слагаемыми в (Lr, 1) формате) был рассмотрен как составляющий элемент обработки данных для задачи классификации. В настоящей работе модели были протестированы в условиях задачи кластеризации. Сведения об использованных в работе обозначениях приведены в п. 10.1.

2. СТРУКТУРИРОВАНИЕ НАБОРОВ ДАННЫХ ДЛЯ ГРУППОВОГО АНАЛИЗА

Одна из главных целей группового анализа данных может быть сформулирована как выделение общей информации из комбинированного набора данных. Вопрос в том, что может быть названо “общей информацией”. В данной работе мы придерживаемся распространенной в области обработки сигналов точке зрения на природу данных: предположения о том, что наблюдения $\{X_i\}_{i=1}^N$ генерируются в результате преобразования активности нескольких источников, представленных в векторном виде. Адаптация такой точки зрения к групповому анализу данных ведет к разделению источников на общие, S_{com}, и индивидуальные, $\{S_i\}_{i=1}^N$.

Следующий шаг состоит в предположении об отделимости (в некотором смысле) общих и индивидуальных источников. Принято считать, что зависимость наблюдений от всех источников аддитивна.

Предположение группового анализа данных. *Наблюдения* $\{X_i\}_{i=1}^N$ могут быть приближены аддитивной функциональной зависимостью от общих, S_{com}, и индивидуальных, $\{S_i\}_{i=1}^N$, источников

$$X_i \approx F_i(S_{\text{com}}, S_i) = f_i(S_{\text{com}}) + g_i(S_i), \quad i = \overline{1, N}. \tag{1}$$

В случае, если наблюдения X_i двумерны (являются матрицами), распространенным способом моделирования указанной зависимости является линейное смешивание:

$$f_i(S_{\text{com}}) = S_{\text{com}} B_{f_i}^{\mathrm{T}}, \quad g_i(S_i) = S_i B_{g_i}^{\mathrm{T}}, \quad i = \overline{1, N}, \tag{2}$$

где B_{f_i}, B_{g_i} – матрицы коэффициентов общих и индивидуальных источников соответственно для образца X_i. Модель факторизации данных ставит вопрос о том, как найти требуемые параметры разложения. В [18] предложен ответ для общей части в виде группового метода независимых компонент (group ICA). Zhou и др. (см. [19]) построили алгоритм вычисления обеих частей разложения через приближение матрицы общих источников, ортогональной всем индивидуальным источникам. В [20] общая часть оценивается через малоранговое разложение склеенных матричных данных $X = [X_1 \ldots X_N] \approx S_{\text{com}}[B_{f_1} \ldots B_{f_N}]^{\mathrm{T}}$ и каждая индивидуальная часть вычисляется как малоранговое разложение данных после удаления групповой части: $X_i - S_{\text{com}} B_{f_i}^{\mathrm{T}} \approx S_i B_{g_i}^{\mathrm{T}}$.

В случае многомерных данных X_i размерами $n_1 \times \ldots \times n_d$, как правило, используются предварительно матризованные наблюдения. Однако в этом случае рассматриваемая модель игнорирует размерность входных данных. Детерминистическая BTD модель, предложенная в данной работе, обобщает подход для больших размерностей следующим образом. Рассмотрим матрицу развертки тензорных данных по выбранной моде источников $k \in \{1, 2, \ldots, d\}$ (например, отвечающей пикселям или времени) и применим к ней сформулированные ранее предположения (1), (2):

$$\text{unfold}_k(X_i) = (X_i)_{(k)} \approx S_{\text{com}} B_{f_i}^{\mathrm{T}} + S_i B_{g_i}^{\mathrm{T}}, \quad i = \overline{1, N}. \tag{3}$$

Далее учтем тензорную структуру слагаемых в правой части (3). Также вместо вычисления набора связанных разложений объединим данные $\{X_i\}_{i=1}^N$ вдоль новой групповой оси, получив общий тензор данных X размерами $n_1 \times \ldots \times n_d \times N$, и наложим условия на специальную структуру параметров, отвечающих групповой моде. Объединение данных требует выполнения условий однородности для $\{X_i\}_{i=1}^N$: все образцы должны иметь равную размерность с одинаковыми размерами мод.

3. ДЕТЕРМИНИСТИЧЕСКАЯ ГРУППОВАЯ МОДЕЛЬ НА ОСНОВЕ BTD РАЗЛОЖЕНИЯ ОБЪЕДИНЕННЫХ ДАННЫХ

После учета предположений из разд. 2 модельная структура входных данных примет следующий вид:

$$X \approx \sum_{i=1}^{N} \left(X_i(S_{\text{com}}, \theta_{\text{com},i}) + X_i(S_i, \theta_{\text{ind},i}) \right) \circ \mathbf{e}_i, \tag{4}$$

где $\mathbf{e}_i$ есть i-й столбец единичной матрицы I_N, $\circ$ обозначает внешнее произведение, $\theta_{\text{com},i}$, $\theta_{\text{ind},i}$ – общие и индивидуальные параметры, соответствующие тому или иному тензорному формату и отвечающие общим, S_{com}, и индивидуальным, S_i, источникам. Сосредоточимся на двух тензорных форматах: формате Таккера и (Lr, 1) (специальном каноническом с параметром P, отвечающим числу мод с полноразмерными фактор-матрицами), который был предложен в [21]. В случае (Lr, 1) формата для $T_i = X_i(S_i, \theta_{\text{ind},i})$ соответствующая сумма может быть переписана в следующем виде:

$$\sum_{i=1}^{N} T_i \circ \mathbf{e}_i = \sum_{i=1}^{N} \left[\!\left[C_1^{[i]}, \ldots, C_P^{[i]}, \mathbf{c}_{P+1}^{[i]} E^{[i]}, \ldots, \mathbf{c}_d^{[i]} E^{[i]} \right]\!\right] \circ \mathbf{e}_i = \left[\!\left[C_1, \ldots, C_d E, I_N E \right]\!\right], \tag{5}$$

где умножение фактор-матриц $\{C_l\}_{l=P+1}^{d}$ на $E = [E^{[1]}, \ldots, E^{[N]}]$ дублирует необходимое количество раз их столбцы. Выражение справедливо для обеих частей разложения: индивидуальных и общей. В последнем случае дополнительно предполагаем, что $X_i(S_{\text{com}}, \theta_{\text{com},i}) = p_i T_G$, с дополнительными параметрами $\sum_{i=1}^{N} p_i = 1$, $p_i \geq 0$, которые позволяют регулировать присутствие общей части T_G. Из этого предположения следует, что групповые части модели могут быть представлены как $T_G \circ \boxed{1}_{N\times 1}$, где $\boxed{1}_{N\times 1}$ – вектор из единиц размерами $N \times 1$. Сводя все предположения воедино, а также задав число компонент (ранги) для индивидуальных и общих слагаемых, $L = [L_1 \ldots L_N L_{N+1}]$, приходим к следующей модели:

$$X \approx \left[\!\left[C_1, \ldots, C_d, C_{d+1} \right]\!\right], \quad C_k = \begin{cases} [\overbrace{C_k^{[1]}}^{L_1}, \ldots, \overbrace{C_k^{[N+1]}}^{L_{N+1}}], & k = \overline{1, P}, \\ [\mathbf{c}_j^{[1]}, \ldots, \mathbf{c}_j^{[N+1]}] E, & k = \overline{P+1, d}, \quad j = k - P, \\ [I_N \mathbf{p}] E, & k = d+1, \end{cases} \tag{6}$$

где $E = \left[\boxed{1}_{1\times L_1} \otimes \mathbf{e}_1, \ldots, \boxed{1}_{1\times L_{N+1}} \otimes \mathbf{e}_{N+1} \right]$, $\otimes$ – произведение Кронекера, $\mathbf{p}$ – вектор параметров, определенный ранее. Однако вариативность групповой части в данной модели может быть слишком ограниченной. Для моделирования более сложной зависимости от общей части можно использовать формат Таккера со схожими условиями на фактор-матрицу групповой оси. Таким образом, полностью (Lr, 1) групповую модель (далее GLRO) дополним похожей моделью, в которой общая часть представлена в формате Таккера (далее GTLD):

$$\begin{gathered} X \approx \left[\!\left[G; A_1, \ldots, A_d, A_{d+1} \right]\!\right] + \left[\!\left[C_1, \ldots, C_d, C_{d+1} \right]\!\right], \\ A_{d+1} = \operatorname{diag}(\mathbf{p}), \quad C_k = \begin{cases} [\overbrace{C_k^{[1]}}^{L_1}, \ldots, \overbrace{C_k^{[N]}}^{L_N}], & k = \overline{1, P}, \\ [\mathbf{c}_j^{[1]}, \ldots, \mathbf{c}_j^{[N]}] E, & k = \overline{P+1, d}, \quad j = k - P, \\ I_N, & k = d+1, \end{cases} \end{gathered} \tag{7}$$

где последняя мода отвечает групповой оси, как и в (6).

В силу того что как канонический и (Lr, 1) форматы, так и формат Таккера одинаковым образом разделяют оси тензора, имеет смысл указать общий случай BTD со слагаемыми во всех трех форматах:

$$X \approx \underbrace{\sum_{m=1}^{M} \left[\!\left[G^{[m]}; A_1^{[m]}, \ldots, A_{d+1}^{[m]} \right]\!\right]}_{\text{в формате Таккера}} + \underbrace{\left[\!\left[B_1, \ldots, B_{d+1} \right]\!\right]}_{\text{канонический}} + \underbrace{\left[\!\left[C_1, \ldots, C_{d+1} \right]\!\right]}_{\text{(Lr,1)}}. \tag{8}$$

Отметим, что программная реализация предложенных моделей позволяет вычислять параметры такого разложения более общего вида. В групповом анализе данных также можно моделировать индивидуальные части в формате Таккера, но это потребует увеличения числа гиперпараметров (рангов) для каждого отдельного слагаемого. В случае же (Lr, 1) формата потребуется указать лишь один вектор гиперпараметров L, который задает ранги всех полноразмерных фактор-матриц, отвечающих индивидуальным слагаемым.

Помимо гиперпараметров, упомянутых ранее (P, L, и/или ранги Таккера), предложенный подход требует выбора некоторого подмножества среди первых P осей, $\Omega \subseteq \{1, 2, \ldots, P\}$, для которых вводится условие отделимости соответствующих общих и индивидуальных фактор-матриц (аналогично [19]) с тем, чтобы убрать пересечения между общей и индивидуальными активностями:

$$\begin{aligned} &(C_\gamma^{[N+1]})^{\mathrm{T}} C_\gamma^{[i]} = 0, \quad \text{для модели (6)}, \\ &(A_\gamma)^{\mathrm{T}} C_\gamma^{[i]} = 0, \quad \text{для модели (7)}, \quad \forall i = \overline{1, N}, \quad \forall \gamma \in \Omega. \end{aligned} \tag{9}$$

В данной работе мы остановились на выборе множества Ω, состоящего из одной моды.

4. ВАРИАЦИОННОЕ БЛОЧНОЕ ТЕНЗОРНОЕ РАЗЛОЖЕНИЕ КАК МОДЕЛЬ СМЕСИ РАСПРЕДЕЛЕНИЙ

Рассмотрим альтернативный подход к параметризации данных с помощью BTD. Первым отличием является вероятностный характер модели: векторизация наблюдаемых данных предполагается реализацией случайного вектора $\mathbf{x}$, зависящего от латентного (скрытого) представления; процесс моделирования состоит в детализации функции плотности распределения $p(\mathbf{x})$, а результат обучения потенциально применим также и для генерации новых данных (генеративная модель). Другое важное отличие состоит в том, что индивидуальные параметры модели специфичны для каждой отдельной группы наблюдений (кластера) в предположении, что известно точное число кластеров K, а не для каждого отдельного представителя.

4.1. Параметризованные смеси распределений

В общих чертах, предложенная вероятностная модель на основе BTD возникает в результате параметризации смеси непрерывных латентных моделей в предположении о структурированности матриц смешивания. Частным случаем подобных смесей является смесь вероятностных главных компонент (mixture of PPCAs, см. [22]). Базовая модель вероятностных главных компонент (PPCA) (см. [23]) представляет собой параметризацию наблюдений $\mathbf{x}$ с помощью скрытого (латентного) представления, имеющего стандартное нормальное распределение, $\mathbf{z} \sim N(\mathbf{0}, I)$: $p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$, $p(\mathbf{x}|\mathbf{z}) = N(\boldsymbol{\mu} + W\mathbf{z}, L^{-1})$, где $\boldsymbol{\mu}$ – постоянная составляющая среднего (сдвиг), W – матрица смешивания, $L^{-1} = \sigma^2 I$ – ковариационная матрица (в более общем случае факторного анализа L^{-1} не обязательно является скалярной). В модели смеси вероятностных непрерывных скрытых представлений предполагается, что $p(\mathbf{x})$ является мультимодальным распределением (см. [24]), порожденным вероятностной смесью нескольких распределений, каждая мода которого параметризована своим непрерывным скрытым представлением:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{x}), \quad p_k(\mathbf{x}) = \int p_k(\mathbf{x}|\mathbf{z}_k)p(\mathbf{z}_k)d\mathbf{z}_k, \quad 0 \le \pi_k \le 1, \quad \sum_{k=1}^{K} \pi_k = 1, \tag{10}$$

где $p_k(\mathbf{x}|\mathbf{z}_k) = N(\boldsymbol{\mu}_k + W_k\mathbf{z}_k, L_k^{-1})$, $p(\mathbf{z}_k) = N(\mathbf{0}, I)$. Если проводить аналогии с разд. 3, то в среднем наблюдения $\mathbf{x}$ предполагаются сгенерированными в результате взвешенной суммы нескольких

процессов линейного смешивания. Коэффициенты $\{\pi_k\}_{k=1}^K$ выступают в роли априорных вероятностей принадлежности к тому или иному кластеру. Соответствующие апостериорные вероятности вычисляются по правилу Байеса и имеют вид

$$\gamma_k(\mathbf{x}) = p(k|\mathbf{x}) = \frac{\pi_k p_k(\mathbf{x})}{\sum_l \pi_l p_l(\mathbf{x})}. \tag{11}$$

Обратим также внимание, что априорные вероятности принадлежности к тому или иному кластеру в построенной модели были параметризованы через исходные наблюдения и параметры кластера $\pi_k \equiv \pi_k(\mathbf{x}, W_k)$ (подробнее см. следующий п. 4.2.).

Дальнейшая связь с BTD устанавливается следующим образом: для каждого $k = \overline{1, K}$ положим $\mu_k = 0$, обозначим реализацию вектора $\mathbf{z_k}$ как $\mathbf{Z_k}$ и рассмотрим $W_k\mathbf{Z}_k$ как векторизацию некоторого тензора, структура которого зависит от выбора того или иного тензорного формата, например:

– канонический (полилинейный, CP):

$$\mathbf{Z}_k = \mathrm{vec}\left(\mathrm{diag}(\hat{\lambda}_k)\right) \in \mathbb{R}^{R_k \times 1}, \quad W_k = C_k^{(d)} \odot \ldots \odot C_k^{(1)}; \tag{12}$$

– (Lr, 1) с $1 < P < d$, $\mathbf{L}_k = (L_{k;1}, \ldots, L_{k;R_k})$, $M_k = \sum_i L_{k;i}$ (CP специального вида):

$$\mathbf{Z_k} \in \mathbb{R}^{M_k \times 1}, \quad W_k = \widetilde{C}_k^{(d)} \odot \ldots \odot \widetilde{C}_k^{(1)}, \quad \widetilde{C}_k^{(\mathrm{p})} = \begin{cases} C_k^{(p)} \in \mathbb{R}^{n_p \times M_k}, & p \le P, \\ \underbrace{C_k^{(p)} E_k}_{\text{повтор столбцов } C_k^{(p)} \in \mathbb{R}^{n_p \times R_k}}, & p > P; \end{cases} \tag{13}$$

– Таккера с ядром в роли латентного представления (“Tucker-core”):

$$\mathbf{Z}_k = \mathrm{vec}(\widehat{G}_k) \in \mathbb{R}^{r_1 \cdot \ldots \cdot r_d \times 1}, \quad W_k = A_k^{(d)} \otimes \ldots \otimes A_k^{(1)}; \tag{14}$$

– Таккера с фактор-матрицей в роли латентного представления (“Tucker-factor”), $G \in \mathbb{R}^{r_0 \times r_1 \times \ldots \times r_d}$:

$$\mathbf{Z}_k = \mathrm{vec}(\hat{\mathbf{a}}_k^{(0)}) \in \mathbb{R}^{r_0 \times 1}, \quad W_k = \left((A_k^{(d)} \otimes \ldots \otimes A_k^{(1)})((G_k)_{(0)})^{\mathrm{T}}\right) \otimes I; \tag{15}$$

– тензорного поезда (TT) [25] с ядрами $G_k^{(i)} \in \mathbb{R}^{r_{i-1} \cdot n_i \times r_i}$, $I_{N_{\{\ldots i\}}} = I_{n_{i+1} \ldots n_d}$:

$$\mathbf{Z}_k \in \mathbb{R}^{r_d \times 1}, \quad W_k = (I_{N_{\{\ldots 1\}}} \otimes G_k^{(1)}) \ldots (I_{N_{\{\ldots d-1\}}} \otimes G_k^{(d-1)}) G_k^{(d)}. \tag{16}$$

Таким образом, правдоподобие модели *вероятностного BTD-смешивания* принимает следующий вид:

$$p(\mathbf{x}|\mathbf{z}_1, \ldots, \mathbf{z}_K) = \sum_{k=1}^K \pi_k N(\mathbf{x}|W_k\mathbf{z}_k, L_k^{-1}), \quad W_k\mathbf{z}_k = \mathrm{vec}(T_k(\mathbf{z}_k)), \tag{17}$$

ковариационные матрицы L_k^{-1} полагаются либо имеющими скалярный вид (изотропный шум), либо диагональными. Обратная связь модели с классическим BTD разложением может быть прослежена через условное математическое ожидание $\mathbf{x}$ при известных реализациях скрытых представлений $\{\mathbf{z}_k\}_{k=1}^K$, результат соответствует развертке тензора в BTD формате с π_k в роли весов для слагаемых:

$$E[\mathbf{x}|\mathbf{z}_1, \ldots, \mathbf{z}_K] = \sum_{k=1}^K \pi_k W_k \mathbf{z}_k = \mathrm{vec}(\widehat{T}). \tag{18}$$

Помимо введенного правдоподобия, требуется также определить распределение для скрытых представлений $\{\mathbf{z}_k\}_{k=1}^K$. В силу того что правдоподобие моделируется с помощью нормального распределения, в качестве априорных распределений $p(\mathbf{z}_k)$ также выберем нормальное (как сопряженное к правдоподобию, см. [24]):

$$p(\mathbf{z}_k) = N(\mathbf{m}_k, \Lambda_k^{-1}) = N(\mathbf{m}_k(\mathbf{x}, W_k), \Lambda_k^{-1}). \tag{19}$$

Отличием от случая PPCA является использование диагональных матриц Λ_k^{-1} и ненулевых средних $\mathbf{m}_k$. Как и в случае вариационных автоэнкодеров (см. [26]), для средних $\mathbf{m}_k$ была введена зависимость от входных данных и параметров разложения, в данном случае $\mathbf{m}_k(\mathbf{x}, W_k) = (W_k^{\mathrm{T}} W_k)^{-1} W_k^{\mathrm{T}} \mathbf{x} = W_k^{+} \mathbf{x}$. Такой выбор объясняется следующими соображениями: рассмотрим условное среднее $E[\mathbf{x}|k] = W_k \mathbf{m}_k$, и для конкретной реализации $\mathbf{x}$ данное выражение есть ни что иное как проекция в пространство столбцов матрицы W_k, $W_k \mathbf{m}_k(\mathbf{x}, W_k) = P_{W_k} \mathbf{x}$, т.е. в итоге кластерное среднее определяется соответствующей проекцией наблюдения.

Параметры ковариационных матриц Λ_k^{-1} и L_k^{-1} полагались реализациями из лог-нормальных распределений с нулевым средним и единичной дисперсией. Наконец, укажем итоговое выражение для модели вариационного BTD разложения (VBTD) с нормальными априорным распределением и правдоподобием:

$$p(\mathbf{x}, \{\mathbf{z}_k, \Lambda_k, L_k^{-1}\}_{k=1}^K) = \sum_{k=1}^K p_k N(\ln L_k^{-1} | 0, I) N(\ln \Lambda_k^{-1} | 0, I),$$
$$p_k = \pi_k(\mathbf{x}, W_k) N(\mathbf{x} | W_k \mathbf{z}_k, L_k^{-1}) N(\mathbf{z}_k | \mathbf{m}_k(\mathbf{x}, W_k), \Lambda_k^{-1}), \quad W_k \mathbf{z}_k = \mathrm{vec}(T_k(\mathbf{z}_k)). \tag{20}$$

Текущая реализация предполагает, что параметры $\{W_k\}_{k=1}^K$ являются числовыми величинами, однако возможно построение более общей версии разложения, если моделировать их как случайные величины.

Аналогично детерминистической модели, введем групповую часть разложения, $\mathbf{z}_g$, а также введем обозначение $\hat{\mathbf{z}}_k = [\mathbf{z}_k \mathbf{z}_g]^{\mathrm{T}}$. Полагая независимость обеих частей скрытого представления $\mathbf{z}_g \perp \mathbf{z}_k$, получаем $p(\hat{\mathbf{z}}_k) = p(\mathbf{z}_k, \mathbf{z}_g) = p(\mathbf{z}_g) p(\mathbf{z}_k)$, и используя свойства нормального распределения, приходим к следующему виду правдоподобия:

$$p_k(\mathbf{x} | \hat{\mathbf{z}}_k) = N(\mathbf{x} | W_k \mathbf{z}_k + W_g \mathbf{z}_g, L_k^{-1} + L_g^{-1}), \quad k = \overline{1, K}, \tag{21}$$

где W_g – параметры разложения, отвечающие общему скрытому представлению $\mathbf{z}_g$, L_g^{-1} – ковариационная матрица для группового слагаемого.

4.2. Априорное решающее правило

Рассмотрим подробнее априорные вероятности принадлежности к кластерам, $\{\pi_k\}_{k=1}^K$. В общем виде они могут иметь сложную структуру, учитывающую дополнительные зависимости, и одной из наиболее важных является зависимость от входных данных $\mathbf{x}$, $\pi_k \equiv \pi_k(\mathbf{x}, \boldsymbol{\theta}_{\pi_k})$. От подобной функциональной зависимости требуется выполнение условия нормировки, т.е. $\pi_k(\mathbf{x}, \boldsymbol{\theta}_{\pi_k}) \geq 0$, $\sum_{k=1}^K \pi_k(\mathbf{x}, \boldsymbol{\theta}_{\pi_k}) = 1$. Распространенный способ учета такого условия состоит в использовании многопеременной логистической функции (softmax) на результате отображения некоторой (дифференцируемой) функции $a_k(\mathbf{x}, \boldsymbol{\theta}_{\pi_k})$ (см. [24]):

$$\pi_k \equiv \pi_k(\mathbf{x}, \boldsymbol{\theta}) = \mathrm{softmax}_k\left(a(\mathbf{x}, \boldsymbol{\theta}_\pi)\right) = \frac{\exp\left(a_k(\mathbf{x}, \boldsymbol{\theta}_{\pi_k})\right)}{\sum\limits_i \exp\left(a_l(\mathbf{x}, \boldsymbol{\theta}_{\pi_l})\right)}. \tag{22}$$

Смеси распределений с параметризованными априорными вероятностями известны как модели коллектива экспертов (см. [24]). В случае, если для нормализации используется функция $\mathrm{softmax}(a(\mathbf{x}, \boldsymbol{\theta}_\pi))$, отображение $a(\mathbf{x}, \boldsymbol{\theta}_\pi)$ должно быть задано таким образом, чтобы значения элементов вектора-результата были прямо пропорциональны близости наблюдения $\mathbf{x}$ к соответствующим кластерам. Выбор параметризации для предложенной модели на основе BTD был мотивирован результатами работы [19] и нашим опытом из [27], в которой аналогичное отображение использовалось для построения классификаторов на основе разложений Таккера, которые, в частности, позволили получить более устойчивые результаты при изменении оборудования сбора данных и метода химической экстракции в сравнении с рядом иных рассмотренных классификаторов.

Параметризация основана на главном угле (см. [28]) между пространствами источников (столбцы фактор-матрицы для соответствующей оси) и соответствующей матризацией наблюдения. Пусть имеется d-мерное тензорное наблюдение X_i, матрица развертки параметров $A_k^{(m)}$ при выбранной моде источников $m \in \{1, 2, \ldots, d\}$, $k = \overline{1, K}$, и их сингулярные разложения, $(X_i)_{(m)} = U_i \Sigma_i V_i^{\mathrm{T}}$ и $A_k^{(m)} = U_k \Sigma_k V_k^{\mathrm{T}}$, тогда отображение $a_k(\mathbf{x}, \boldsymbol{\theta}_{\pi_k})$ может быть вычислено следующим образом:

$$a_k(X_i, A_k^{(m)}) = \tilde{\mathbf{u}}^T \Sigma_k^{-1} V_k^{\mathrm{T}} (A_k^{(m)})^{\mathrm{T}} (X_i)_{(m)} V_i \Sigma_i^{-1} \tilde{\mathbf{v}} = \tilde{\mathbf{u}}^T U_k^{\mathrm{T}} U_i \tilde{\mathbf{v}} = \sigma_{\max}(U_k^{\mathrm{T}} U_i), \tag{23}$$

где $\tilde{\mathbf{u}}$, $\tilde{\mathbf{v}}$ – левый и правый сингулярные векторы для матрицы $U_k^{\mathrm{T}} U_i$, отвечающие ее максимальному сингулярному значению $\sigma_{\max}(\cdot)$.

5. ОБУЧЕНИЕ МОДЕЛЕЙ

5.1. Нелинейная задача наименьших квадратов

Для предложенных детерминистических моделей GLRO и GTLD вычисление параметров производится через условную оптимизацию нелинейных наименьших квадратов. Для заданного набора d-мерных тензоров $\{X_i\}_{i=1}^N$ одинаковых размеров, объединенных вдоль групповой оси в $(d+1)$-мерный тензор X, задача поиска параметров θ принимает следующий вид:

$$\min_{\boldsymbol{\theta}} \|X - T(\boldsymbol{\theta})\|_F^2 = \min_{\boldsymbol{\theta}_g, \{\boldsymbol{\theta}_i\}_{i=1}^N} \left\| X - \left(T(\boldsymbol{\theta}_g) + \sum_{i=1}^N T(\boldsymbol{\theta}_i) \right) \right\|_F^2 \quad \text{s.t.} \quad h(\boldsymbol{\theta}_g, \{\boldsymbol{\theta}_i\}_{i=1}^N) = 0, \tag{24}$$

где через $\boldsymbol{\theta}_g$, $\{\boldsymbol{\theta}_i\}_{i=1}^N$ обозначены параметры общих и индивидуальных частей соответственно (согласно (6) или (7)), и в выражении $h(\boldsymbol{\theta}_g, \{\boldsymbol{\theta}_i\}_{i=1}^N) = 0$ собраны условия на фактор-матрицы групповой моды и условия отделимости (9).

Среди существующих методов оптимизации параметров в задаче нелинейных наименьших квадратов был использован ряд методов первого и второго порядка (выражения для градиентов и матрицы Гессе приведены в п. 10.2., 10.3.), а также метод попеременных наименьших квадратов (ALS). Для вычисления параметров различных тензорных разложений существуют готовые Матлаб пакеты (например, Tensorlab, см. [29]), однако для проведения исследований из области машинного обучения более распространено использование языка Python, на котором и был реализован вычислительный код для настоящей работы.

В приложении моделей к реальным данным становится критичным учитывать вычислительные затраты по времени, поэтому мы ограничили число итераций до 10. Предварительные эксперименты на наборе данных ETH80 показали, что наиболее быстрым из рассмотренных методов является проекционный ALS, при этом в случае использования методов второго порядка не наблюдалось значимого прироста качества приближения. По этим причинам для последующих экспериментов был использован проекционный метод ALS.

5.2. Стохастический вариационный вывод (SVI)

Вероятностные модели способны ассимилировать новые наблюдения и делать предсказания с помощью статистического вывода. Для моделей со скрытым представлением $\mathbf{z}$ в общем случае для этого требуется вычисление апостериорного распределения $p(\mathbf{z}|\mathbf{x}) = p(\mathbf{x}, \mathbf{z})/p(\mathbf{x})$. Часто проблему представляет вычисление $p(\mathbf{x})$, и для статистического вывода используются различные приближения. Вариационный вывод представляет собой метод преобразования задачи вывода к задаче оптимизации. Основная идея заключается в замене точного апостериорного распределения на некоторого представителя из определенного семейства распределений, который как можно точнее приближает истинное апостериорное (подробнее о методе стохастического вариационного вывода см., например, [30]).

Непосредственная оптимизация параметров распределения производилась через максимизацию нижней грани правдоподобия (ELBO) (см. [31]). Для заданных моделью правдоподобия

$p(\mathbf{x}|\mathbf{z},\boldsymbol{\theta}_{\text{lh}})$ и априорного распределения $p(\mathbf{z}|\boldsymbol{\theta}_z)$, $p_\theta(\mathbf{x},\mathbf{z}) = p(\mathbf{x}|\mathbf{z},\boldsymbol{\theta}_{\text{lh}})p(\mathbf{z}|\boldsymbol{\theta}_z)$, а также для вариационного распределения $q(\mathbf{z}|\phi) = q_\phi(\mathbf{z})$ вариационная нижняя грань возникает в уравнении $\ln p_\theta(\mathbf{x}) = \text{ELBO} + \text{KL}\big(q_\phi(\mathbf{z})||\, p_\theta(\mathbf{z}|\mathbf{x})\big)$ и определяется как

$$\text{ELBO} \equiv E_{q_\phi(\mathbf{z})}[\ln p_\theta(\mathbf{x},\mathbf{z}) - \ln q_\phi(\mathbf{z})], \tag{25}$$

$\text{KL}(q_\phi(\mathbf{z})\|\, p_\theta(\mathbf{z}|\mathbf{x})) = \int q_\phi(\mathbf{z}) \ln \dfrac{q_\phi(\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})} d\mathbf{z}$ – дивергенция Кульбака–Лейблера, статистическая мера различий между двумя распределениями. Важно, что метод ELBO поддерживает обучение по подвыборкам и оценки Монте-Карло для стохастического градиента. Для реализации предложенной модели и оптимизации ее параметров методом стохастического вариационного вывода был использован пакет Pyro (см. [32]), в котором вычисления базируются на построении динамических вычислительных графов с помощью pytorch (см. [33]).

Для модели VBTD (20) было использовано следующее вариационное распределение:

$$\begin{gathered} p(\{\mathbf{z}_k, \Lambda_k, \sigma_k^2\}_{k=1}^K) = \prod_{k=1}^K \gamma_k^{z_{\pi_k}} \cdot \prod_{k=1}^K N(\mathbf{z}_k|\hat{\mathbf{m}}_k, \hat{\Lambda}_k^{-1}), \\ \gamma_k = \frac{p_k}{\sum\limits_l p_l}, \quad z_{\pi_k} \in \{0,1\}, \quad \sum_{k=1}^K z_{\pi_k} = 1, \quad p(z_{\pi_k} = 1) = \pi_k, \quad p(\mathbf{z}_\pi) = \prod_{k=1}^K \pi_k^{z_{\pi_k}}, \end{gathered} \tag{26}$$

где $N(\mathbf{z}_k|\widehat{\mathbf{m}}_k, \widehat{\Lambda}_k^{-1})$ – истинные апостериорные распределения для каждой отдельной модели непрерывного смешивания на основе нормального распределения.

6. КРАТКИЙ ОБЗОР ИНЫХ МОДЕЛЕЙ

В [34] представлена практическая методология использования специальных матричных разложений для группового анализа данных. Calhoun и др. строят групповой анализ данных функциональной магнитно-резонансной томографии (фМРТ) на основе метода независимых компонент (ICA). В деталях обсуждаются проблемы выбора числа компонент, моделирования зависимости данных от общих компонент, а также статистической валидации результатов. Метод ориентирован на выделение только общей активности, включает в себя двухшаговое выделение главных компонент (на уровне отдельных субъектов и на групповом уровне, для объединения найденных индивидуальных главных компонент) с последующим выделением независимых компонент. Вкратце модель выглядит следующим образом:

$$\underset{\text{(PCA, инд.)}}{X_i \approx Y_i V_i^{\mathrm{T}}}, \quad \underset{\text{(PCA, общ.)}}{Z \approx [Y_1, \ldots, Y_N]W}, \quad \underset{\text{(ICA)}}{Z \approx SA^{\mathrm{T}}}, \quad X_i \approx SA^{\mathrm{T}} W_i^{\mathrm{T}} V_i^{\mathrm{T}} = S(V_i W_i A)^{\mathrm{T}} = SM_i^{\mathrm{T}}, \tag{27}$$

где столбцы матрицы S – источники, M_i – матрица смешивания для наблюдения X_i, $i = \overline{1, N}$.

Несколькими годами позже появился обзор [18] прикладных исследований, использующих групповой метод независимых компонент (GICA), в частности, для анализа фМРТ данных, вызванных потенциалов, данных точечного нуклеотидного полиморфизма. В частности, в обзоре перечислены исследования, в которых метод независимых компонент адаптирован к задаче группового анализа данных; подходы сгруппированы в пять категорий: (1) построение отдельных факторизаций для каждого отдельного субъекта с последующим поиском взаимосвязей между ними; (2)–(3) объединение данных по одной из осей (пространственный/временной ICA); (4) усреднение данных по групповой оси; (5) тензорные разложения. Кроме того, авторы демонстрируют возможность использования ICA для мультимодального (т.е. включающего разнородные данные) анализа через модели общего (joint) и параллельного (parallel) ICA.

Другая известная модель, тензорный вероятностный метод независимых компонент (TPICA), основана на совмещении канонического разложения и вероятностного ICA (см. [10]). В данном подходе нет строгого разделения параметров на общие и индивидуальные, а матрица смешивания имеет структуру произведения Хатри–Рао между фактор-матрицами групповой оси и оси времени.

Позднее данные методы были рассмотрены с более общей точки зрения Guo, Pagnoni (см. [35]). В работе предложен модифицированный EM-алгоритм для максимизации правдоподобия с учетом трех вариантов структуры матрицы смешивания:

$$M_{\text{TPICA}} = M^{\text{m}_{\text{subj}}} \otimes M^{\text{m}_{\text{time}}}, \quad M_{\text{GICA}} = [M_1 \ \dots \ M_N], \quad M_{\text{gTPICA}} = \begin{bmatrix} M_1^{\text{m}_{\text{subj}}} \otimes M_1^{\text{m}_{\text{time}}} \\ \vdots \\ M_N^{\text{m}_{\text{subj}}} \otimes M_N^{\text{m}_{\text{time}}} \end{bmatrix}. \tag{28}$$

Дополнительно авторами рассмотрена методология выбора оптимальной структуры матрицы смешивания на основе критерия отношения правдоподобия, валидация которой была произведена на фМРТ данных.

Работа [36] посвящена построению группового анализа данных электрической активности мозга (ЭЭГ) на основе неотрицательных матричных разложений с учетом как групповой, так и индивидуальных составляющих: разделение индивидуальных фактор-матриц $\{S_i^{(I)}\}_{i=1}^N$, как и увеличение степени схожести между групповыми фактор-матрицами $\{S_i^{(C)}\}_{i=1}^N$, достигалось с помощью введения соответствующих слагаемых в оптимизируемый функционал:

$$\min_{S_i^{(\text{C})}, S_i^{(\text{I})}, M_i|_{i=1}^N} \sum_{i=1}^{N} \left[\left\| X_i - [S_i^{(\text{C})}, S_i^{(\text{I})}] M_i^{\text{T}} \right\|_F^2 + \gamma \left\| [S_i^{(\text{C})}, S_i^{(\text{I})}] \right\|_F^2 + \right.$$
$$\left. + \sum_{j=1}^{i-1} \left(\alpha \left\| S_i^{(\text{C})} - S_j^{(\text{C})} \right\|_F^2 - \beta \left\| S_i^{(\text{I})} - S_j^{(\text{I})} \right\|_F^2 \right) \right], \quad \text{s.t.} \quad S_i^{(\text{C})}, S_i^{(\text{I})}, M_i|_{i=1}^N \geq 0. \tag{29}$$

В [37] представлен вариант модели группового анализа данных на основе неотрицательного разреженного канонического разложения и концепта “связанных тензорных разложений”, в которой с помощью иерархического ALS алгоритма (вариант блочно-координатного спуска, каждая итерация которого состоит в последовательном обновлении строк/столбцов фактор-матриц, см. [38], [39]) оцениваются статистически независимые индивидуальные компоненты. Оптимизационная задача для вычисления разложения имеет вид (условия неотрицательности и разреженности опущены):

$$\min \sum_{i=1}^{N} \left\| X_i - \left[\left[\Lambda; U_1^{(i)}, \dots, U_d^{(i)} \right] \right] \right\|_F^2$$
$$\text{s.t.} \quad U_k^{(i)}\Big|_{i=1}^N = U_k, \quad \|U_k[:, r]\|_F = 1, \quad r = \overline{1, R}, \quad k = \overline{1, K}, \quad K < d, \quad \Lambda = \text{Diag}(\lambda_1, \dots, \lambda_R). \tag{30}$$

В [40] была предложена модель на основе разложения Таккера с двумя фактор-матрицами для случая 3D данных. Общие компоненты моделируются как первые C столбцов левой сингулярной матрицы соответствующей фактор-матрицы, а индивидуальные компоненты вычисляются как первые I_k столбцов левой сингулярной матрицы для проекции той же фактор-матрицы на пространство, ортогональное пространству общих компонент.

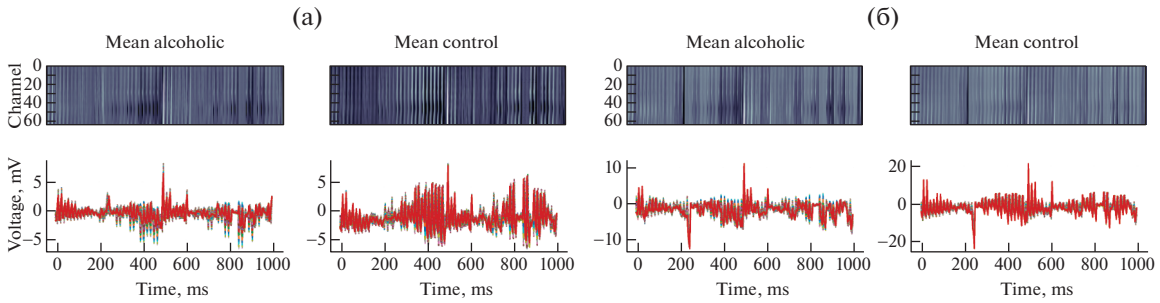
В [19] был предложен метод выделения общей и индивидуальной активностей на основе матричной модели (COBE). Предполагаемая структура данных схожа с рассматриваемой в [36], но модель выделяет общие источники иным образом:

$$\min_{\hat{A}, A_i} \sum_{i=1}^{N} \left\| X_i - \hat{A} \hat{B}_i^T - A_i B_i^{\text{T}} \right\|_F^2$$
$$\text{s.t.} \quad \hat{A}^{\text{T}} \hat{A} = I_C, \quad A_i^{\text{T}} A_i = I_{n_2 - C}, \quad \hat{A}^{\text{T}} A_i = 0, \quad i = \overline{1, N}, \tag{31}$$

где $\hat{A}$ – общие источники (C столбцов), A_i – индивидуальные источники.

Из существующих расширений BTD выделим вариант разложения со слагаемыми в формате Таккера, предложенный в [41] для случая совмещенных фактор-матриц. Предложенные вычислительные алгоритмы опираются на расширенный QZ алгоритм и метод Якоби. Кроме того, важно отметить, что идея использования BTD разложения с разным типом слагаемых не является новой, например, в [42] были использованы одновременно слагаемые в формате Таккера и каноническом формате для построения приближений классических потенциалов.

Фиг. 1. ETH80, изображения разных классов (а) и выборочные ракурсы 10 объектов класса “чашка” (б). Набор данных доступен по адресу: http://datasets.d2.mpi-inf.mpg.de/eth80/eth80-contours.tgz



Фиг. 2. Усредненные данные ЭЭГ для первого (а) и третьего (б) экспериментов; на изображениях сверху сигналы изображены в 2D виде, где более светлый цвет соответствует большим значениям магнитуды; разные цвета на изображениях снизу отвечают разным каналам. Набор данных доступен по ссылке: https://archive.ics.uci.edu/ml/datasets/eeg+database

7. ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

7.1. Программная реализация

Все модели и вычислительные эксперименты были реализованы на языке программирования Python с использованием дистрибутива Anaconda (см. [43]), который включает в себя различные пакеты для научных вычислений. В данном исследовании были использованы следующие пакеты: numpy [44], scipy [45], pandas [46], scikit-learn [47], matplotlib [48], seaborn [49], pyro [32]. Весь код опубликован в следующих репозиториях: https://github.com/kharyuk/gbtd; https://github.com/kharyuk/vbtd. Эксперименты были систематизированы с помощью Jupyter Notebooks (см. [50]).

7.2. Наборы данных

ETH-80. ETH80 (см. [51]) состоит из изображений различных объектов, снятых с разных ракурсов. Всего доступно восемь классов: “яблоко”, “машина”, “корова”, “чашка”, “собака”, “лошадь”, “груша”, “томат”; каждый класс включает 10 объектов (наблюдений) (фиг. 1). Полученный в результате набор данных имеет размеры $(N_{\text{samples}}, N_{\text{pixels}}, N_{\text{angles}}, N_{\text{colors}})$, где $N_{\text{angles}} = 41$. Разрешение изображений было понижено до 32×32, откуда $N_{\text{pixels}} = 1024$.

SMNI EEG (ERP). Набор данных SMNI EEG состоит из измерений электрической активности мозга по $N_{\text{electrode}} = 64$ электродам для двух групп субъектов: страдающих алкогольной зависимостью и контрольной группы. Согласно описанию, данные были записаны с частотой 256 Гц и нарезаны на фрагменты длиной в 1 с, содержащие момент предъявления стимула. Данные были собраны для $N_{\text{condition}} = 3$ условий: предъявление одиночного стимула (изображения), двух совпадающих и двух различных стимулов. Данные были дополнительно усреднены по повторностям, помеченные как испорченные образцы были исключены, и итоговый набор данных был представлен как тензор размерами $(N_{\text{subject}}, N_{\text{time}}, N_{\text{electrode}}, N_{\text{condition}})$, где $N_{\text{subject}} = 119$, из них 76 субъектов из первой группы, 43 из контрольной (фиг. 2).

Таблица 1. Сравнительные результаты моделей с подобранными гиперпараметрами для набора данных ETH80

		ARI	AMI	FMI
HAC	Raw[1]	.568	.749	.638
	COBE[1]	.643 ± .023	.774 ± .016	.691 ± .019
	COBE[2]	.573 ± .013	.780 ± .005	.663 ± .009
	GICA[1]	.722 ± .000	.794 ± .000	.753 ± .000
	GICA[2]	.577 ± .000	.802 ± .000	.661 ± .000
	GLRO[1]	.472 ± .093	.665 ± .060	.559 ± .074
	GLRO[2]	.471 ± .098	.693 ± .076	.580 ± .067
	GTLD[1]	.497 ± .071	.668 ± .047	.573 ± .054
	GTLD[2]	.493 ± .072	.699 ± .043	.585 ± .047
Kmeans	Raw	.533 ± .068	.690 ± .047	.597 ± .055
	COBE[3]	.571 ± .049	.729 ± .030	.635 ± .037
	GICA[3]	.559 ± .057	.712 ± .042	.624 ± .046
	GLRO[2]	.461 ± .077	.645 ± .062	.543 ± .062
	GLRO[3]	.445 ± .065	.645 ± .047	.535 ± .050
	GTLD[3]	.473 ± .065	.652 ± .050	.550 ± .052
GMM	Raw	.475 ± .072	.639 ± .064	.549 ± .060
	COBE[3]	.333 ± .262	.438 ± .341	.376 ± .293
	GICA[4]	.384 ± .229	.513 ± .301	.439 ± .258
	GLRO[2]	.473 ± .079	.651 ± .065	.553 ± .063
	GLRO[3]	.466 ± .072	.662 ± .054	.554 ± .054
	GTLD[3]	.476 ± .073	.656 ± .056	.555 ± .058

Примечание. Формат ячеек таблицы: "$x \pm y$", где x – среднее значение, y – стандартное отклонение. Для HAC за R_{average} и R_{complete} обозначены пересчет расстояний через усреднение и взятие максимума, за ρ_{corr}, $\rho_{\cos}$, $\rho_{\text{Canb.}}$ – корреляция, косинусное расстояние и расстояние Канберра; Raw[1]: ρ_{corr}, R_{average}; COBE[1]: $r_c = 2$, $\rho_{\cos}/\rho_{\text{corr}}$, R_{complete}; COBE[2]: $r_c = 2$, $\rho_{\cos}/\rho_{\text{corr}}$, R_{average}; COBE[3]: $r_c = 4$; GICA[1]: $r_c = 2$, $r_i = \overline{3,5}$, $\rho_{\cos}/\rho_{\text{corr}}$, R_{complete}; GICA[2]: $r_c = 2$, $r_i = \overline{4,5}$, $\rho_{\text{Canb.}}$, R_{average}; GICA[3]: $r_c = 4$, $r_i = 3$; GICA[4]: $r_c = 2$, $r_i = 2$; GLRO[1]: $r_c = 4$, $r_i = 4$, ρ_{corr}, R_{complete}; GLRO[2]: $r_c = 4$, $r_i = 3$, для HAC – $\rho_{\text{Canb.}}$, R_{average}; GLRO[3]: $r_c = 5$, $r_i = 2$; GTLD[1]: $r_c = 5$, $r_i = 3$, ρ_{corr}, R_{complete}; GTLD[2]: $r_c = 4$, $r_i = 4$, $\rho_{\text{Canb.}}$, R_{average}; GTLD[3]: $r_c = 5$, $r_i = 4$.

7.3. Оценка качества методов кластеризации

Кластеризация представляет собой одну из стандартных задач машинного обучения, в которой требуется сгруппировать различные образцы по заданному критерию близости. Более формально, указывается некоторый функционал близости $\rho(X_i, X_j)$, и для входных данных $\{X_i\}_{i=1}^N$ необходимо ввести разбиение на непересекающиеся подмножества (кластеры) таким образом, что объекты внутри одного кластера были более близки друг к другу, чем к образцам из прочих кластеров, в смысле $\rho(X_i, X_j)$.

Если для данных известны истинные метки, качество кластеризации можно измерить следующим образом. Рассмотрим два разбиения, $U = \{U_i\}$, $V = \{V_j\}$, где U отвечает истинным меткам, и введем на них четыре стандартные величины: число истинно-положительных (TP), истинно-отрицательных (TN), ложноотрицательных (FN), ложноположительных (FP) пар:

$$\begin{aligned} &\mathrm{TP} = \left|\{(a_k, a_l) \,|\, a_k, a_l \in U_i, a_k, a_l \in V_j\}\right|, \quad \mathrm{TN} = \left|\{(a_k, a_l) \,|\, a_k \in U_{i_1}, a_l \in U_{i_2}, a_k \in V_{j_1}, a_l \in V_{j_2}\}\right|, \\ &\mathrm{FP} = \left|\{(a_k, a_l) \,|\, a_k \in U_{i_1}, a_l \in U_{i_2}, a_k, a_l \in V_j\}\right|, \quad \mathrm{FN} = \left|\{(a_k, a_l) \,|\, a_k, a_l \in U_i, a_k \in V_{j_1}, a_l \in V_{j_2}\}\right|. \end{aligned} \tag{32}$$

Также определим следующие вспомогательные величины:

$$P_{ij} = \left|U_i \cap V_j\right|/N, \quad P_i = \left|U_i\right|/N, \quad P_j = \left|V_j\right|/N, \quad H(W) = E[-\log P(W)], \tag{33}$$

где W – некоторое разбиение, $H(\cdot)$ – энтропия.

Таблица 2. Сравнительные результаты моделей с подобранными гиперпараметрами для набора данных SMNI EEG

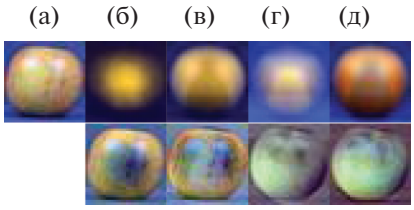
		ARI	AMI	FMI
HAC	Raw[1]	.161	.104	.601
	COBE[1]	.092 ± .060	.057 ± .044	.577 ± .029
	COBE[2]	.077 ± .053	.057 ± .032	.591 ± .047
	GICA[1]	.253 ± .000	.198 ± .000	.715 ± .000
	GLRO[1]	.048 ± .075	.036 ± .044	.621 ± .059
	GTLD[1]	.060 ± .078	.046 ± .037	.597 ± .052
	GTLD[2]	.009 ± .047	.058 ± .060	.568 ± .038
Kmeans	Raw	.196 ± .015	.123 ± .009	.624 ± .009
	COBE[2]	.168 ± .059	.106 ± .037	.608 ± .033
	GICA[2]	.236 ± .004	.153 ± .005	.646 ± .006
	GLRO[2]	.039 ± .054	.027 ± .037	.676 ± .080
	GTLD[2]	.126 ± .094	.083 ± .063	.674 ± .051
GMM	Raw	.156 ± .049	.103 ± .032	.597 ± .027
	COBE[2]	.151 ± .060	.101 ± .038	.597 ± .031
	GICA[2]	.196 ± .092	.127 ± .059	.630 ± .147
	GLRO[3]	.037 ± .059	.018 ± .044	.709 ± .037
	GLRO[4]	.033 ± .021	.021 ± .021	.720 ± .006
	GTLD[2]	.052 ± .056	.038 ± .049	.712 ± .030
	GTLD[3]	.058 ± .060	.034 ± .041	.704 ± .032

Примечание. Формат ячеек таблицы: "$x \pm y$", где x – среднее значение, y – стандартное отклонение. Для HAC за R_{complete} обозначен пересчет расстояний через взятие максимума, за ρ_{corr}, $\rho_{\cos}$, ρ_{RBF}, ρ_{l_2}, $\rho_{\text{Canb.}}$ – корреляция, косинусное, RBF, l_2 расстояния и расстояние Канберра; Raw[1]: для HAC – $\rho_{l_2}/\rho_{\cos}/\rho_{\text{RBF}}/\rho_{\text{corr}}$, R_{complete}; COBE[1]: $r_c = 3$, $\rho_{l_2}/\rho_{\cos}/\rho_{\text{RBF}}/\rho_{\text{corr}}$, R_{complete}; COBE[2]: $r_c = 4$, для HAC – $\rho_{l_2}/\rho_{\cos}/\rho_{\text{RBF}}/\rho_{\text{corr}}$, R_{complete}; GICA[1]: $r_c = 2$, $r_i = 1$, ρ_{l_2}, R_{complete}; GICA[2]: $r_c = 1$, $r_i = 2$; GLRO[1]: $r_c = 4$, $r_i = 5$, $\rho_{\text{Canb.}}$, R_{complete}; GLRO[2]: $r_c = 3$, $r_i = 5$; GLRO[3]: $r_c = 3$, $r_i = 4$; GLRO[4]: $r_c = 1$, $r_i = 3$; GTLD[1]: $r_c = 1$, $r_i = 3$, ρ_{corr}, R_{complete}; GTLD[2]: $r_c = 4$, $r_i = 3$, для HAC – $\rho_{\text{Canb.}}$, R_{complete}; GTLD[3]: $r_c = 2$, $r_i = 1$.

Используя величины (32), (33), определим три стандартных показателя качества, индекс Фолькс–Мэллоуз (FMI), как геометрическое среднее между точностью и полнотой, скорректированный индекс Рэнда (ARI), который схож с долей верных ответов, и скорректированная взаимная информация (AMI), $C_N^2 = \binom{N}{2}$:

$$\begin{gathered}\text{RI} = (TP + TN)/C_N^2, \quad \text{MI} = \sum_{i=1}^{I}\sum_{j=1}^{J} P_{ij}\log\frac{P_{ij}}{P_i P_j},\\ \text{ARI} = \frac{\text{RI} - E[\text{RI}]}{\max(\text{RI}) - E[\text{RI}]}, \quad \text{AMI} = \frac{\text{MI} - E[\text{MI}]}{\max(H(U), H(V)) - E[\text{MI}]},\\ \text{FMI} = \sqrt{\frac{TP}{TP + FP}\frac{TP}{TP + FN}}.\end{gathered} \tag{34}$$

В силу того что модели GLRO и GTLD были построены как расширение модели, предложенной в [19] (будем ссылаться на нее по названию алгоритма COBE), модель COBE была также использована для сравнения. Другой альтернативой для сравнения качества была выбрана модель группового метода независимых компонент GICA (см. [18]). Оригинальная версия модели была разработана для выделения только общей активности; в наших экспериментах была воспроизведена процедура выделения общих независимых компонент, дополненная вычитанием восстановленных общих частей из исходных данных (т.е. контрастированием). Выделение индивиду-



Фиг. 3. Пример изображения из набора ETH80 и соответствующие ему индивидуальная и групповая части: (а) оригинал; (б) COBE, $r_c = 2$; (в) GICA, $r_c = 2$, $r_i = 4$; (г) GLRO, $r_c = 4$, $r_i = 3$; (д) GTLD, $r_c = 5$, $r_i = 3$. (б)–(д) – Верхний ряд изображений отвечает групповым частям, нижний – индивидуальным.

альных частей в случае COBE было проведено способом, описанным в оригинальной статье, для GLRO и GTLD были использованы соответствующие слагаемые полного разложения. Для моделей были использованы разные сочетания общих рангов, $r_C = \overline{1,5}$, и рангов индивидуальных, $r_I = \overline{1,5}$ (кроме COBE, где используются только r_C). Для GICA r_C соответствовал одновременно числу вторичных главных компонент и числу независимых компонент. В случае GTLD все ранги Таккера полагались равными r_C.

Контрастированные и необработанные данные были кластеризованы с помощью следующих алгоритмов: иерархическая агломеративная кластеризация HAC (см. [52]), K-средних (см. [52]) и кластеризации на основе модели смеси гауссовских распределений GMM (см. [52]) (с диагональными ковариационными матрицами). В табл. 1, 2 представлены результаты, полученные для ETH80 и SMNI EEG соответственно. Результаты для всех методов контрастирования были усреднены по 10 запускам; для алгоритмов K-средних/GMM результаты дополнительно усреднены по 20 разным инициализациям для каждого запуска.

На фиг. 3 приведены примеры общих и индивидуальных частей, полученных разными способами для одного из изображений из данных ETH80.

Кластеризация с помощью предложенной VBTD модели может быть выполнена через использование как априорного правила $\pi_k(\mathbf{x}, \theta_{\pi_k})$, так и через вычисление апостериорных вероятностей γ_k. На наборах данных были протестированы различные конфигурации VBTD моделей с фиксированными рангами (равными 3). Индивидуальные слагаемые моделировались в (Lr, 1) формате. Результаты для лучших пяти конфигураций на апостериорном правиле приведены в табл. 3 в смысле AMI и в табл. 4 в смысле ARI. Все эксперименты проведены с ограничением на число итераций $N_{\text{it}} = 100$ при фиксированном случайном состоянии.

Таблица 3. VBTD, топ-5 конфигураций в смысле апостериорного AMI

Данные	Формат слагаемых		Шум, L_k^{-1}	Априорное правило			Апостериорное правило		
	(инд.)	(груп.)		AMI	ARI	FMI	AMI	ARI	FMI
ETH80	(Lr, 1)	–	изотр.	.646 (.619)	.335 (.320)	.512 (.498)	.683 (.640)	.412 (.404)	.560 (.538)
	(Lr, 1)	TT	диаг.	.585 (.529)	.383 (.272)	.522 (.486)	.650 (.340)	.392 (.116)	.544 (.378)
	(Lr, 1)	Таккер (ядро)	изотр.	.709 (.458)	.508 (.257)	.616 (.455)	.646 (.420)	.454 (.175)	.567 (.405)
	(Lr, 1)	TT	изотр.	.427 (.382)	.169 (.139)	.412 (.389)	.623 (.496)	.395 (.298)	.527 (.429)
	(Lr, 1)	–	диаг.	.541 (.507)	.255 (.235)	.484 (.467)	.600 (.541)	.336 (.255)	.531 (.484)
SMNI	Таккер (ядро)	–	изотр.	.032 (–.005)	.081 (.001)	.641 (.528)	.166 (.166)	.215 (.215)	.708 (.708)
	Таккер (фактор)	–	изотр.	.047 (.013)	.094 (.039)	.705 (.697)	.139 (.000)	.060 (.000)	.731 (.731)
	Таккер (ядро)	–	диаг.	.026 (–.010)	.053 (.010)	.712 (.710)	.128 (.000)	.186 (.000)	.731 (.731)
	TT	–	диаг.	.017 (.000)	.046 (.026)	.644 (.644)	.104 (.091)	.161 (.147)	.601 (.596)
	(Lr, 1)	–	изотр.	.044 (.001)	.098 (.020)	.669 (.560)	.090 (.006)	.162 (.004)	.660 (.585)

Примечание. Формат ячеек таблицы: "$x(y)$", где x – максимальное значение индекса, y – после 100 итераций.

Таблица 4. VBTD, топ-5 конфигураций в смысле апостериорного ARI

Данные	Формат слагаемых		Шум, L_k^{-1}	Априорное правило			Апостериорное правило		
	(инд.)	(груп.)		AMI	ARI	FMI	AMI	ARI	FMI
ETH80	(Lr, 1)	Таккер (ядро)	изотр.	.709 (.458)	.508 (.257)	.616 (.455)	.646 (.420)	.454 (.175)	.567 (.405)
	TT	–	диаг.	.310 (.238)	.188 (.091)	.394 (.344)	.555 (.463)	.433 (.271)	.501 (.410)
	(Lr, 1)	–	изотр.	.646 (.619)	.335 (.320)	.512 (.498)	.683 (.640)	.412 (.404)	.560 (.538)
	(Lr, 1)	TT	изотр.	.427 (.382)	.169 (.139)	.412 (.389)	.623 (.496)	.395 (.298)	.527 (.429)
	(Lr, 1)	TT	диаг.	.585 (.529)	.383 (.272)	.522 (.486)	.650 (.340)	.392 (.116)	.544 (.378)
SMNI	Таккер (ядро)	–	изотр.	.032 (–.005)	.081 (.001)	.641 (.528)	.166 (.166)	.215 (.215)	.708 (.708)
	Таккер (ядро)	–	диаг.	.026 (–.010)	.053 (.010)	.712 (.710)	.128 (.000)	.186 (.000)	.731 (.731)
	(Lr, 1)	–	изотр.	.044 (.001)	.098 (.020)	.669 (.560)	.090 (.006)	.162 (.004)	.660 (.585)
	TT	–	диаг.	.017 (.000)	.046 (.026)	.644 (.644)	.104 (.091)	.161 (.147)	.601 (.596)
	(Lr, 1)	(Lr, 1)	диаг.	.035 (.000)	.074 (–.021)	.720 (.689)	.071 (.057)	.136 (.117)	.731 (.640)

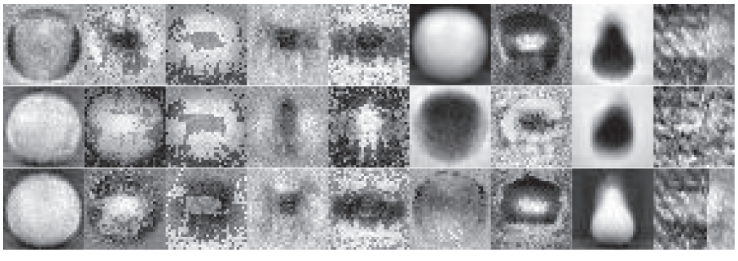
Примечание. Формат ячеек таблицы: "$x(y)$", где x – максимальное значение индекса, y – после 100 итераций.

В отличие от детерминистических моделей, в которых каждый отдельный образец раскладывался на общие и индивидуальные составляющие, VBTD модель выделяет индивидуальные части для отдельного кластера (своеобразные кластерные "тренды"). Эта особенность позволяет визуализировать выделенные источники для каждого отдельного кластера. На фиг. 4 изображены примеры для набора данных ETH80.

8. ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Если ориентироваться на индекс FMI, все алгоритмы показывают достаточно оптимистичные результаты. Однако индекс не является устойчивым к предсказаниям с константными значениями, что делает его особенно ненадежным в случае несбалансированных наборов данных с малым числом кластеров (SMNI EEG). В то же время показатели AMI и ARI отражают качество кластеризации более надежным образом. Следует отметить особенность набора данных SMNI EEG: в целом, все использованные алгоритмы показали невысокое качество кластеризации; предполагаем, что результаты можно улучшить с помощью дополнительной предварительной обработки данных, которая выделит более тонкие различия между группами субъектов, но этот вопрос оставлен за рамками работы.

GLRO и GTLD модели показали несколько худшие результаты в сравнении с другими рассмотренными методами, что может быть объяснено особенной структурой контрастированных данных. В случае GICA или COBE предположение о малоранговости применялось только к об-



Фиг. 4. Примеры источников, построенных с помощью VBTD. Конфигурация модели: (Lr, 1) формат индивидуальных слагаемых (ранг 3, $P = 2$), TT формат для общей части ($r = (1, 3, 3, 3)$), диагональная ковариационная матрица шума. Столбцы отвечают кластерам (последний из них – общей части), строки – отдельным восстановленным источникам (без упорядочивания).

щей части разложения, которая затем вычиталась из данных. В моделях GLRO и GTLD малоранговое представление строится также и для индивидуальных частей. Предварительно настроенные VBTD модели показали качество кластеризации, сравнимое с GICA и COBE. Вместе с тем в текущей реализации модели мы столкнулись с нестабильностями процесса обучения, которые в ряде случаев проявляют себя заметным зазором между лучшими значениями показателей и значениями на последней итерации. Хотя рассмотренные матричные модели имеют высокие показатели качества, данные модели опираются на избыточное представление данных. В отличие от них, предложенная VBTD модель не требует вычисления параметров для каждого отдельного образца, что делает ее предпочтительнее с точки зрения затрат на хранение параметров. Кроме того, в модели может быть учтено масштабирование по числу размерностей (например, с помощью формата квантизированного тензорного поезда QTT, см. [53]). Используемый метод для вычисления параметров, стохастический вариационный вывод через оптимизацию вариационной нижней грани правдоподобия, поддерживает обучение по подвыборкам, что делает модель масштабируемой в смысле числа данных.

Важно отметить, что хотя в некоторых случаях удалось получить высокое качество кластеризации на исходных данных без какой-либо предварительной обработки, подходы такого плана не дают возможности для последующей интерпретации внутренней структуры данных.

9. ВЫВОДЫ И ДАЛЬНЕЙШАЯ РАБОТА

Для связанного многомерного анализа компонент были предложены новые модели на основе блочного тензорного разложения объединенных вдоль новой оси данных с дополнительными условиями на параметры (GLRO, GTLD), а также вероятностная модель вариационного блочного тензорного разложения со слагаемыми в различных форматах. Вычисление параметров детерминистических моделей производилось через решение задачи нелинейных наименьших квадратов; для вычисления параметров вероятностной модели был использован метод стохастического вариационного вывода.

Как и матричные модели GICA и COBE, предложенные модели на основе BTD показали способность к приближению групповой и индивидуальной активности, что было проверено методом кластеризации контрастированных (т.е. сохранивших только индивидуальную часть) данных. Заметим, что результаты кластеризации зависят не только от метода контрастирования, но и от метода кластеризации. Вместе с тем, если общие и индивидуальные компоненты разложения потребуется использовать в смысле, отличном от признакового представления для кластеризации, то для выбора модели потребуется использование иного способа тестирования.

Одним из недостатков предложенных моделей GLRO и GTLD является большая вычислительная сложность, которая может быть компенсирована возможностью выделения факторов, специфичных для разных мод, в случае малого числа образцов. Для выделения зависимостей из больших наборов данных более перспективным выбором выглядит VBTD модель.

Дальнейшая работа над VBTD моделью включает рассмотрение распределений компонент, отличных от нормального, решение проблемы выбора рангов или увеличение степени разреженности через вероятностное моделирование, переход к полностью вероятностной версии разложения, в которой все параметры модели рассматриваются как случайные величины. Перспективным направлением является усовершенствование модели через искусственную тензоризацию данных.

ПРИЛОЖЕНИЕ

1. Использованные в работе обозначения

В настоящей статье использовались следующие обозначения.

Определение. Многомерный массив с вещественными элементами $X \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ называется (вещественным) *тензором*. Элемент тензора X, отвечающий мульти-индексу $(i_1, \ldots, i_d)$, обозначается как $X[i_1, \ldots, i_d]$. Его k-я ось называется также модой размера n_k. Любой тензор может быть записан как вектор-столбец с помощью операции векторизации. Соответствие между индексами оригинала и векторизированного результата задается правилом

$$f(i_1, \ldots, i_d) = 1 + \sum_{k=1}^{d} \left[(i_k - 1) \prod_{l=1}^{k-1} n_l \right], \quad \mathrm{vec}(X)[f(i_1, \ldots, i_d)] = X[i_1, \ldots, i_d], \quad i_k = \overline{1, n_k}, \quad k = \overline{1, d}.$$

Таблица 5. Использованные в работе обозначения

$\boxed{1}$	Массив из единиц	$X_{(k)}$	Матризация тензора X по оси k
$i = \overline{1, I}$	$i \in \{1, \ldots, I\}$	A^{+}	Псевдообращение Мура–Пенроуза
$(\ldots)_{\{k\}}$	Операция без k-го составляющего	$X \times_k Y$	Перемножение тензоров по k-й оси
$\operatorname{diag}(A)$	Главная диагональ матрицы	$\operatorname{diag}(\mathbf{v})$	Диагональная матрица, $\mathbf{v}$ – диагональ
$\operatorname{off}\operatorname{diag}(A)$	Матрица A с нулями на диагонали	$\operatorname{Diag}(\mathbf{p})$	Диагональный тензор, $\mathbf{p}$ – диагональ
$\widehat{\operatorname{vec}}(X)$	Векторизация тензора X (строка)	$\operatorname{vec}(X)$	Векторизация тензора X (столбец)
$\langle X, Y \rangle$	Внутреннее произведение	$X \circ Y$	Внешнее произведение двух тензоров
$\llbracket C_1, \ldots, C_d \rrbracket$	Канонический формат	$A \odot B$	Матричное произведение Хатри–Рао
$\llbracket G; A_1, \ldots, A_d \rrbracket$	Формат Таккера	$A \otimes B$	Матричное кронекерово произведение

Для преобразования тензора в вектор-строку использовано схожее обозначение: $\widehat{\operatorname{vec}}(\cdot)$. Для тензоров размерности d также вводится операция матризации, для чего в случае $d > 2$ объединяют его оси. Операцию можно вводить различными способами, здесь рассмотрим подход с сохранением в неизменном виде одной из осей с номером k. Будем называть такую операцию разверткой по индексу k: $\operatorname{unfold}_k(X^{n_1 \times \ldots \times n_k \times \ldots n_d}) = X_{(k)} = \hat{X}^{n_k \times \prod_{l \neq k} n_l}$. Кроме того, если у тензора есть мода с некоторым номером l, размер n_l которой может быть разложен на множители, $n_l = \prod_{p=1}^{P_l} n_{l,p}$, к такому тензору можно применить операцию *тензоризации* (искусственного повышения размерности). Использовано обозначение $\operatorname{Tens}(\cdot)$. Операция требует принять соглашение о пересчете индексов, в данной статье применялся стандартный порядок “по столбцам” (так называемый фортран-порядок).

Пусть имеется два тензора $X \in \mathbb{R}^{n_1^{(x)} \times \ldots \times n_k \times \ldots n_{d_1}^{(x)}}$, $Y \in \mathbb{R}^{n_1^{(y)} \times \ldots \times n_k \times \ldots n_{d_2}^{(y)}}$ размерностями d_1 и d_2, и пусть для них имеется общая мода с номером k размером n_k. В этом случае определим операцию *умножения по* k-й моде, $X \times_k Y = Z$, между X и Y по следующему поэлементному правилу:

$$Z[\ldots, i_{k-1}, i_{k+1}, \ldots, j_{k-1}, j_{k+1}, \ldots] = \sum\nolimits_{l=1}^{n_k} X[\ldots, i_{k-1}, l, i_{k+1}, \ldots] Y[\ldots, j_{k-1}, l, j_{k+1}, \ldots],$$

где $i_p = 1, \ldots, n_p^{(x)}$, $j_q = 1, \ldots, n_q^{(y)}$, $p = \overline{1, d_1}$, $q = \overline{1, d_2}$. В литературе данную операцию также называют свертыванием (см. [2]).

Фробениусова норма для тензора X определяется выражением вида $\|X\|_F^2 = \langle \operatorname{vec}(X), \operatorname{vec}(X) \rangle$, где $\langle \cdot, \cdot \rangle$ – скалярное (внутреннее) произведение между двумя векторами. Операцию можно обобщить на случай тензоров без дополнительной их векторизации: результат эквивалентен последовательному свертыванию аргументов-тензоров одинакового размера по всем их осям. Если же имеется некоторое подмножество осей, по которым перемножение выполнять не следует, то будем обозначать такую ситуацию следующим образом, используя скобки внутреннего произведения: $\langle X, Y \rangle_{\{\alpha \in \Omega\}} = X \times_{k \in \{1, \ldots, d\} \setminus \Omega} Y$. Иными словами, подстрочные скобки для операции означают, что операция выполняется для всех объектов (по всем индексам), кроме указанных внутри этих скобок. В дополнение к внутреннему, внешнее произведение $X \circ Y = Z$ между двумя тензорами X, Y определяется как $Z[i_1, \ldots, i_{d_1}, j_1, \ldots, j_{d_2}] = X[i_1, \ldots, i_{d_1}] \cdot Y[j_1, \ldots, j_{d_2}]$.

Отметим еще два важных произведения. Кронекерово произведение двух матриц A и B размерами (n_1, n_2) и (m_1, m_2) определяется как $(A \otimes B)[m_1(i-1) + k, m_2(j-1) + l] = A[i, j]B[k, l]$. Если обе матрицы A, B имеют одинаковое количество столбцов (т.е. выполнено $n_2 = m_2$), то для них определено произведение Хатри–Рао, которое удобно использовать в контексте развертки тензоров, представленных в каноническом формате: $(A \odot B)[m_1(i-1) + k, j] = A[i, j]B[k, j]$.

Для удобства ряд обозначений, использованных в работе, приведен в табл. 5. Более подробный разбор базовых операций и тензорных разложений можно найти в [1].

2. Структура матрицы Гессе для BTD разложения с тремя типами слагаемых

Пусть требуется приблизить данный тензор $T \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ в формате (8). Обозначим через $\mathbf{x}$ вектор-столбец, составленный из векторизации всех параметров модели:

$$\mathbf{x} = \left[\ldots \widehat{\mathrm{vec}}(C_k^{[s]}) \ldots \widehat{\mathrm{vec}}(A_k^{[m]}) \ldots \widehat{\mathrm{vec}}(G^{[m]}) \ldots \right]^{\mathrm{T}}, \tag{35}$$

где $C_k^{[s]}$ есть k-я фактор-матрица для s-го слагаемого в каноническом формате, $A_k^{[m]}$ есть k-я фактор-матрица для m-го слагаемого в формате Таккера, $G^{[m]}$ – ядро Таккера для m-го слагаемого в формате Таккера, и пусть $F = F(\mathbf{x})$ – восстановленный тензор с помощью текущего приближения параметров:

$$F(\mathbf{x}) = \sum\nolimits_{m=1}^{M} \left\| \left[G^{[m]}; A_1^{[m]}, \ldots, A_d^{[m]} \right] \right\| + \left\| [C_1, \ldots, C_d] \right\|$$

(как в (8)). Для дальнейших целей введем дополнительное обозначение $Z(\mathbf{x}) = F(\mathbf{x}) - T$. Соответствующая оптимизационная задача может быть сформулирована в виде задачи нелинейных наименьших квадратов, градиент и матрица Гессе которой имеют специальный вид:

$$\begin{gathered} \min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2} \| F(\mathbf{x}) - T \|_F^2, \quad H[f](\mathbf{x}) = J^{\mathrm{T}}(\mathbf{x}) J(\mathbf{x}) + Q(\mathbf{x}), \\ \nabla f(\mathbf{x}) = J^{\mathrm{T}}(\mathbf{x}) \mathrm{vec}(F(\mathbf{x}) - T), \quad Q(\mathbf{x}) = \sum_i \mathrm{vec}(Z(\mathbf{x}))[i] \cdot \nabla^2 (\mathrm{vec}(Z(\mathbf{x}))[i]). \end{gathered} \tag{36}$$

Здесь $J(\mathbf{x})$ – матрица Якоби для $f(\mathbf{x})$:

$$J(\mathbf{x}) = \left[(V_{\{k\}}^{[s]} \otimes I_{n_k}) \ldots (G_{(k)}^{[m]} V_{\{k\}}^{[m]\mathrm{T}} \otimes I_{n_k}) \ldots V^{[m]} \ldots \right], \tag{37}$$

$k = \overline{1,d}$, $m = \overline{1,M}$, $s = \overline{1,S}$ и $V_{\{k\}}^{[s]} = C_d^{[s]} \odot \ldots \odot C_{k+1}^{[s]} \odot C_{k-1}^{[s]} \odot \ldots \odot C_1^{[s]}$, $V_{\{k\}}^{[m]} = A_d^{[m]} \otimes \ldots \otimes A_{k+1}^{[m]} \otimes \otimes A_{k-1}^{[m]} \otimes \ldots \otimes A_1^{[m]}$, $V^{[m]} = A_d^{[m]} \otimes \ldots \otimes A_1^{[m]}$. Главная часть матрицы Гессе имеет следующую структуру:

$$J^{\mathrm{T}} J = \begin{bmatrix} (\mathrm{Gr}^{\mathrm{CC}})_{k_1,k_2;s_1,s_2;\cdot,\cdot} & (\mathrm{Gr}^{\mathrm{CA}})_{k_1,k_2;s_1,\cdot;\cdot,m_2} & (\mathrm{Gr}^{\mathrm{CG}})_{k_1,\cdot;s_1,\cdot;\cdot,m_2} \\ (\mathrm{Gr}^{\mathrm{AC}})_{k_1,k_2;\cdot,s_2;m_1,\cdot} & (\mathrm{Gr}^{\mathrm{AA}})_{k_1,k_2;\cdot,\cdot;m_1,m_2} & (\mathrm{Gr}^{\mathrm{AG}})_{k_1,\cdot;\cdot,\cdot;m_1,m_2} \\ (\mathrm{Gr}^{\mathrm{GC}})_{k_1,k_2;\cdot,s_2;m_1,\cdot} & (\mathrm{Gr}^{\mathrm{GA}})_{\cdot,k_2;\cdot,\cdot;m_1,m_2} & (\mathrm{Gr}^{\mathrm{GG}})_{\cdot,\cdot;\cdot,\cdot;m_1,m_2} \end{bmatrix}, \tag{38}$$

где $k_1, k_2 = \overline{1,d}, m_1, m_2 = \overline{1,M}, s_1, s_2 = \overline{1,S}$, и соответствующие блоки имеют вид

$$\begin{gathered} \mathrm{Gr}^{\mathrm{CC}}_{k_1,k_2;s_1,s_2;\cdot,\cdot} = (V_{\{k_1\}}^{[s_1]\mathrm{T}} \otimes I_{n_{k_1}}) P_{1,k_2}^{k_1,1} (V_{\{k_2\}}^{[s_2]} \otimes I_{n_{k_2}}), \\ \mathrm{Gr}^{\mathrm{CA}}_{k_1,k_2;s_1,\cdot;\cdot,m_2} = (V_{\{k_1\}}^{[s_1]\mathrm{T}} \otimes I_{n_{k_1}}) P_{1,k_2}^{k_1,1} (V_{\{k_2\}}^{[m_2]} G_{(k_2)}^{[m_2]\mathrm{T}} \otimes I_{n_{k_2}}), \\ \mathrm{Gr}^{\mathrm{CG}}_{k_1,k_2;s_1,\cdot;\cdot,m_2} = (V_{\{k_1\}}^{[s_1]\mathrm{T}} \otimes I_{n_{k_1}}) P_1^{k_1} (V^{[m_2]}), \\ \mathrm{Gr}^{\mathrm{AA}}_{k_1,k_2;m_1,m_2} = (G_{(k_1)}^{[m_1]} V_{\{k_1\}}^{[m_1]\mathrm{T}} \otimes I_{n_{k_1}}) P_{1,k_2}^{k_1,1} (V_{\{k_2\}}^{[m_2]} G_{(k_2)}^{[m_2]\mathrm{T}} \otimes I_{n_{k_2}}), \\ \mathrm{Gr}^{\mathrm{AG}}_{k_1,\cdot;\cdot,\cdot;m_1,m_2} = (G_{(k_1)}^{[m_1]} V_{\{k_1\}}^{[m_1]\mathrm{T}} \otimes I_{n_{k_1}}) P_1^{k_1} (V^{[m_2]}), \\ \mathrm{Gr}^{\mathrm{GG}}_{\cdot,\cdot;\cdot,\cdot;m_1,m_2} = (V^{[m_1]\mathrm{T}}) (V^{[m_2]}), \end{gathered} \tag{39}$$

здесь $P_{\cdot}^{\cdot}$ – коммутационные матрицы (“матрицы перестановки осей”), верхние и нижние индексы обозначают исходные и новые номера мод соответственно. Например, векторизация транспонирования матрицы A может быть записана в виде $P_{2,1}^{1,2} \mathrm{vec}(A) = \mathrm{vec}(A^{\mathrm{T}})$. В тензорном случае все прочие индексы при этом упорядочиваются согласно их естественной нумерации.

На практике слагаемое $Q(\mathbf{x})$ часто исключается по нескольким причинам: требует дополнительных вычислительных ресурсов, конфликтует с требованием оптимизационного метода на симметричность и положительную определенность приближения гессиана, а также по причине того, что в окрестности локального минимума его влиянием можно пренебречь. Вместе с тем

приводим выражения для вычисления данной квадратичной части гессиана, используя тензорное представление:

$$\operatorname{Tens}\left(\frac{\partial^2 \mathbf{z}}{\partial\widehat{\operatorname{vec}}(C_k^{[s_1]})\partial\widehat{\operatorname{vec}}(C_l^{[s_2]})}\cdot\mathbf{z}\right)=\begin{cases}\left\langle\left[\!\left[C_1^{[s_1]},\ldots,C_d^{[s_1]}\right]\!\right]\Big|_{C_k^{[s_1]},C_l^{[s_1]}=I_{L_{s_1}}},\ Z\right\rangle_{\{k,l\}}, & k\neq l,\quad s_1=s_2,\\ 0, \quad \text{иначе},\end{cases} \tag{40}$$

$$\operatorname{Tens}\left(\frac{\partial^2 \mathbf{z}}{\partial\widehat{\operatorname{vec}}(A_k^{[m_1]})\partial\widehat{\operatorname{vec}}(A_l^{[m_2]})}\cdot\mathbf{z}\right)=\begin{cases}\langle\underbrace{\left[\!\left[G^{[m_1]};A_1^{[m_1]},\ldots,A_d^{[m_1]}\right]\!\right]}_{A_k^{[m_1]}=I_{r_k^{[m_1]}},A_l^{[m_1]}=I_{r_l^{[m_1]}}},Z\rangle_{\{k,l\}}, & k\neq l,\quad m_1=m_2,\\ 0, \quad \text{иначе},\end{cases} \tag{41}$$

$$\operatorname{Tens}\left(\frac{\partial^2 \mathbf{z}}{\partial\widehat{\operatorname{vec}}(A_k^{[m_1]})\partial\widehat{\operatorname{vec}}(G^{[m_2]})}\cdot\mathbf{z}\right)=\begin{cases}\left[\!\left[Z;A_1^{[m_1]},\ldots,A_d^{[m_1]}\right]\!\right]\Big|_{A_k^{[m_1]}=I_{r_k^{[m_1]}}}, & m_1=m_2,\\ 0, \quad m_1\neq m_2,\end{cases} \tag{42}$$

$m_1,m_2=\overline{1,M}$, $k,l=\overline{1,d}$, $\mathbf{z}=\operatorname{vec}(Z(\mathbf{x}))$, $Z=Z(\mathbf{x})$, и все остальные вторые производные равны нулю. Важно отметить, что использование явного представления матрицы Гессе требует больших затрат по памяти. Для оптимизации расходования памяти имеет смысл использовать не полное представление, а использовать результат умножения матрицы на вектор, опираясь на структуру параметров, в том или ином итеративном методе, что и было сделано в используемой реализации.

3. Встраивание групповых условий в оптимизационную задачу

В данном приложении описано, как встроить сформулированные условия в оптимизационную задачу. Предложенная детерминистическая модель для группового анализа данных включает в себя условия на фактор-матрицу групповой оси, а также условие отделимости вида (9). Суммируем их для обоих вариантов модели, полной (Lr, 1), GLRO (6), и модели GTLD (7):

$$\begin{gathered}C_{d+1}=[I_N\ \ \mathbf{p}],\qquad A_{d+1}=\operatorname{diag}(\mathbf{p}),\\ p_1+\ldots+p_N=p_{\text{cum}},\quad p_i\geq p_{\min},\qquad p_1+\ldots+p_N=p_{\text{cum}},\quad p_i\geq p_{\min},\\ (C_\gamma^{[N+1]})^{\mathrm{T}}C_\gamma^{[i]}=0,\quad \gamma\in\Omega,\quad i=\overline{1,N},\qquad A_\gamma^{\mathrm{T}}C_\gamma^{[i]}=0,\quad \gamma\in\Omega,\quad i=\overline{1,N},\end{gathered} \tag{43}$$

где $p_{\min}$ и p_{cum} – фиксированные константы, Ω – набор мод для условия отделимости, N – общее число образцов. Существуют разные способы учета этих условий в задаче (36), обратимся к двум методам: проекционному и методу множителей Лагранжа. В проекционном методе каждая итерация состоит из шага обновления решения безусловной задачи оптимизации с последующим его проектированием на область допустимости, заданной через условия. Первая часть проектора, отвечающая условию отделимости, является проектированием столбцов индивидуальных фактор-матриц на пространство, ортогональное столбцам общей фактор-матрицы: $P_Y^{\perp}(X)=(I-Y(Y^{\mathrm{T}}Y)^{-1}Y^{\mathrm{T}})X$. Оставшаяся часть проектора является проекцией $\mathbf{p}$ на внутренность l_1 шара с вырезом в окрестности нуля и положительными значениями координат (см. [54]), $\{\mathbf{y}\in\mathbb{R}^N\,|\,y_1+\ldots+y_N=p_{\text{cum}},y_i\geq p_{\min}>0\}$.

Второй метод состоит в использовании множителей Лагранжа, сводящих условную задачу оптимизации к безусловной. Запишем соответствующее слагаемое целевого функционала $G(\mathbf{x})$:

$$\begin{gathered}G(\mathbf{x})\ =\sum_{\gamma\in\Omega}\sum_{i=1}^{N}\left\langle\operatorname{vec}(U_\gamma^{\mathrm{T}}C_\gamma^{[i]}),\hat{\boldsymbol{\mu}}_{\gamma,i}\right\rangle-\kappa\cdot\left(p_{\text{cum}}-\left\langle\mathbf{p},\boxed{1}\right\rangle\right)-\left\langle\zeta,\mathbf{p}-p_{\min}\cdot\boxed{1}\right\rangle,\\ U_\gamma=\begin{cases}C_\gamma^{[N+1])}, & \text{для модели (6)},\\ A_\gamma, & \text{для модели (7)},\end{cases}\qquad\begin{cases}C_{d+1}=[I_N\mathbf{p}], & \text{для модели (6)}\\ C_{d+1}=I_N,A_{d+1}=\operatorname{diag}(\mathbf{p}), & \text{для модели (7)}.\end{cases}\end{gathered} \tag{44}$$

Число дополнительных параметров можно уменьшить, ослабив условия на ортогональность групповых и индивидуальных фактор-матриц:

$$G(\mathbf{x})=\sum_{\gamma\in\Omega}\frac{\mu_\gamma}{2}\sum_{i=1}^{N}\left\|U_\gamma^{\mathrm{T}}C_\gamma^{[i]}\right\|_F^2-\kappa\cdot\left(p_{\text{cum}}-\left\langle\mathbf{p},\boxed{1}\right\rangle\right)-\left\langle\zeta,\mathbf{p}-p_{\min}\boxed{1}\right\rangle, \tag{45}$$

с теми же U_γ, что и в (44). Обозначим вектор производных по вспомогательным переменным для $G(\mathbf{x})$ как $g(\mathbf{x})$. Градиент для $g(\mathbf{x})$ в случае модели (7) имеет следующий вид:

$$\nabla g(\mathbf{x}) = [\nabla g_{\mu_{\gamma_1}}(\mathbf{x}) \ \ldots \ \nabla g_{\mu_{\gamma_{|\Omega|}}}(\mathbf{x}) \ \nabla g_{\kappa}(\mathbf{x}) \ \nabla g_{\zeta_1}(\mathbf{x}) \ \ldots \ \nabla g_{\zeta_N}(\mathbf{x})],$$

$$\nabla g_{\mu_{\gamma_j}}(\mathbf{x})^{\mathrm{T}} = [\ldots \ 0 \ \widehat{\mathrm{vec}}(A_{\gamma_j} A_{\gamma_j}^{\mathrm{T}} C_{\gamma_j}) \ 0 \ \ldots \ 0 \ \widehat{\mathrm{vec}}(C_{\gamma_j} C_{\gamma_j}^{\mathrm{T}} A_{\gamma_j}) \ 0 \ \ldots], \tag{46}$$

$$\nabla g_{\kappa}(\mathbf{x})^{\mathrm{T}} = [\ldots \ 0 \ \boxed{1}_{1\times N} \ 0 \ \ldots], \quad \nabla g_{\zeta_i}(\mathbf{x})^{\mathrm{T}} = \Bigg[\ldots \ 0 \ \underbrace{\underset{\smile}{1}}_{i\text{-я позиция } \mathbf{p}} \ 0 \ \ldots\Bigg].$$

Наконец, каждая итерация требует решения системы с расширенной матрицей Гессе, $\tau = [\mu_{\gamma_1}, \ldots, \mu_{\gamma_{|\Omega|}}, \kappa, \zeta_1, \ldots, \zeta_N]^{\mathrm{T}}$:

$$\begin{bmatrix} H[f](\mathbf{x}) & \nabla g(\mathbf{x}) \\ \nabla g(\mathbf{x})^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\tau \end{bmatrix} = \begin{bmatrix} -\nabla f(\mathbf{x}) \\ -g(\mathbf{x}) \end{bmatrix}. \tag{47}$$

СПИСОК ЛИТЕРАТУРЫ

1. *Kolda T.G., Bader B.W.* Tensor decompositions and applications // SIAM Rev. 2009. V. 51 Iss. 3. P. 455–500.
2. *Cichocki A., Lee N., Oseledets I., Phan A.-H., Zhao Q., Mandic D.P.* Tensor networks for dimensionality reduction and large-scale optimization: Part 1, low-rank tensor decompositions // Found. Trends Mach. Learn. 2016. V. 9. Iss. 4–5. P. 249–429.
3. *Sidiropoulos N.D., De Lathauwer L., Fu X., Huang K., Papalexakis E.E., Faloutsos C.* Tensor decomposition for signal processing and machine learning // IEEE Trans. Signal Proc. 2017. V. 65. Iss. 13. P. 3551–3582.
4. *Phan A.-H., Cichocki A.* Tensor decompositions for feature extraction and classification of high dimensional datasets // IEICE Nonlin. Theory and its App. 2010. V. 1. Iss. 1. P. 37–68.
5. *De Lathauwer L.* Decompositions of a higher-order tensor in block terms – part I: Lemmas for partitioned matrices // SIAM J. Matrix Anal. Appl. 2008. V. 30. Iss. 3. P. 1022–1032.
6. *De Lathauwer L.* Decompositions of a higher-order tensor in block terms – part II: Definitions and uniqueness // SIAM J. Matrix Anal. Appl. 2008. V. 30. Iss. 3. P. 1033–1066.
7. *De Lathauwer L., Nion D.* Decompositions of a higher-order tensor in block terms – part III: Alternating least squares algorithms // SIAM J. Matrix Anal. Appl. 2008. V. 30. Iss. 3. P. 1067–1083.
8. *Prasad G., Jahanshad N., Aganj I., Lenglet C., Sapiro G., Toga A.W., Thompson P.M.* Atlas-based fiber clustering for multi-subject analysis of high angular resolution diffusion imaging tractography // Proc. IEEE Int. Symp. Biomed. Imaging. 2011. P. 276–280.
9. *Calhoun V.D., Adali T.* Multisubject independent component analysis of fMRI: a decade of intrinsic networks, default mode, and neurodiagnostic discovery // IEEE Rev. Biomed. Eng. 2012. V. 5. P. 60–73.
10. *Beckmann C.F., Smith S.M.* Tensorial extensions of independent component analysis for multisubject fMRI analysis. // Neuroimage. 2005. V. 25. Iss. 1. P. 294–311.
11. *Nazarenko D.V., Kharyuk P.V., Oseledets I.V., Rodin I.A., Shpigun O.A.* Machine learning for LC–MS medicinal plants identification // Chemom. Intell. Lab. Syst. 2016. V. 156. P. 174–180.
12. *Xia P., Bai Z., Liang T., Yang D., Liang Z., Yan X., Liu Y.* High-performance liquid chromatography based chemical fingerprint analysis and chemometric approaches for the identification and distinction of three endangered panax plants in Southeast Asia // J. Sep. Sci. 2016. V. 39. Iss. 20. P. 3880–3888.
13. *Smilde A., Bro R., Geladi P.* Multi-way analysis: applications in the chemical sciences // UK: John Wiley & Sons, 2005.
14. *Cichocki A., Mandic D., De Lathauwer L., Zhou G., Zhao Q., Caiafa C., Phan A.H.* Tensor decompositions for signal processing applications: From two-way to multiway component analysis // IEEE Signal Process. Mag. 2015. V. 32. Iss. 2. P. 145–163.
15. *Zhou G., Zhao Q., Zhang Y., Adalı T., Xie Sh., Cichocki A.* Linked component analysis from matrices to high-order tensors: Applications to biomedical data // Proc. IEEE. 2016. V. 104. Iss. 2. P. 310–331.
16. *Харюк П.В.* Групповой анализ данных на основе блочного канонического разложения // Сб. тезисов 59-й научной конференции МФТИ. М.: МФТИ, 2016.
17. *Харюк П.В.* Классификация сигналов с помощью блочного тензорного разложения в задаче группового анализа данных // Сб. тезисов XXIV Международной научной конференции Ломоносов-2017. М.: ООО “МАКС Пресс”, 2017. С. 152–153.
18. *Calhoun V.D., Liu J., Adalı T.* A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data. Neuroimage, 2009. V. 45. Iss. 1. P. S163–S172.
19. *Zhou G., Cichocki A., Zhang Y., Mandic D.P.* Group component analysis for multiblock data: Common and individual feature extraction. IEEE Trans. Neural Netw. Learn. Syst. 2016. V. 27. Iss. 11. P. 2426–2439.
20. *Lock E.F., Hoadley K.A., Marron J.S., Nobel A.B.* Joint and individual variation explained (JIVE) for integrated analysis of multiple data types // Ann. Appl. Stat. 2013. V. 7. Iss. 1. P. 523.
21. *Sorber L., Van Barel M., De Lathauwer L.* Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-(Lr, Lr, 1) terms, and a new generalization // SIAM J. Optim. 2013. V. 23. Iss. 2. P. 695–720.

22. *Tipping M.E., Bishop C.M.* Mixtures of probabilistic principal component analyzers // Neural Comput. 1999. V. 11. Iss. 2. P. 443–482.
23. *Tipping M.E., Bishop C.M.* Probabilistic principal component analysis // J. R. Stat. Soc. Series B Stat. Methodol. 1999. V. 61. Iss. 3. P. 611–622.
24. *Bishop C.M.* Pattern recognition and machine learning // NY: Springer, 2006.
25. *Oseledets I.V.* Tensor-train decomposition // SIAM J. Sci. Comput. 2011. V. 33. Iss. 5. P. 2295–2317.
26. *Kingma D.P., Welling M.* Auto-encoding variational bayes // arXiv:1312.6114, 2013.
27. *Kharyuk P., Nazarenko D., Oseledets I., Rodin I., Shpigun O., Tsitsilin A., Lavrentyev M.* Employing fingerprinting of medicinal plants by means of LC-MS and machine learning for species identification task // Sci. Rep. 2018. V. 8. Iss. 1. N. 17053.
28. *Björck A., Golub G.H.* Numerical methods for computing angles between linear subspaces // Math. of Computat. 1973. V. 27. Iss. 123. P. 579–594.
29. *Vervliet N., Debals O., Sorber L., Van Barel M., De Lathauwer L.* Tensorlab user guide // 2016.
30. *Blei D., Ranganath R., Mohamed S.* Variational inference: Foundations and modern methods // NIPS Tutorial, 2016.
31. *Wingate D., Weber T.* Automated variational inference in probabilistic programming arXiv:1301.1299, 2013.
32. *Bingham E., Chen J.P., Jankowiak M., Obermeyer F., Pradhan N., Karaletsos T., Singh R., Szerlip P., Horsfall P., Goodman N.D.* Pyro: Deep universal probabilistic programming // J. Mach. Learn. Res. 2019. V. 20. Iss. 1. P. 973–978.
33. *Paszke A., Gross S., Chintala S., Chanan G., Yang E., DeVito Z., Lin Z., Desmaison A., Antiga L., Lerer A.* Automatic differentiation in PyTorch // NIPS Autodiff Workshop, 2017.
34. *Calhoun V.D., Adali T., Pearlson G.D., Pekar J.J.* A method for making group inferences from functional MRI data using independent component analysis // Hum. Brain Mapp. 2001. V. 14. Iss. 3. P. 140–151.
35. *Guo Y., Pagnoni G.* A unified framework for group independent component analysis for multi-subject fMRI data // NeuroImage. 2008. V. 42. Iss. 3. P. 1078–1093.
36. *Lee H., Choi S.* Group nonnegative matrix factorization for EEG classification // Artif. Int. Stat. 2009. P. 320–327.
37. *Yokota T., Cichocki A., Yamashita Y.* Linked PARAFAC/CP tensor decomposition and its fast implementation for multi-block tensor analysis // Springer Int. Conf. Neural Inform. Proc. 2012. P. 84–91.
38. *Cichocki A., Zdunek R., Amari S.-I.* Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization // Springer Int. Conf. Indep. Comp. Anal. Sig. Sep. 2007. P. 169–176.
39. *Gillis N., Glineur F.* Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization // Neural Comput. 2012. V. 24. Iss. 4. P. 1085–1105.
40. *Yokota T., Cichocki A.* Linked Tucker2 decomposition for flexible multi-block data analysis // Springer Int. Conf. Neural Inform. Proc. 2014. P. 111–118.
41. *Gong X.F., Lin Q.-H., Debals O., Vervliet N., De Lathauwer L.* Coupled rank-(Lm, Ln, 1) block term decomposition by coupled block simultaneous generalized Schur decomposition // IEEE ICASSP. 2016. P. 2554–2558.
42. *Khoromskij B., Khoromskaia V.* Low rank tucker-type tensor approximation to classical potentials // Open Math. 2007. V. 5. Iss. 3. P. 523–550.
43. *Continuum Analytics.* Anaconda software distribution // [Электронный ресурс] vers. 2-2.4.0, 11.2015. Дата обращения: 1.11.2019.
44. *Oliphant T.E.* A guide to NumPy // USA: Trelgol Publ. USA. V. 1. 2006.
45. *Jones E., Oliphant T., Peterson P.* SciPy: Open source scientific tools for Python // [Электронный ресурс] 2001. Дата обращения: 1.11.2019.
46. *McKinney W.* Data structures for statistical computing in python // Proceed. of the 9th Python in Sci. Conf. V. 2010. 445, P. 51–56.
47. *Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., DuchesnayE.* Scikit-learn: machine learning in Python // J. of Mach. Learn. Res. 2011. V. 12. P. 2825–2830.
48. *Hunter J.D.* Matplotlib: A 2D graphics environment // Comput. Sci. Eng. 2007. V. 9. Iss. 3. P. 90–95.
49. *Waskom M., Botvinnik O., O'Kane D., Hobson P., Lukauskas S., Gemperline D.C., Augspurger T., Halchenko Y., Cole J.B., Warmenhoven J., de Ruiter J., Pye C., Hoyer S., Vanderplas J., Villalba S., Kunter G., Quintero E., Bachant P., Martin M., Meyer K., Miles A., Ram Y., Yarkoni T., Williams M.L., Evans C., Fitzgerald C., Brian, Fonnesbeck C., Lee A., Qalieh A.* Seaborn: statistical data visualization // [Электронный ресурс], v.0.8.1, 09.2017. Дата обращения: 1.11.2019.
50. *Kluyver T., Ragan-Kelley B., Pérez F., Granger B., Bussonnier M., Frederic J., Kelley K., Hamrick J., Grout J., Corlay S., Ivanov P., Avila D., Abdalla S., Willing C.* Jupyter notebooks – a publishing format for reproducible computational workflows // ELPUB 2016. IOS Press, P. 87–90.
51. *Leibe B., Schiele B.* Analyzing appearance and contour based methods for object categorization // IEEE CVPR. 2003. V. 2.
52. *Hastie T., Tibshirani R., Friedman J., Franklin J.* The elements of statistical learning: data mining, inference and prediction // Springer Series in Statistics, 2009.
53. *Khoromskij B.N., Oseledets I.V.* QTT approximation of elliptic solution operators in higher dimensions // Russ. J. Numer. Anal. Math. Model. 2011. V. 26. Iss. 3. P. 303–322.
54. *Gupta M.D., Kumar S., Xiao J.* L1 projections with box constraints. // arXiv:1010.0141, 2010.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 865–877

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ

УДК 517.977.54

TT-QI: УСКОРЕННАЯ ИТЕРАЦИЯ ФУНКЦИИ ЦЕННОСТИ В ФОРМАТЕ ТЕНЗОРНОГО ПОЕЗДА ДЛЯ ЗАДАЧ СТОХАСТИЧЕСКОГО ОПТИМАЛЬНОГО УПРАВЛЕНИЯ[1)]

© 2021 г. А. И. Бойко[1,*], И. В. Оселедец[1,2,], Г. Феррер[1]**

[1] *121205 Москва, Большой бульвар, 30, стр. 1, Сколковский институт науки и технологий, Россия*

[2] *119333 Москва, ул. Губкина, 8, ИВМ РАН, Россия*

**e-mail: alexey.boyko@skolkovotech.ru*

***e-mail: i.oseledets@skoltech.ru*

Поступила в редакцию 24.11.2020 г.
Переработанный вариант 24.11.2020 г.
Принята к публикации 14.01.2021 г.

Рассматривается задача стохастического оптимального управления общего вида с малым винеровским шумом. Данная задача аппроксимируется с помощью марковского процесса принятия решений. Решение уравнения Беллмана на функцию ценности вычисляется с помощью метода итерации ценности (VI) в формате малорангового тензорного поезда (TT-VI). Предложена модификация данного алгоритма (TT-QI): нелинейный оператор Беллмана итеративно применяется сначала с использованием малоранговых алгебраических операций, а затем с использованием алгоритма крестовой аппроксимации. Показана более низкая, чем в основном методе, сложность на одну итерацию в случае малых TT-рангов тензоров вероятностей перехода. На примере задач управления обратным маятником и машинами Дубинса показано ускорение времени расчета оптимального регулятора в 3–10 раз по сравнению с существующим методом. Библ. 13. Фиг. 6. Табл. 1.

Ключевые слова: динамическое программирование, оптимальное управление, марковские процессы принятия решений, малоранговые разложения.

DOI: 10.31857/S0044466921050045

1. ВВЕДЕНИЕ

Задачи оптимального управления часто возникают в различных областях робототехники. Для многих типовых задач были разработаны решения с помощью оптимизации PID-регуляторов, линейно-квадратичных регуляторов, либо более общих уравнения Риккатти и принципа максимума Понтрягина.

Однако для многих новых задач робототехники, таких как динамическое управление шагающими роботами на ландшафте произвольной формы или акробатическое маневрирование колесными и летающими роботами далеко за областью линейности и при воздействии случайных возмущений, синтез оптимального регулятора в общем виде может быть осуществлен только с помощью уравнения Беллмана.

Стохастические динамические системы могут быть представлены (с точностью до погрешности дискретизации) как марковский процесс принятия решений (см. [1], [2]). Формулировка задачи оптимального управления на языке уравнения Беллмана также позволяет нахождение оптимального регулятора в случае произвольной нелинейной, разрывной или точечной награды. Это делает возможным переписать на беллмановский язык такие задачи, как терминальную задачу об оптимальном быстродействии, задачу о минимальном затраченном топливе (с произвольной, а не только квадратичной характеристикой), задачу максимизации вероятности попадания в целевую область.

Ключевой проблемой в таком подходе является тот факт, что решение задачи стохастического оптимального управления на языке Беллмана страдает от “проклятия размерности”, так как

[1)]Работа выполнена при частичной финансовой поддержке Минобрнауки РФ (проект 14.756.31.0001).

имеет крайне высокую асимптотику вычислительной сложности. Сложность по памяти растет как $O(N^{d_s+d_a})$, где N – количество узлов дискретизации сетки по одной координате, а d_s и d_a – размерности пространства состояний и пространства управляющих сигналов соответственно. Например, для простого беспилотного летательного аппарата $d_s + d_a = 16$.

Один из методов обойти данную проблему – использовать дифференцируемые функциональные аппроксиматоры, например нейронные сети. Такой подход изначально был развит Бертсекасом под названием *нейродинамического программирования* (см. [3]), а позже стал известен как (глубокое) машинное обучение с подкреплением. К настоящему моменту действие в этом направлении привело к значительным успехам в решении задач управления существенно нелинейными малоприводными системами, в том числе с неголономными связями (см. [4]). Пожалуй, самым выдающимся примером этого является синтез регулятора для движения бега в подробной многозвенной малоприводной математической модели человека, управляемой более чем 100 мышцами (см. [5]). Такой подход, однако, имеет свои недостатки. Главным образом, это очень высокие требования к количеству данных (симуляций Монте-Карло) и к вычислительным мощностям, а также необходимость вручную подбирать различные параметры оптимизатора и топологию нейросети и отсутствие гарантий сходимости оптимального управления к глобальному оптимуму даже после длительной оптимизации.

Другой способ обойти "проклятие размерности" уравнения Беллмана – это использовать малоранговые тензорные разложения. Эффективность такого подхода для решения задачи оптимального управления была продемонстрирована Н. Хоровицем (см. [6]) из Калифорнийского технологического института с использованием линеаризованного уравнения Гамильтона–Якоби–Беллмана для контрольно-аффинных систем совместно с каноническим тензорным разложением. Несколько иной метод был предложен А. Городетским из Массачуссетского технологического института с использованием нелинейного уравнения Беллмана для систем общего вида и TT-разложения (см. [7]) и его дифференцируемого обобщения (см. [8], [9]). В этих статьях были найдены решения различных нелинейных малоприводных задач управления единообразным подходом, в том числе задачи акробатического пилотажа квадрокоптером далеко за областью линейности, используя лишь однопроцессорную многоядерную рабочую станцию. В данной статье мы рассматриваем именно подход, описанный А. Городетским.

2. ЗАДАЧА СТОХАСТИЧЕСКОГО ОПТИМАЛЬНОГО УПРАВЛЕНИЯ

Рассмотрим стохастическую динамическую полностью наблюдаемую систему, задаваемую уравнением

$$ds_i(t) = b_i(s,a)dt + \sigma_{ij}(s)dw. \tag{1}$$

Система в каждый момент времени описывается вектором состояния $s \in \mathcal{S}$ и вектором действия $a \in \mathcal{A}$ (также иногда называемым управляющим сигналом). В качестве пространства состояний системы $\mathcal{S}$ и пространства управляющих сигналов $\mathcal{A}$ рассматриваются d_s и d_a – мерные области, порожденные произведением замкнутых отрезков

$$s \in \mathcal{S} = [s_{\min}^{(1)}, s_{\max}^{(1)}] \times \ldots \times [s_{\min}^{(d_s)}, s_{\max}^{(d_s)}],$$
$$a \in \mathcal{A} = [a_{\min}^{(1)}, a_{\max}^{(1)}] \times \ldots \times [a_{\min}^{(d_a)}, a_{\max}^{(d_a)}],$$

dw – стандартный броуновский шум (винеровский процесс). Даны функции *дрейфа* $b(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_s}$ и *диффузии* $\sigma_{ij}(s) : \mathcal{S} \to \mathbb{R}^{d_s^2}$. Для постановки задачи оптимального управления также зададимся *моментальной функцией награды* $r(s,a)$, зависящей от текущего состояния системы s и текущего управляющего сигнала a, а также коэффициентом дисконтирования β. Функции $b(s,a)$, $\sigma_{ij}(s)$ полагаются ограниченными на своих областях определений.

На функцию $r(s,a)$ накладывается следующее ограничение:

$$|r(s,a)| \le C(1 + |s|^k + |a|^k), \tag{2}$$

где $C \in \mathbb{R}_+$ и $k \in \mathbb{N}$ – некоторые константы.

Потребуем выполнения еще одного условия: матрично-значная функция диффузии $\sigma_{ij}(s)$ является диагональной ($\sigma_{ij} = 0$ если $i \neq j$). Такая ситуация имеет место, например, в случае присутствия нескоррелированного шума в датчиках, измеряющих текущее состояние системы s.

Ставится следующая задача: найти оптимальную детерминистическую политику (в зависимости от области науки также называемую оптимальным управлением, регулятором или стратегией) $\pi^*(s) : \mathcal{S} \to \mathcal{A}$. Оптимальность здесь подразумевается в следующем смысле: если данная политика используется для генерации сигнала управления в каждый момент времени $a(t) = \pi^*\big(s(t)\big)$, то матожидание суммарной награды (в общем случае с дисконтированием) по реализациям всех возможных случайных траекторий за все доступное время будет максимальным:

$$\pi^*(s) = \arg\max_a \left[\mathbb{E} \int_{t=0}^{T} \exp(-\beta t) r(s(t), a) dt \right], \tag{3}$$

где β – коэффициент дисконтирования в непрерывном времени, T – время конца эпизода (для дисконтированных нетерминальных задач управления $T = \infty$). Необходимым образом требуется, чтобы матожидание награды было ограничено: $\mathbb{E}\int_{t=0}^{T} e^{-\beta t} r(s(t), a) dt < \infty$.

3. АППРОКСИМАЦИЯ МАРКОВСКИМ ПРОЦЕССОМ ПРИНЯТИЯ РЕШЕНИЙ

Говорят, что задан *марковский процесс принятия решений* (МППР) в дискретном времени, если задан следующий кортеж:

$$(\mathcal{S}, \mathcal{A}, T, R, \gamma), \tag{4}$$

где заданы конечное множество допустимых значений состояний $\mathcal{S}$, конечное множество допустимых значений действий (управляющих сигналов, управляющих решений) $\mathcal{A}$, известны все (условные) вероятности перехода из любого состояния s в любое состояние s' через любое действие a: $T(s, a, s') = P(s' | s, a)$, а также функция награды для любого такого перехода $R(s, a, s')$ и коэффициент дисконтирования $\gamma \in (0, 1]$. Если $\gamma = 1$, то МППР называется *недисконтированным*.

Для численного нахождения решения исходной задачи стохастического оптимального управления (1) дискретизуем ее. Общая теория дискретизации для таких задач может быть найдена в книгах В. Флеминга (см. [2]) и Г. Кушнера (см. [1]). Основная идея этой теории в том, чтобы спроектировать непрерывные пространства состояний и действий на сетку, а затем построить МППР, который был бы эквивалентен нашему стохастическому процессу управления (1) в смысле дрифта и диффузии за единицу времени:

$$\lim_{h \to 0} \frac{\mathbb{E}[s_{t+1} - s_t \,|\, s_t = s,\ a_t = a]}{\Delta t} = b(s, a),$$

$$\lim_{h \to 0} \frac{\mathrm{Cov}[s_{t+1} - s_t \,|\, s_t = s,\ a_t = a]}{\Delta t} = \sigma(s, a).$$

Здесь $h = \min(h_i)$, а h_i – шаги дискретизации равномерной сетки вдоль i-й оси: $h_i = (s^{(i)}_{\max} - s^{(i)}_{\min})/(N^{(i)}_s - 1)$.

Мы будем использовать модифицированную версию схемы расщепления против потока, предложенной в [9], построенной по общей методике Г. Кушнера. Данная схема разрешает переходы с ненулевой вероятностью только между соседними узлами сетки, и при этом все диагональные переходы запрещены. С учетом того, что диффузионный член $\sigma_{ij}(s)$ диагонален, схема дискретизации принимает следующий вид:

$$Q^h = \max_{s,a} \left(\sum_i \frac{|b_i(s, a)|}{h_i} + \frac{\sigma_i^2(s)}{h_i^2} \right),$$

$$\Delta t^h = 1/Q^h,$$

$$b^+ = \begin{cases} b(s, a), & \text{если} \quad b(s, a) > 0, \\ 0 & \text{иначе}, \end{cases}$$

$$b^{-} = \begin{cases} -b(s,a), & \text{если} \quad b(s,a) < 0, \\ 0 & \text{иначе}, \end{cases} \tag{5}$$

$$T(s,a,s \pm e_i h_i) = \Delta t^h \left(\frac{b_i^{\pm}(s,a)}{h_i} + \frac{\sigma_i^2(s)}{2h_i^2} \right),$$

$$T(s,a,s) = 1 - \sum_{s'} T(s,a,s' \neq s),$$

$$R(s,a,s') = r(s,a)\Delta t^h,$$

$$\gamma = e^{-\beta \Delta t^h},$$

где $s \pm e_i h_i$ обозначает дискретное состояние (узел сетки), которое является соседним по отношению к узлу s вдоль i-й оси, в направлении увеличения или уменьшения индекса соответственно.

4. МАЛОРАНГОВОЕ РАЗЛОЖЕНИЕ В ТЕНЗОРНЫЙ ПОЕЗД

Рассмотрим d-мерный массив (тензор) $V(i_1, \ldots, i_d)$. Допустим, мы хотим найти его представление в следующем виде:

$$V(i_1, i_2, \ldots, i_d) = \sum_{m_1, \ldots, m_{d-1}}^{r_1, r_2, \ldots, r_{d-1}} G^{(1)}(i_1, m_1) G^{(2)}(m_2, i_2, m_2) \cdot \ldots \cdot G^{(d)}(m_{d-1}, i_d). \tag{6}$$

Такое разложение называется *разложением в тензорный поезд* или *TT-разложением*. Числа $r_1, \ldots, r_{d-1}$ называются *TT-рангами*. Чтобы равенство (6) выполнялось точно, необходимо хранить в худшем случае $O(dN^3)$ чисел из вместо $O(N^d)$. Зачастую имеет смысл найти такой тензорный поезд, что данное равенство соблюдается с некоторой погрешностью (в смысле среднеквадратичной ошибки), но TT-ранги малы ($\forall i : r_i < 100$). В таком случае можно ожидать, что функция на сетке будет сжата с требованием по памяти, равным $O(dNR^2)$, где $R = \max(r_i)$.

Важным фактом является то, что для достаточно широкого класса функций, спроецированных на равномерную сетку (более полное описание дано в [10]), если мы зафиксируем наибольшую допустимую ошибку аппроксимации, при измельчении сетки TT-ранги не будут расти, либо будут расти лишь логарифмически. Отметим, что в дополнение к сжатию в тензорных поездах сохраняется возможность выполнять алгебраические ($+$, $-$, $*$, $\otimes$, $\circ$) и линейно-алгебраические операции (внешние и внутренние произведения, свертки, суммирования по индексам), операции взятия подмассива и доступа к индивидуальным элементам, а также применять произвольные функции к тензорным поездам с помощью алгоритмов из семейства многомерной крестовой аппроксимации (см. [11]).

5. УРАВНЕНИЕ БЕЛЛМАНА

Метод нахождения глобально-оптимальной политики решений для марковского процесса принятия решений был предложен Р. Беллманом в свом классическом труде [12]. Рассмотрим его основные положения.

Зададимся МППР ($\mathcal{S}$, $\mathcal{A}$, T, R, γ). Задачей является найти такую управляющую политику, которая, стартуя из любого состояния s, максимизирует матожидание (возможно, дисконтированной) награды за все последующее время:

$$V(s) = \mathbb{E} \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \to \max.$$

Величина $V(s)$ называется *функцией ценности* (value function) и является ключевым элементом исследований в стохастическом оптимальном управлении и науке о машинном обучении с подкреплением.

Один из главных результатов Беллмана заключался в следующем: глобально оптимальная политика (управление, регулятор) для МППР не зависит от предыстории, а зависит только от текущего состояния. Из этого следует, что для глобальной оптимальности политики достаточно, что-

бы действие было локально оптимально, но не в смысле текущего приращения функции награды $R(s,a)$, а в смысле текущего приращения функции ценности $V(s)$.

Рекуррентные соотношения значений функции ценности в соседних состояниях, разделенных всего одним временным шагом, называются *уравнениями Беллмана* и имеют вид

$$V^*(s) = \max_a \left[R(s,a) + \gamma \sum_{s'} V^*(s') T(s,a,s') \right]. \tag{7}$$

Для краткости мы использовали следующее обозначение:

$$R(s,a) = \mathbb{E}_{s'} R(s,a,s') = \sum_{s'} R(s,a,s') T(s,a,s').$$

Если рассмотреть правую часть уравнения (7) как оператор $\hat{\beta}$, действующий на функцию $V(s)$, то такой оператор в литературе называется *оператором оптимальности Беллмана* (Bellman Optimality Operator).

Уравнение на оптимальную политику выглядит следующим образом:

$$\pi^*(s) = \arg\max_a \left[R(s,a) + \gamma \sum_{s'} V^*(s') T(s,a,s') \right]. \tag{8}$$

Заметим, что ряд задач оптимального управления (называемых *терминальными*), например, задача оптимального быстродействия, требуют задания терминальной области. При попадании в терминальную область счетчик времени останавливается, и дальшейшее накопление награды невозможно. Чтобы математически ввести терминальную область $\mathcal{S}_{\text{terminal}} \subset \mathcal{S}$ на языке Беллмана, необходимо положить нулем все вероятности переходов для состояний из данной области, кроме переходов каждого состояния само в себя: $\forall (s,a) \in \mathcal{S}_{\text{terminal}} \times \mathcal{A} : T(s,a,s') = \delta_{ss'}$, где $\delta_{ss'}$ – символ Кронекера. Также функция награды должна быть положена нулем для всех переходов, начинающихся из терминальной области: $\forall s \in \mathcal{S}_{\text{terminal}} : R(s,a) = 0, V(s) = 0$.

5.1. Итерация функции ценности в формате тензорного поезда

Уравнение Беллмана является уравнением на неподвижную точку для нелинейного оператора оптимальности Беллмана $\hat{\beta}$, и оно может быть решено методом простых итераций. Такой подход был предложен самим Р. Беллманом в его классической работе [12] и называется методом *итерации функции ценности* (Value Iteration, VI). Он использует то обстоятельство, что оператор $\hat{\beta}$ является сжимающим отображением почти во всех практически применимых случаях (в случае дисконтированных задач, либо в случаях недисконтированных задач с терминальной областью, достижимой хотя бы одной политикой).

Для обоснования использования малорангового разложения рассмотрим количество занимаемой памяти для МППР, аппроксимирующего задачу стохастического оптимального управления (1). Поскольку используется равномерная сетка дискретизации вдоль каждой оси пространства состояний в N_s узлов, результирующая сложность по памяти для хранения оптимальной функции ценности $V^*(s)$ и оптимальной политики $\pi^*(s)$ будет составлять $O(N_s^{d_s})$ для каждой из этих функций. Даже простые мобильные робототехнические аппараты, такие как БПЛА самолетного типа или квадрокоптеры, задаются $d_s = 12$-мерным вектором состояния и минимум $d_a = 3$-мерным вектором управляющих сигналов. В итоге при достаточно грубой дискретизации сетки в 100 узлов по каждой координате в задаче возникает минимум $N_s^{d_s} = 10^{24}$ элементов, что делает прямое численное нахождение решения в беллмановском формализме невозможным. Если при поиске решения вероятности переходов $T(s,a,s')$ не вычисляются каждый раз, а хранятся, то дополнительно потребуется сохранить еще $(2d_s + 1)N_s^{d_s} N_a^{d_a}$ чисел (где N_a – число узлов дискретизации вдоль осей пространства управляющих сигналов, а d_a – размерность этого пространства).

В статье [7], предложенной в Массачуссетском технологическом институте А. Городетским, предлагалось хранить функцию ценности в виде малорангового тензорного поезда и применять оператор оптимальности Беллмана в рамках итерации функции ценности методом крестовой аппроксимации для тензорных поездов (см. [11]). В случае малых TT-рангов функции ценности

ее сложность по памяти понижается с $O(N^d)$ до $O(dNr^2)$, что делает задачу вычислительно решаемой даже на маломощном компьютере. Также в [7], [9] показано, что в TT-формате оператор $\hat{\beta}$ сохраняет свойство быть сжимающим отображением, и гарантии сходимости, выведенные для изначальной итерации функции ценности (см. [12]), сохраняются:

Data: $R(s,a), \gamma, T(s,a,s')$

Result: $V^*(s)$

while $\epsilon = \dfrac{\left\|V^{(k+1)} - V^{(k)}\right\|_2}{\left\|V^{(k)}\right\|_2} < tol$ **do**

$V^{k+1} = TTCROSS((7), \epsilon)$

$k = k + 1$

end

Алгоритм 1: TT-Value Iteration (TT-VI)

Комбинация тензорных разложений и беллмановской постановки задачи управления показала отличные результаты для квадратичных и терминальных задач управления для ряда нелинейных неаффинных систем управления, соответствующих различным малоприводным робототехническим платформам: обратному маятнику, машине Дубинса, планеру, квадрокоптеру (см. [6], [7]), в том числе при наличии стохастического воздействия в системе.

Для дальнейших рассуждений о сложности алгоритмов введем следующие обозначения:

$$R_V = \max \operatorname{rank} V(s,a),$$
$$R_P = \max_{i=0,1,\ldots d_s,\pm} \operatorname{rank} P_i^{\pm}(s,a).$$

В наивной имплементации TT-VI алгоритма крестовая аппроксимация требует $O(dNR_V^3)$ на каждое применение оператора оптимальности Беллмана, а также требуется $O(dR_V^2)$ операций для распаковывания значений из TT в каждой точке, что в итоге приводит к вычислительной сложности $O(d^2NR_V^5)$.

Однако вычисления в алгоритме TT-VI возможно оптимизировать. Для этого вспомним, что метод крестовой аппроксимации считывает значения функции в точках, которые сгруппированы в виде строк, столбцов и их многомерных обобщений (распорок) многомерной сетки. Оценим сложность такого алгоритма. Алгоритм крестовой аппроксимации требует $O(dR_V^3)$ распорок, а извлечение одной распорки из TT-разложения требует $O(dR_V^2)$ операций (что меньше, чем извлечение одного числа). Результирующая сложность алгоритма составляет $O(d^2R_V^5)$ операций. Далее в статье мы будем использовать именно эту оптимизированную версию как базовый алгоритм для сравнения.

5.2. Q-итерация в формате тензорного поезда

В данной статье мы предлагаем модификацию алгоритма TT-VI. Ее суть заключается в следующем: использовать стандарную итерацию функции ценности (алгоритм 1), но применять оператор оптимальности Беллмана в два этапа:

$$Q^k(s,a) = R(s,a) + \gamma \sum_{s'} T(s,a,s')V^k(s'), \tag{9}$$

$$V^{k+1} = \max_a Q^k(s,a). \tag{10}$$

Первый этап (9) включает в себя только стандартные тензорные операции, такие как сложения и суммирования по индексам, тогда как второй (10) требует взятия нелинейной функции (максимума), для чего мы используем крестовый метод, также как в TT-VI алгоритме. Так как у двух алгоритмов совпадают вторые этапы, мы опустим оценку сложности для них в сравнительном анализе.

Рассмотрим первый этап применения оператора $\hat{\beta}$, в частности суммирование по индексам с тензором T. Как следует из используемой нами дискретизации (5), из каждого состояния s возможны только переходы в соседние состояния вдоль каждой оси $s' = s \pm e_i h_i$. Это означает, что сумма вероятностей по s' состоит из $2d_s + 1$ слагаемых: два на каждую координатную ось (вероятность перехода в направлении увеличения или уменьшения на 1) и еще одно на вероятность остаться на месте s:

$$\sum_{s'} T(s,a,s')V(s') = P_0(s,a)V(s) + \sum_{\text{sign}\in\{+,-\}} \sum_{i=1}^{d_s} P_i^{\text{sign}}(s,a)V_i^{-\text{sign}}(s),$$

где P_i^{sign} и P_0 — вероятности, посчитанные с помощью схемы дискретизации (5), а $V_i^{-\text{sign}}(s) = \text{circshift}(V(s_1, \ldots, s_{d_s}), -\text{sign}, i)$ — функция ценности (на сетке), циклически сдвинутая вдоль i-й оси на 1 узел, в направлении $-\text{sign}$.

Предложение 1. *Циклический сдвиг по k-й координате функции на сетке, представленной в виде тензорного поезда, достигается циклическим сдвигом всего одного (k-го) тензорного ядра $G^{(k)}(m_k, i_k, m_{k+1})$. Он стоит $O(NR^2)$ операций и не меняет тензорный ранг.*

Доказательство. Из формулы TT-разложения видно, что при циклическом сдвиге любого из свободных индексов $i_k \to i_k \pm 1 \bmod N_k$ равенство сохраняется:

$$V(i_1, \ldots, i_k \pm 1, \ldots, i_d) = \sum_{m_1,\ldots,m_{d-1}}^{r_1,\ldots,r_{d-1}} G^{(1)}(i_1, m_1) \cdot \ldots \cdot G^{(k)}(m_k, i_k \pm 1 \bmod N_k, m_k) \cdot \ldots \cdot G^{(d)}(m_{d-1}, i_d).$$

Замена $G^{(k)}(m_2, i_k, m_2) \to G^{(k)}(m_2, i_k \pm 1 \bmod N_k, m_2)$ требует перемещения всех элементов массива $G^{(k)}$, которых насчитывается $r_k N_k r_{k+1}$ для всех ядер кроме первого и последнего. Для первого и последнего ядра потребуется $N_1 r_1$ и $N_d r_{d-1}$ элементов соответственно, что в общем случае оценивается как $O(NR^2)$. Очевидно, что такая операция не меняет размер ядра $G^{(k)}$, поэтому ранг r_k сохраняется.

Чтобы вычислить поэлементные произведения $P(s,a)V(s)$ в формате тензорного поезда, необходимо добавить в функцию ценности “лишние” измерения, добавив новые (единичные) ядра: $P(s_1 \ldots s_{d_s}, a_1 \ldots a_{d_a}) \circ ((V(s)) \otimes 1_{a_1} \ldots \otimes 1_{a_{d_a}})$.

Предложение 2. *Сложность первой половины* (9) *применения оператора $\hat{\beta}$ в TT-QI составляет*

$$O(d^2 N R_V^3 R_P^3).$$

Доказательство. Одно применение оператора оптимальности Беллмана в алгоритме TT-QI (до этапа применения алгоритма крестовой аппроксимации) использует следующие тензорные операции:

- $2d_s$ циклических сдвигов $V(s)$, каждый по $O(NR_V^2)$, что в итоге составляет $O(dNR_V^2)$,
- 1 акт добавления в $V(s)$ единичных ядер с лишними измерениями, что составляет $O(d_a)$,
- взятие $2d_s + 1$ поэлементных произведений от тензорных поездов, сложностью $O(dNR_P^2 R_V^2)$ каждый, что дает $O(d^2 NR_P^2 R_V^2)$ в сумме,
- взятие $2d_s + 1$ TT-SVD округлений, что дает $O(dNR_P^3 R_V^3)$ каждый, и $O(d^2 NR_P^3 R_V^3)$ в сумме.

Полная сложность первого этапа составляет в итоге

$$O(dNR_V^2 + d_a + d^2 NR_P^2 R_V^2 + d^2 NR_P^3 R_V^3) = O(d^2 NR_P^3 R_V^3). \tag{11}$$

В случае существенно малых рангов R_P тензоров вероятностей эта асимптотика более выгодна относительно $O(d^2 R_V^5)$ в алгоритме TT-VI. Тензоры вероятностей $P_{0\ldots d_s}(s,a)$ вычисляются только один раз и их ранги R_P постоянны во время итерационного процесса, что в итоге дает асимптотическую сложность $O(R_V^3)$ в случае TT-QI вместо $O(R_V^5)$ в случае TT-VI. В наших численных экспериментах это привело к существенному (в 5–6 раз) выигрышу в производительности.

Таблица 1. Сравнение производительности

Параметр	TT-VI (оптимизизированный) : время, с	TT-QI : время, с
3-маятник, квадратичная награда, целевая ошибка $\delta = 10^{-4}$	1959.9	536.7
3-маятник, задача оптимального быстродействия, целевая ошибка $\delta = 5 \times 10^{-4}$	6024.2	2116.8
3-машина, квадратичная награда, целевая ошибка $\delta = 2 \times 10^{-3}$	2952.2	293.4
4-машина, квадратичная награда, целевая ошибка $\delta = 5 \times 10^{-3}$	3061.8	1109.3

6. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

Чтобы показать верность оптимальных политик (регуляторов), полученных на базе уравнения Беллмана, следует рассмотреть достаточно сложную систему, для которой решение не может быть получено более простыми методами из теории управления (такие как PID или линейно-квадратичные регуляторы).

Данный раздел содержит графики сходимости к решению уравнения Беллмана, а также результаты прямых симуляций траекторий, полученных под управлением оптимальной политики. Для интегрирования стохастических дифференциальных уравнений по Ито и расчета траекторий мы используем схему высокого порядка точности (1.0), предложенную Росслером (см. [13]).

6.1. Система 1: непериодический обратный маятник

Рассмотрим модифицированную задачу обратного маятника, где требуется поставить маятник в верхнее положение с нулевой угловой скоростью:

$$s = [\phi, \dot{\phi}]^{\mathrm{T}}, \tag{12}$$

$$b(s,a) = \left[\dot{\phi}, a - \sin(\phi)\right]^{\mathrm{T}}, \tag{13}$$

$$\sigma(s) = \mathrm{diag}\left([10^{-2}, 10^{-2}]\right). \tag{14}$$

Первая модификация – существенно малая мощность управляющего момента на валу маятника (максимальный момент сил на вал, вызванный управляющим сигналом, равен лишь 30% от максимального момента сил тяжести). Такая постановка задачи делает невозможным для регулятора достижение верхней точки любым способом, кроме как с помощью эксплуатации резонанса системы. Вторая модификация – это использование непериодического маятника, что также делает задачу управления сложнее.

Границы областей состояний и действий определены как

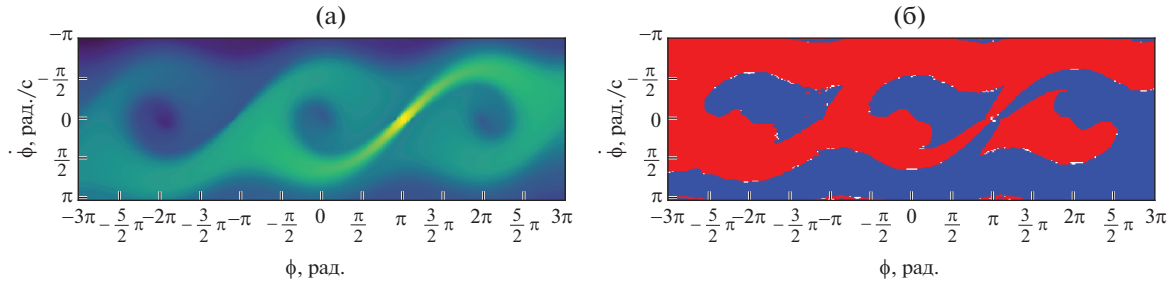
$$s \in [-3\pi, 3\pi] \times [-\pi, \pi], \tag{15}$$

$$a \in [-0.3, 0.3]. \tag{16}$$

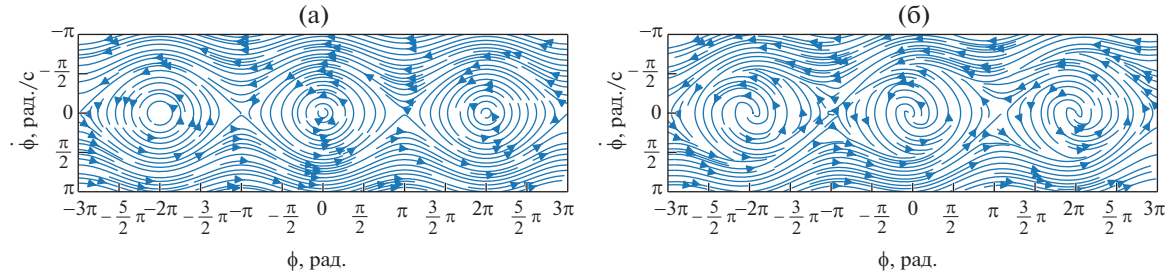
Мы рассмотрим решения двух вариантов данной задачи: с квадратичной наградой и задачу оптимального быстродействия.

6.1.1. Задача управления с квадратичной наградой. Задачи управления с квадратичной наградой/ценой хорошо изучены в литературе, поэтому в данной статье мы будем использовать такую постановку как тестовую для сравнения производительности (см. табл. 1).

Квадратичная награда задается следующим образом: $R_{QR}(s,a) = -(\phi - \pi)^2 - 0.8(\phi)^2 - 0.01a^2$ и коэффициент дисконтирования $\gamma = 0.999$.



Фиг. 1. Решение для задачи оптимального быстродействия непериодическим маятником: (а) – оптимальная функция ценности $V(s)$, (б) – оптимальный регулятор $\pi^*(s)$.



Фиг. 2. Фазовые портреты динамической системы непериодического маятника: (а) – без управления ($a \equiv 0$), (б) – под управлением оптимального регулятора ($a = \pi^*(s)$).

6.1.2. Задача оптимального быстродействия. Для того чтобы сформулировать задачу оптимального быстродействия на беллмановском языке, необходимо положить награду для каждого перехода ($s \xrightarrow{a} as'$) равной отрезку времени, которое требуется, чтобы этот переход совершить:

$$R(s,a,s') = \begin{cases} -\Delta t(s,a,s'), & \text{если} \quad s \notin \mathcal{S}_{\text{terminal}}, \\ 0, & \text{если} \quad s \in \mathcal{S}_{\text{terminal}}. \end{cases} \tag{17}$$

В таком случае функция ценности становится равна матожиданию времени прибывания в терминальную область:

$$V(s) = \sum_{t=0}^{\infty} \mathbb{E} R(s,a,s') = \sum_{t=0}^{\infty} \mathbb{E}\Delta t(s,a,s') = -\sum_{t=0}^{\tau} \mathbb{E}\Delta t = -\tau. \tag{18}$$

Терминальную область в данном случае мы полагаем малой окрестностью целевого состояния $(\pi, 0)$, что соответствует перевернутому маятнику с нулевой угловой скоростью:

$$\mathcal{S}_{\text{terminal}} = [\pi - 2h_0, \pi + 2h_0] \times [-2h_1, 2h_1]. \tag{19}$$

Коэффициент дисконтирования положен равным единице ($\gamma = 1$), т.е. решается недисконтированная задача.

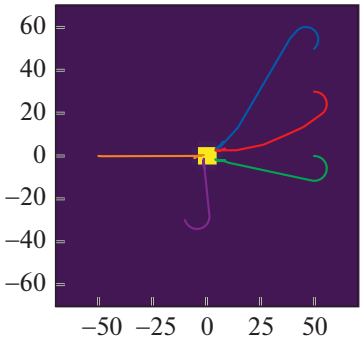
Решение данной задачи на сетке с дискретизацией $301 \times 151 \times 51$ дает оптимальную функцию ценности, оптимальный регулятор (см. фиг. 1) и соответствующим образом изменяет фазовый портрет динамической системы под действием регулятора, как показано на фиг. 2.

6.2. Система 2: машина Дубинса

Другим известным примером малоприводной системы является машина (автомобиль) Дубинса.

6.2.1. Простая машина Дубинса. Рассмотрим простейшую модель колесного автомобиля, описываемого уравнениями

$$s = [x, y, \phi]^{\mathrm{T}}, \tag{20}$$



Фиг. 3. Симулированные траектории (x, y) машины Дубинса с инерцией.

$$a = [u, \theta]^{\mathrm{T}}, \tag{21}$$

$$b(s,a) = \left[u\cos\phi,\ u\sin\phi,\ \frac{u}{L}\operatorname{tg}\theta\right]^{\mathrm{T}}, \tag{22}$$

$$\sigma(s) = \operatorname{diag}(10^{-3}, 10^{-3}, 10^{-3}). \tag{23}$$

Для усложнения маневра положим, что данная машина может ехать только вперед. Пространства состояний и действий тогда имеют следующий вид:

$$\mathcal{S} = [-100, 100]\times[-100, 100]\times SO(2),$$
$$\mathcal{A} = [0, 1]\times\left[-\frac{\pi}{3}, -\frac{\pi}{3}\right],$$

где $SO(2)$ – одномерная окружность. Функция награды задана следующим образом:

$$r(s,a) = -10^{-4}x^2 - 10^{-4}y^2 - 10^{-5}\phi^2 - 10^{-3}u^2.$$

Результаты тестов сходимости показаны ниже на фиг. 6а.

6.2.2. Машина Дубинса с инерцией. Данная модель машины является усложненной версией модели из п. 6.2.1. В ней сигнал управления не влияет на скорость напрямую, а лишь управляет ускорением. В этой модели автомобиль так же может ехать только вперед, а ускоряться может и вперед, и назад:

$$s = [x, y, v, \phi]^{\mathrm{T}}, \tag{24}$$

$$a = [u, \theta]^{\mathrm{T}}, \tag{25}$$

$$b(s,a) = \left[v\cos\phi,\ v\sin\phi,\ u,\ \frac{u}{L}\operatorname{tg}\theta\right]^{\mathrm{T}}, \tag{26}$$

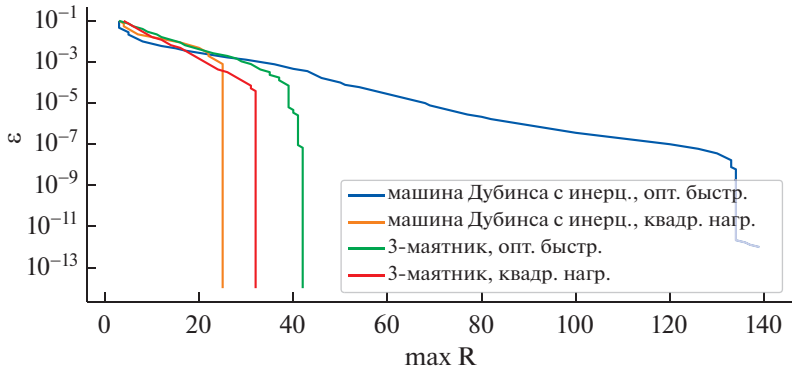
$$\sigma(s) = \operatorname{diag}(10^{-3}, 10^{-3}, 10^{-3}, 10^{-3}), \tag{27}$$
$$\mathcal{S} = [-70, 70]\times[-70, 70]\times SO(2),$$
$$\mathcal{A} = [-2, 1]\times\left[-\frac{\pi}{3}, -\frac{\pi}{3}\right].$$

Мы рассмотрим два варианта награды для этой системы: (A) – для задачи оптимального быстродействия ($r = r_A$, $\gamma = 1$), (B) – для задачи с квадратичной наградой (r_B, $\gamma = 0.999$):

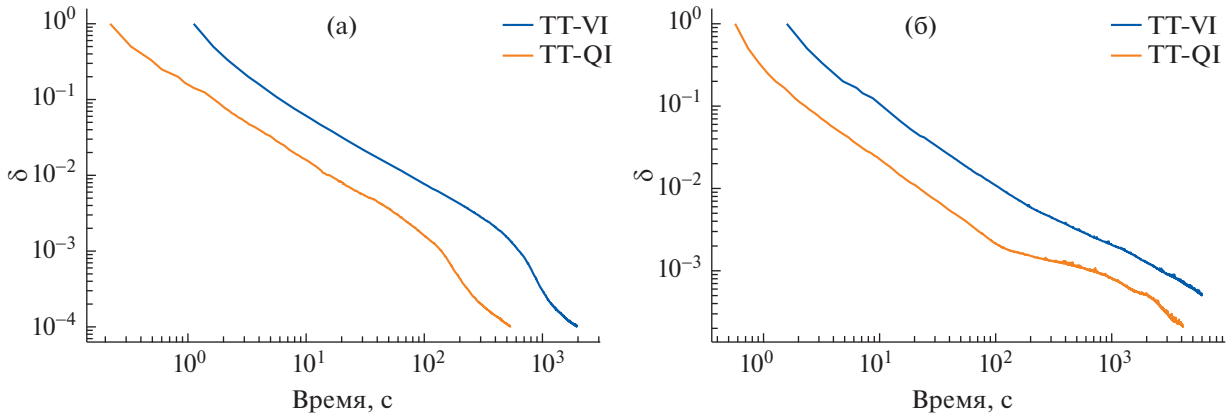
$$r_A(s,a) = -\Delta t(s,a), \tag{28}$$

$$r_B(s,a) = -10^{-4}x^2 - 10^{-4}y^2 - 10^{-5}\phi^2 - 10^{-3}u^2. \tag{29}$$

На фиг. 3 видно несколько симулированных траекторий координат (x, y) для задачи оптимального быстродействия машины Дубинса с инерцией (вариант A). Все траектории прибыли в



Фиг. 4. Сжимаемость функции ценности в разных задачах.



Фиг. 5. График относительной ошибки решения уравнения Беллмана для алгоритмов TT-VI и TT-QI: (а) – 3-маятник (квадратичная награда), (б) – 3-маятник (задача оптимального быстродействия).

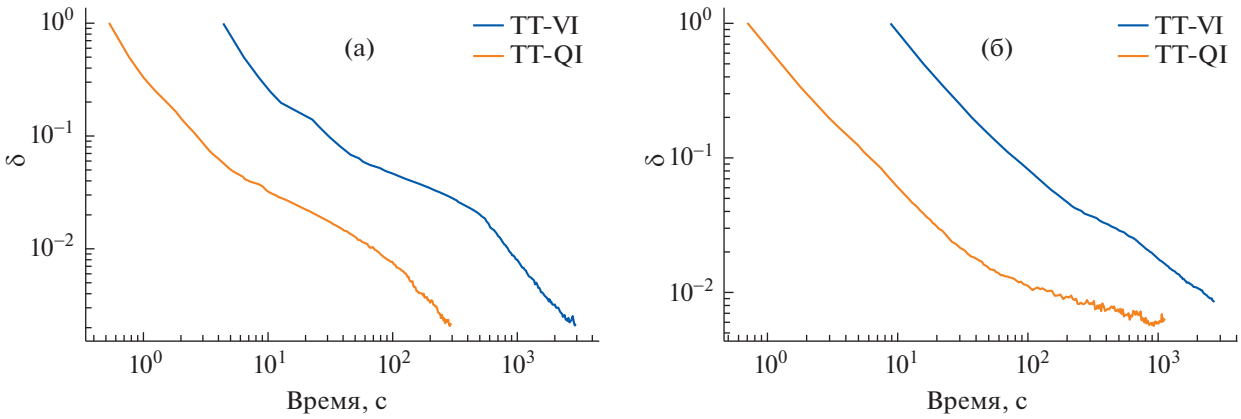
терминальную область. Также видно, что траектории близко соответствуют теоретически предсказаннной форме кривых (путям Дубинса).

7. СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ

В данном разделе мы сравниваем производительность нашего алгоритма (TT-QI) с оптимизированной версией алгоритма TT-VI. На фиг. 4 изображена зависимость максимального ранга функций ценности R_V от целевой ошибки сжатия ε с помощью TT-SVD. Видно, что функции ценности для данных задач действительно с высокой точностью являются малоранговыми.

Для экспериментов, приведенных ниже, функция ценности инициализировалась нулевыми малоранговыми тензорами. Вычисления проводились на рабочей станции с 3.5 ГГц 8-ядерным процессором производства AMD и 12 Гб оперативной памяти. На фиг. 5 и 6 видно, что оба алгоритма ведут себя практически идентично и имеют степенной закон сходимости, но отличаются по затраченному времени на константный множитель.

Важно отметить, что графики сходимости обоих алгоритмов от числа итераций ведут себя идентично, так как аппроксимируют один и тот же оператор оптимальности Беллмана с достаточно высокой точностью. А скорость сходимости итерации функции ценности зависит только от размера пространства действий и скорости перемешивания в марковской цепи, которые совпадают.



Фиг. 6. График относительной ошибки решения уравнения Беллмана для алгоритмов TT-VI и TT-QI: (а) – простая машина Дубинса (квадратичная награда), (б) – машина Дубинса с инерцией (квадратичная награда).

На фиг. 5 и 6 построена относительная ошибка δ, которая для k-й итерации определяется как

$$\delta_k = \frac{\|V_k - V_{k-1}\|_2}{\|V_{k-1}\|_2}. \tag{30}$$

Сравнение времени решения с относительной ошибкой δ показано в табл. 1.

8. ЗАКЛЮЧЕНИЕ

Мы предложили модифицированный алгоритм для итерации функции ценности для задач стохастического оптимального управления в формате малорангового тензорного поезда. Если TT-ранги функции вероятностей перехода (после применения схемы расщепления) малы, а также TT-ранги функции награды малы, наш алгоритм имеет меньшую вычислительную сложность, чем существующий метод решения стохастического оптимального управления для общего случая неаффинных систем, существующий в литературе (см. [7]).

В численных экспериментах на примере классических нелинейных малоприводных задач управления (непериодический обратный маятник, машины Дубинса) для задачи с квадратичной наградой и задачи оптимального быстродействия наш метод позволил сократить время нахождения точного решения вплоть до 10 раз.

Так как данный подход позволяет решать задачи управления для систем достаточно большой размерности и общего вида, данный результат может быть важен для синтеза сложных движений и маневров в различных сферах робототехники.

Авторы благодарны С. Долгову (университет Бата), С. Матвееву (ИВМ РАН, Сколтех) и Г. Овчинникову (Сколтех) за ценные обсуждения.

СПИСОК ЛИТЕРАТУРЫ

1. *Kushner H., Dupuis P.G.* Numerical methods for stochastic control problems in continuous time. Springer, 2013, V. 24.
2. *Fleming W.H., Soner H.M.* Controlled Markov Processes and Viscosity Solutions. Springer, 2006.
3. *Bertsekas D.P., Tsitsiklis J.N.* Neuro-Dynamic Programming, 1st ed. Athena Scientific, 1996.
4. *Lillicrap T.P. et al.* Continuous control with deep reinforcement learning // 4th Inter. Conf. Learn. Represent. ICLR, 2016.
5. *Kidzinski et al.* Learning to run challenge solutions: Adapting reinforcement learning methods for neuromusculoskeletal environments // Proceed. NIPS, 2017.
6. *Horowitz M., Damle A., Burdick J.W.* Linear Hamilton-Jacobi-Bellman equations in high dimensions // IEEE Conf. Decis. Control, 2014.
7. *Gorodetsky A.A., Karaman S., Marzouk Y.M.* Efficient high-dimensional stochastic optimal motion control using Tensor Train decomposition // Robotics: Sci. Syst. 2015.

8. *Gorodetsky A.A., Karaman S., Marzouk Y.M.* High-dimensional stochastic optimal control using continuous tensor decompositions // Inter. J. Robot. Res. 2018. № 37. Iss. 2–3.
9. *Tal E., Gorodetsky A., Karaman A.* Continuous Tensor Train-based dynamic programming for high-dimensional zero-sum differential games // Am. Control Conf. 2018.
10. *Oseledets I.V., Tyrtyshnikov E.E.* Breaking the curse of dimensionality, or how to use SVD in many dimensions // SIAM J. Sci. Comp. 2009. V. 31. № 5. P. 3744–3759.
11. *Oseledets I.V., Tyrtyshnikov E.E.* TT-cross approximation for multidimensional arrays // Lin. Alg. Appl. 2010. V. 432. № 1. P. 70–88.
12. *Bellman R.E.* Dynamic programming. Princeton Univ. Press, 1957.
13. *Rossler A.* Runge–Kutta methods for the strong approximation of solutions of stochastic differential equations // SIAM J. Numer. Anal. 2010. V. 3. № 48. P. 922–952.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 878–884

УРАВНЕНИЯ В ЧАСТНЫХ ПРОИЗВОДНЫХ

УДК 517.63

ЧИСЛЕННЫЙ МЕТОД РЕШЕНИЯ ОБЪЕМНЫХ ИНТЕГРАЛЬНЫХ УРАВНЕНИЙ НА НЕРАВНОМЕРНОЙ СЕТКЕ[1)]

© 2021 г. А. Б. Самохин[1],*, Е. Е. Тыртышников[2],**

[1] 119454 Москва, пр. Вернадского, 78, МИРЭА – Российский технологический университет, Россия

[2] 119333 Москва, ул. Губкина, 8, Институт вычислительной математики им. Г.И. Марчука РАН, Россия

*e-mail: absamokhin@yandex.ru

**e-mail: eugene.tyrtyshnikov@gmail.com

Поступила в редакцию 24.12.2020 г.
Переработанный вариант 24.12.2020 г.
Принята к публикации 14.01.2021 г.

Рассматриваются численные методы решения объемных интегральных уравнений, описывающих задачи рассеяния волн на прозрачных препятствиях. Для аппроксимации уравнений применяется метод коллокации на неравномерной сетке и задача сводится к решению системы линейных алгебраических уравнений. Предлагается эффективный метод приближенного умножения матрицы этой системы на вектор, сравнимый по сложности с методом, который применяется в случае равномерной сетки. При построении метода вводится вспомогательная равномерная сетка, используются методы интерполяции функций и алгоритмы быстрого дискретного преобразования Фурье. Существенно то, что число узлов вспомогательной равномерной сетки сопоставимо с числом узлов исходной неравномерной сетки. Библ. 5.

Ключевые слова: объемные интегральные уравнения, метод коллокации, неравномерная сетка, методы интерполяции функций, эффективные алгоритмы.

DOI: 10.31857/S0044466921050161

1. ВВЕДЕНИЕ

Задачи рассеяния волн различной физической природы на неоднородных структурах, находящихся в трехмерной ограниченной области Q, имеют большое значение как с теоретической, так и с практической точек зрения. Указанные задачи могут быть описаны объемным интегральным уравнением следующего общего вида:

$$(1 + \alpha\eta(x))u(x) + \int_Q \frac{K(x-y)}{4\pi R^d}\eta(y)u(y)dy = u_0(x), \quad x \in Q, \quad d \le 3, \tag{1}$$

где $R = |x - y|$ – расстояние между точками $x = (x_1, x_2, x_3)$ и $y = (y_1, y_2, y_3)$, α, η, K, u_0 – известные функции, причем $K(x - y)$ – дифференцируемая функция координат, u – неизвестная функция. Приведем некоторые важные классы задач, сводящихся к уравнению (1).

• Рассеяние акустических волн на прозрачном неоднородном препятствии (см. [1]). В этом случае $d = 1$, $\alpha = 0$, а остальные функции, входящие в уравнение (1), являются скалярными.

• Рассеяние электромагнитных волн на неоднородном, в общем случае анизотропном, теле (см. [2]). В этом случае $d = 3$ и поэтому оператор в (1) будет сингулярным, функции u и u_0 – векторными, а η и K – тензорными. Величина α является внеинтегральным членом сингулярного оператора и зависит от формы выделяемой особенности и ее центра. Например, если выделяемая особеность есть шар, то $\alpha = 1/3$.

Для решения уравнения (1), которое описывает реальные задачи, возможно использование только численных методов. В методе Галеркина или в методе коллокации уравнение (1) аппрок-

[1)]Работа выполнена при поддержке Московского центра фундаментальной и прикладной математики (соглашение 075-15-2019-1624 с Минобрнауки РФ).

симируется системой линейных алгебраических уравнений. Для трехмерных задач размер N получаемых систем достаточно велик.

Основными критериями эффективности численного алгоритма являются число арифметических операций T и объем памяти M, необходимой для его реализации. При использовании прямых методов $T \sim N^3$ и $M \sim N^2$. Для итерационных методов

$$T \sim LT_A, \quad M \sim M_A, \tag{2}$$

где T_A – число арифметических операций, которое требуется для умножения матрицы на вектор, L – количество итераций, необходимое для получения решения с заданной точностью, а M_A – число параметров, определяющих матрицу. Используя метод коллокации на равномерной сетке и свойства ядер интегральных операторов, уравнение (1) можно аппроксимировать системой линейных уравнений с многоуровневой блочно тёплицевой матрицей и строить эффективные алгоритмы матрично-векторного умножения, в которых значения T_A и M_A практически пропорциональны размеру N (см. [3]).

Однако во многих случаях целесообразно использование неравномерной сетки. Такая потребность возникает при аппроксимации сложной границы области Q или при значительном изменении параметров среды в области неоднородности. К сожалению, при применении метода коллокации на неравномерной сетке матрица получаемой системы утрачивает свойство тёплицевости и поэтому непосредственное использование вышеуказанной техники невозможно.

В настоящей работе для аппроксимации интегрального уравнения (1) применяется метод коллокации на неравномерной сетке, а затем используется некоторая равномерная сетка, методы интерполяции функций и алгоритмы быстрого преобразования Фурье. Подчеркнем, что число узлов равномерной сетки оказывается сопоставимым с числом узлов исходной неравномерной сетки. Поэтому в итоге строятся эффективные алгоритмы численного решения уравнения (1) итерационными методами, в которых значения T_A и M_A практически пропорциональны размеру N, хотя коэффициент пропорциональности несколько больше по сравнению с алгоритмами на равномерной сетке

2. ПОСТАНОВКА ЗАДАЧИ

Для аппроксимации интегрального уравнения (1) будем использовать метод коллокации на неравномерной сетке. Представим область Q в виде объединения N_Q ячеек Ω_n, $n = 1, 2, \ldots, N_Q$. Узловые точки в этих ячейках будем выбирать в их центрах, которые определяются формулами

$$x_i^c = \frac{\int\limits_{\Omega} x_i dx_1 dx_2 dx_3}{|\Omega|}, \quad i = 1, 2, 3, \tag{3}$$

где $x^c = (x_1^c, x_2^c, x_3^c)$ – центр ячейки Ω, а $|\Omega|$ – ее объем. Если в качестве ячеек выбираются тетраэдры произвольной формы, то можно достаточно точно описать многие сложные конфигурации области Q и среды. Центр тетраэдра определяется простой формулой

$$x_i^c = \frac{x_i^{(1)} + x_i^{(2)} + x_i^{(3)} + x_i^{(4)}}{4}, \quad i = 1, 2, 3, \tag{4}$$

где $x_1^{(k)}$, $x_2^{(k)}$, $x_3^{(k)}$ – декартовы координаты k-й вершины тетраэдра.

Будем аппроксимировать интегральное уравнение (1) системой линейных алгебраических уравнений размера $\sim N_Q$ относительно значений неизвестного поля в узловых точках области Q, находящихся в центрах ячеек Ω_n:

$$(1 + \alpha_n \eta_n) u_n + \sum_{m=1}^{N_Q} A(n, m) \eta_m u_m = u_n^0, \quad n = 1, 2, \ldots, N_Q,$$

$$A(n, m) = \int\limits_{\Omega_m} \frac{K(x^{cn} - y)}{4\pi |x^{cn} - y|^d} dy, \quad n \neq m, \quad A(n, n) = 0, \tag{5}$$

$$u_n = u(x^{cn}), \quad u_n^0 = u_0(x^{cn}), \quad \eta_n = \eta(x^{cn}).$$

Для векторных задач тензор α_n определяется формой ячейки Ω_n и ее центром (см. [4]). Отметим, что, поскольку узловые точки находятся в центре ячеек, точность аппроксимации интегральных операторов $\sim h^2$, где h – максимальный диаметр ячеек.

Для относительно небольших значений $N_Q \sim 10\,000$ можно решить систему уравнений (5) прямыми или итерационными методами. Ниже мы предлагаем эффективные алгоритмы, которые с использованием итерационных методов позволяют решать систему (5) со значительно большим размером N_Q.

3. ПОСТРОЕНИЕ АЛГОРИТМА РЕШЕНИЯ

Сначала рассмотрим следующий алгоритм интерполяции функций. Пусть известны значения дифференцируемой функции $f(x_1, x_2, x_3)$ в вершинах параллелепипеда. Тогда приближенное значение функции в любой точке параллелепипеда может быть представлено формулой

$$\begin{aligned} f(x_1, x_2, x_3) \approx {} & \frac{(a_1 + h_1 - x_1)(b_1 + h_2 - x_2)(c_1 + h_3 - x_3)}{h_1 h_2 h_3} f(a_1, b_1, c_1) + \\ & + \frac{(x_1 - a_1)(b_1 + h_2 - x_2)(c_1 + h_3 - x_3)}{h_1 h_2 h_3} f(a_1 + h_1, b_1, c_1) + \\ & + \frac{(x_1 - a_1)(x_2 - b_1)(c_1 + h_3 - x_3)}{h_1 h_2 h_3} f(a_1 + h_1, b_1 + h_2, c_1) + \\ & + \frac{(a_1 + h_1 - x_1)(x_2 - b_1)(c_1 + h_3 - x_3)}{h_1 h_2 h_3} f(a_1, b_1 + h_2, c_1) + \\ & + \frac{(a_1 + h_1 - x_1)(b_1 + h_2 - x_2)(x_3 - c_1)}{h_1 h_2 h_3} f(a_1, b_1, c_1 + h_3) + \\ & + \frac{(x_1 - a_1)(b_1 + h_2 - x_2)(x_3 - c_1)}{h_1 h_2 h_3} f(a_1 + h_1, b_1, c_1 + h_3) + \\ & + \frac{(x_1 - a_1)(x_2 - b_1)(x_3 - c_1)}{h_1 h_2 h_3} f(a_1 + h_1, b_1 + h_2, c_1 + h_3) + \\ & + \frac{(a_1 + h_1 - x_1)(x_2 - b_1)(x_3 - c_1)}{h_1 h_2 h_3} f(a_1, b_1 + h_2, c_1 + h_3). \end{aligned} \tag{6}$$

В формуле (6) a_1, b_1, c_1 – декартовы координаты левой нижней вершины параллелепипеда, а h_1, h_2, h_3 – длины ребер по осям x_1, x_2, x_3. Формула (6) дает точное значение в любой точке параллелепипеда, если функция имеет вид

$$f(x_1, x_2, x_3) = d_0 + d_1 x_1 + d_2 x_2 + d_3 x_3 + d_{12} x_1 x_2 + d_{13} x_1 x_3 + d_{23} x_2 x_3 + d_{123} x_1 x_2 x_2.$$

Точность интерполяции по формуле (6) имеет второй порядок по $h = \max(h_1, h_2, h_3)$. Запишем (6) в виде

$$f(x) \approx \sum_{t=1}^{8} \beta(t, x) f^t, \tag{7}$$

где f^t – значение функции $f(x)$ в t-й вершине параллелепипеда в нумерации, соответствующей (6), а значения $\beta(t, x)$ также определяются формулой (6).

В прямоугольной декартовой системе координат введем сетку так, чтобы область Q целиком находилась в прямоугольном параллелепипеде Π, стороны которого имеют длины $N_i h_i$, $i = 1, 2, 3$, где h_i – шаги сетки по декартовым координатам. Параллелепипед Π разбивается данной сеткой на элементарные параллелепипеды $\Pi(p)$, $p = (p_1, p_2, p_3)$, $p_i = 0, 1, \ldots, N_i - 1$. Количество элемен-

тарных параллелепипедов в Π равно $N = N_1N_2N_3$. Центр параллелепипеда $\Pi(p)$ в единицах шагов сетки обозначим через

$$p^c(p) = (p_1 + 1/2, p_2 + 1/2, p_3 + 1/2). \tag{8}$$

Определим узловые точки в параллелепипеде Π формулами

$$x = (x_1, x_2, x_3), \quad x_i = \tilde{p}_i h_i, \quad \tilde{p}_i = 0, 1, \ldots, N_i, \quad i = 1, 2, 3. \tag{9}$$

Узловую точку сетки будем обозначать $\tilde{p} = (\tilde{p}_1, \tilde{p}_2, \tilde{p}_3)$. Понятно, что вершины элементарного параллелепипеда $\Pi(p)$ задаются восемью узловыми точками. Общее число узловых точек в параллелепипеде Π равно $(N_1 + 1)(N_2 + 1)(N_3 + 1)$. Оно несколько больше, чем число элементарных параллелепипедов, но при больших значениях N_1, N_2, N_3 асимптотически такое же. Отметим, что узловая точка может быть вершиной от одного (вершины исходного параллелепипеда Π) до восьми (внутренние узловые точки) элементарных параллелепипедов. Запись $\tilde{p} \in \Pi(p)$ означает, что узловая точка $\tilde{p}$ является вершиной элементарного параллелепипеда $\Pi(p)$. Условие принадлежности можно сформулировать в виде равенства

$$|p^c(p) - \tilde{p}|^2 = 3/4, \quad \Pi(p) \subset \Pi. \tag{10}$$

Предположим, что на параллелепипеде Π определена функция вида $F(x - y)$, которая зависит только от разности декартовых координат и является дифференцируемой функцией точек x и y при $x \in \Pi(p)$, $y \in \Pi(q)$, $p \neq q$. Рассмотрим сумму вида

$$W(x) = \sum_{l=1}^{k} a_l F(x - y_l), \quad x \in \Pi(p), \quad y_l \in \Pi(q), \tag{11}$$

где $k \geq 1$, а величины a_l постоянны. Применим формулы интерполяции (6),(7) к функции $F(x - y)$ в точках x и y_l в параллелепипедах $\Pi(p)$ и $\Pi(q)$ соответственно. Получим

$$W(x) \approx \sum_{l=1}^{k} a_l \sum_{\tilde{p} \in \Pi(p)} \beta(\tilde{p}, p, x) F(x(\tilde{p}) - y_l) \approx \sum_{l=1}^{k} a_l \sum_{\tilde{p} \in \Pi(p)} \beta(\tilde{p}, p, x) \sum_{\tilde{q} \in \Pi(q)} \beta(\tilde{q}, q, y_l) F(x(\tilde{p}) - y(\tilde{q})).$$

Окончательно имеем

$$W(x) \approx \sum_{\tilde{p} \in \Pi(p)} \beta(\tilde{p}, p, x) \sum_{\tilde{q} \in \Pi(q)} \nu(\tilde{q}, q) F(x(\tilde{p}) - y(\tilde{q})), \tag{12}$$

$$\nu(\tilde{q}, q) = \sum_{l=1}^{k} a_l \beta(\tilde{q}, q, y_l), \quad y_l \in \Pi(q). \tag{13}$$

В (12), (13) коэффициенты $\beta(\tilde{q}, q, y)$, $y \in \Pi(q)$, и $\beta(\tilde{p}, p, x)$, $x \in \Pi(p)$, определяются согласно формулам (6), (7). Формулы (12), (13) имеют второй порядок точности по h. Они будут использоваться для вычисления выражений вида (11), когда входящие в них точки y_l находятся внутри нескольких элементарных параллелепипедов. В этом случае суммирование во второй сумме (12) проводится по всем вершинам элементарных параллелепипедов, содержащих точки y_l. При этом веса интерполяции суммируются в (13) в общих вершинах элементарных параллелепипедов.

Запишем уравнение (5) в виде, удобном для построения алгоритма. Будем полагать, что каждая ячейка $\Omega_n \subset Q$ находится внутри только одного из элементарных параллелепипедов, причем внутри него может быть несколько ячеек. Данное требование практически не накладывает ограничений на выбор ячеек: если ячейка принадлежит нескольким элементарным параллелепипедам, то ее можно разбить на несколько ячеек. Обозначим через $k(p)$ число ячеек в элементарном параллелепипеде $\Pi(p)$, через $x(l, p)$ центр l-й ячейки в $\Pi(p)$, а через $\eta(l, p)$ и $\alpha(l, p)$ значения η_n и α_n для соответствующей ячейки. Обозначим через Π_Q объединение всех элементарных паралле-

лепипедов, внутри которых находятся ячейки, а через $\Omega(l, p)$ обозначим l-ю ячейку в параллелепипеде $\Pi(p)$. Тогда систему линейных уравнений (5) можно переписать в следующем виде:

$$\gamma(l, p)u(l, p) + \sum_{q}^{\Pi(q)\subset\Pi_Q}\sum_{m=1}^{k(q)} A(l, p, m, q)\eta(m, q)u(m, q) = u_0(l, p),$$
$$f(l, p) \equiv f(x(l, p)), \quad \gamma(l, p) = 1 + \alpha(l, p)\eta(l, p), \quad \Pi(p) \subset \Pi_Q, \quad l = 1, 2, \ldots, k(p), \tag{14}$$
$$A(l, p, m, q) = \int_{\Omega(m,q)} \frac{K(x(l, p) - y)}{4\pi|x(l, p) - y|^d}\,dy, \quad \text{при} \quad (l, p) \neq (m, q), \quad A(l, p, l, p) = 0.$$

Основные вычислительные затраты в итерационных алгоритмах решения системы (14) связаны с умножением матрицы на вектор, т.е. с вычислением сумм вида

$$W(l, p) = \sum_{q}^{\Pi(q)\subset\Pi_Q}\sum_{m=1}^{k(q)} A(l, p, m, q)U(m, q). \tag{15}$$

Обозначим через $\Pi_0(p)$ объединение элементарных параллелепипедов в Π_Q, содержащих $\Pi(p)$ и расположенных в некоторой окрестности $\Pi(p)$. Опишем два возможных варианта определения $\Pi_0(p)$:

• область $\Pi_0^1(p)$ содержит параллелепипед $\Pi(p)$ и все элементарные параллелепипеды, соприкасающиеся с ним (максимальное число таких параллелепипедов равно 27);

• область $\Pi_0^2(p)$ содержит параллелепипед $\Pi_0^1(p)$ и все элементарные параллелепипеды, соприкасающиеся с ним (максимальное количество таких параллелепипедов равно 125).

Обозначим $\Pi_Q(p) = \Pi_Q\backslash\Pi_0(p)$. Тогда (15) можно представить в виде

$$W(l, p) = W_1(l, p) + W_2(l, p),$$
$$W_1(l, p) = \sum_{q}^{\Pi(q)\subset\Pi_0(p)}\sum_{m=1}^{k(q)} A(l, p, m, q)U(, m, q), \tag{16}$$
$$W_2(l, p) = \sum_{q}^{\Pi(q)\subset\Pi_Q(p)}\sum_{m=1}^{k(q)} A(l, p, m, q)U(, m, q).$$

Значения $W_1(l, p)$ будем вычислять непосредственно по вышеприведенной формуле. Число арифметических операций и память для этих вычислений пропорциональны N_Q.

Перейдем к построению алгоритма для вычисления значений $W_2(l, p)$. Определим функцию

$$B(x - y) = \frac{K(x - y)}{4\pi|x - y|^d}, \quad x \neq y, \quad B(0) = 0. \tag{17}$$

Для вычисления интегралов (14) по ячейке в выражении для W_2 будем использовать приближенную формулу

$$\int_{\Omega} f(x)dx \approx f(x^c)|\Omega|. \tag{18}$$

Формула (18) дает точное значение интеграла на линейных функциях вида $f(x) = a_0 + a_1x_1 + a_2x_2 + a_3x_3$ и имеет второй порядок по h, где h – диаметр ячейки Ω. Из (14), (16)–(18) находим

$$W_2(l, p) \approx \sum_{q}^{\Pi(q)\subset\Pi_Q(p)}\sum_{m=1}^{k(q)} B(x(l, p) - y(m, q))V(m, q),$$
$$V(m, q) = U(m, q)|\Omega(m, q)|. \tag{19}$$

Отметим, что формулы вида (18) для вычисления интегралов, входящих в W_1, и применение интерполяции могут приводить к значительным погрешностям. Это обстоятельство необходимо учитывать при определении областей $\Pi_0(p)$ и шагов сетки.

В узловых точках параллелепипеда Π определим функцию дискретного аргумента

$$B(\tilde{p}-\tilde{q}) = B(x(\tilde{p}) - y(\tilde{q})), \quad \tilde{p},\tilde{q} \in \Pi. \tag{20}$$

Из свойств симметрии функции следует, что весь массив значений $B(\tilde{p}-\tilde{q})$, $\tilde{p},\tilde{q} \in \Pi$, определяется его частью, содержащей $(N_1+1)(N_2+1)(N_3+1)$ элементов. Функцию $B(\tilde{p}-\tilde{q})$ и формулы (12), (13) будем использовать для интерполяции функций $B(x(l,p)-y(m,q))$, входящих в (19). Положим

$$\tilde{V}(\tilde{p},p) = \sum_{m=1}^{k(p)} \beta(\tilde{p},p,m)V(m,p), \quad \tilde{p} \in \Pi(p), \quad \Pi(p) \subset \Pi_Q, \tag{21}$$

где $\beta(\tilde{p},p,m) = \beta(\tilde{p},p,x(m,p))$. В (21) веса интерполяции значений V в центрах ячеек параллелепипеда $\Pi(p)$ суммируются в узловых точках $\Pi(p)$. Тогда, учитывая (10), W_2 можно приближенно представить в виде

$$W_2(l,p) = \sum_{\tilde{p}\in\Pi(p)} \beta(\tilde{p},p,l)\tilde{W}_2(\tilde{p},p), \quad \Pi(p) \subset \Pi_Q, \tag{22}$$

$$\begin{gathered}\tilde{W}_2(\tilde{p},p) = \sum_{\tilde{q}\in\Pi_Q(p)} B(\tilde{p}-\tilde{q})\tilde{V}_2(\tilde{q},p), \quad \tilde{p} \in \Pi(p), \\ \tilde{V}_2(\tilde{q},p) = \sum_q \tilde{V}(\tilde{q},q), \quad \left|q^c(q)-\tilde{q}\right|^2 = \frac{3}{4}, \quad \Pi(q) \subset \Pi_Q(p), \quad \tilde{q} \in \Pi_Q(p).\end{gathered} \tag{23}$$

Далее, в узловых точках параллелепипеда Π определим функцию

$$\tilde{V}(\tilde{q}) = \begin{cases} \sum\limits_q V(\tilde{q},q), & \left|q^c(q)-\tilde{q}\right|^2 = \frac{3}{4}, \quad \Pi(q) \subset \Pi_Q, \quad \tilde{q} \in \Pi_Q, \\ 0, & \tilde{q} \notin \Pi_Q, \quad \tilde{q} \in \Pi. \end{cases} \tag{24}$$

Рассмотрим сумму

$$\tilde{W}(\tilde{p}) = \sum_{\tilde{q}\in\Pi} B(\tilde{p}-\tilde{q})\tilde{V}(\tilde{q}), \quad \tilde{p} \in \Pi. \tag{25}$$

Сравнивая $\tilde{W}(\tilde{p})$ и $\tilde{W}_2(\tilde{p},p)$, при $\Pi(p) \subset \Pi_Q$ находим

$$\tilde{W}_2(\tilde{p},p) = \tilde{W}(\tilde{p}) - \tilde{W}_1(\tilde{p},p), \quad \tilde{p} \in \Pi(p), \tag{26}$$

$$\begin{gathered}\tilde{W}_1(\tilde{p},p) = \sum_{\tilde{q}\in\Pi_0(p)} B(\tilde{p}-\tilde{q})\tilde{V}_1(\tilde{q},p), \quad \tilde{p} \in \Pi(p), \\ \tilde{V}_1(\tilde{q},p) = \sum_q V(\tilde{q},q), \quad \left|q^c(q)-\tilde{q}\right|^2 = \frac{3}{4}, \quad \Pi(q) \subset \Pi_0(p), \quad \tilde{q} \in \Pi_0(p).\end{gathered} \tag{27}$$

Суммы $\tilde{W}(\tilde{p})$ можно записать в следующем виде:

$$\begin{gathered}\tilde{W}(\tilde{p}_1,\tilde{p}_2,\tilde{p}_3) = \sum_{\tilde{q}=1}^{N_1}\sum_{\tilde{q}_2=1}^{N_2}\sum_{\tilde{q}_3=1}^{N_3} B(\tilde{p}_1-\tilde{q}_1,\tilde{p}_2-\tilde{q}_2,\tilde{p}_3-\tilde{q}_3)\tilde{V}(\tilde{q}_1,\tilde{q}_2,\tilde{q}_3), \\ 0 \le \tilde{p}_1 \le N_1, \quad 0 \le \tilde{p}_2 \le N_2, \quad 0 \le \tilde{p}_3 \le N_3.\end{gathered} \tag{28}$$

Для вычисления этих сумм будем использовать технику умножения тёплицевой матрицы на вектор (см. [3]). Кратко опишем ее применительно к (28). Обозначим через Π_2 параллелепипед со сторонами $2N_1h_1$, $2N_2h_2$, $2N_3h_3$. Продолжим матричную функцию дискретного аргумента $B(\tilde{p}_1,\tilde{p}_2,\tilde{p}_3)$ на все целочисленные значения $\tilde{p}_1$, $\tilde{p}_2$, $\tilde{p}_3$, полагая ее периодической по каждой переменной с периодами соответственно $2(N_1+1)$, $2(N_2+1)$, $2(N_3+1)$. При этом доопределим функцию $B(\tilde{p}_1,\tilde{p}_2,\tilde{p}_3)$ нулем в тех точках, где она не определена. Доопределим вектор-функцию дискретного аргумента $V(\tilde{p}_1,\tilde{p}_2,\tilde{p}_3)$ нулем во всех узловых точках Π_2, не принадлежащих Π. Ясно, что при $\tilde{p} \in \Pi$ функция

$$W(\tilde{p}_1,\tilde{p}_2,\tilde{p}_3) = \sum_{\tilde{q}_1=1}^{2N_1+1}\sum_{\tilde{q}_2=1}^{2N_2+1}\sum_{\tilde{q}_3=1}^{2N_3+1} B(\tilde{p}_1-\tilde{q}_1,\tilde{p}_2-\tilde{q}_2,\tilde{p}_3-\tilde{q}_3)V(\tilde{q}_1,\tilde{q}_2,\tilde{q}_3) \tag{29}$$

совпадает с $W(\tilde{p}_1, \tilde{p}_2, \tilde{p}_3)$ из (28). Выполнив дискретное преобразование Фурье по каждой переменной от обеих частей (29), получим равенство

$$W^F(k_1, k_2, k_3) = B^F(k_1, k_2, k_3)V^F(k_1, k_2, k_3), \quad k \in \Pi_2. \tag{30}$$

Таким образом, функцию $W(\tilde{p}, \tilde{p}) \in \Pi$ можно вычислить, используя прямое и обратное быстрые дискретные преобразования Фурье.

Отметим, что точности интерполяции функций и аппроксимации интегралов имеют один и тот же порядок точности по h. Заметим, что точности интерполяции функции $B(x - y)$ и вычисления интегралов по формулам (3), (18) улучшаются с увеличением расстояния между точками x и y. Это обстоятельство необходимо учитывать при выборе областей $\Pi_0(p)$.

Полученный результат можно сформулировать в матричном виде следующим образом.

Теорема. *В случае неравномерной сетки матрица A порядка N_Q рассматриваемой системы линейных алгебраических уравнений допускает приближение*

$$A \approx \tilde{A} = S_0 + S_1 T S_2,$$

где T – многоуровневая блочно-тёплицева матрица порядка $N = O(N_Q)$, соответствующая равномерной сетке, а S_0, S_1, S_2 – некоторые разреженные матрицы. Число ненулевых элементов в матрицах S_1 и S_2 имеет вид $O(N_Q)$. Число арифметических операций при умножении матрицы $\tilde{A}$ на вектор имеет вид $O(N_Q \log N_Q)$.

4. ЗАКЛЮЧЕНИЕ

В работе представлены численные методы решения объемных интегральных уравнений, которые описывают задачи рассеяния волн различной физической природы на неоднородных структурах, находящихся в трехмерной ограниченной области. Для аппроксимации интегральных уравнений системами линейных алгебраических уравнений применяется метод коллокации на неравномерной сетке. Для построения эффективного алгоритма выбирается некоторая равномерная метка и на основе методов интерполяции функций и алгоритмов быстрого дискретного преобразования Фурье строятся эффективные алгоритмы приближенного умножения матрицы системы уравнений на вектор, существенно ускорящие выполнение каждой итерации применяемого итерационного метода. Проведенный нами анализ показывает, что точность приближенного умножения определяется исключительно точностью интерполяции ядра, а число узлов вспомогательной равномерной сетки сопоставимо с числом узлов исходной неравномерной сетки.

СПИСОК ЛИТЕРАТУРЫ

1. *Colton D., Kress R.* Inverse acoustic and electromagnetic scattering theory // Appl. Math. Sci. V. 93. Berlin: Springer-Verlag, 1992.
2. *Самохин А.Б.* Интегральные уравнения и итерационные методы в электромагнитном рассеянии. М.: Радио и связь, 1998.
3. *Воеводин В.В., Тыртышников Е.Е.* Вычислительные процессы с тёплицевыми матрицами. М.: Наука, 1987.
4. *Yaghjian A.D.* Electric dyadic Green's function the source region // Proc. IEEE. 1980. V. 68. P. 248–263.
5. *Gaudi O.M.* Integration of the ordinary differential equations // J. Math. Phys. 1964. V. 5. Iss. 420. P. 420–430.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 885–894

МАТЕМАТИЧЕСКАЯ ФИЗИКА

УДК 519.63

РАСЧЕТ ИНДУКТИВНОСТЕЙ И ПРОСТРАНСТВЕННЫХ РАСПРЕДЕЛЕНИЙ ТОКОВ В МОДЕЛИ СВЕРХПРОВОДНИКОВОГО НЕЙРОНА[1)]

© 2021 г. С. В. Бакурский[1,2,3], Н. В. Кленов[4,5], М. Ю. Куприянов[1], И. И. Соловьев[1,3,6], М. М. Хапаев[1,7,*]

[1] 119991 Москва, Ленинские горы, 1, стр. 2, МГУ им. М.В. Ломоносова, НИИЯФ им. Д.В. Скобельцына, Россия

[2] 141701 Долгопрудный М.о., Институтский пер., 9, МФТИ, Россия

[3] 127055 Москва, ул. Сущевская, 22, ВНИИА им. Н.Л. Духова, Россия

[4] 119991 Москва, Ленинские горы, 1, стр. 2, МГУ им. М.В. Ломоносова, Физический факультет, Россия

[5] 111024 Москва, ул. Авиамоторная, 8а, МТУСИ, Россия

[6] 603950 Нижний Новгород, пр-т Гагарина, 23, ННГУ им. Н.И. Лобачевского, Россия

[7] 119991 Москва, Ленинские горы, 1, стр. 52, МГУ им. М.В. Ломоносова, факультет ВМК, каф. выч. методов, Россия

**e-mail: vmhap@cs.msu.su*

Поступила в редакцию 24.12.2020 г.
Переработанный вариант 24.12.2020 г.
Принята к публикации 14.01.2021 г.

Предложены математическая модель и вычислительный метод расчета индуктивностей и пространственных распределений сверхпроводящих токов в адиабатическом искусственном нейроне, представляющем собой многослойную структуру, содержащую джозефсоновские переходы. Вычислительный метод основан на совместном решении уравнений Лондонов для токов в слоях сверхпроводника и уравнений Максвелла, задающих пространственное распределение магнитного поля, а также модели листового тока, учитывающей конечную толщину проводящих слоев и токовых контактов. Этот подход эффективно учитывает межслойные контакты и джозефсоновские переходы в виде распределенных источников тока. Полученные уравнения решаются с использованием метода конечных элементов с плотными матрицами большой размерности. Представлены результаты расчетов для модели проектируемого нейрона с сигмоидальной передаточной функцией. С целью оптимизации конструкции устройства вычисляются как рабочие (запланированные на первом этапе проектирования), так и паразитные индуктивности, а также распределение токов. Предлагаемая методология и программное обеспечение могут быть использованы для моделирования широкого спектра сверхпроводящих устройств на основе сверхпроводниковых квантовых интерферометров. Библ. 19. Фиг. 4.

Ключевые слова: сверхпроводимость, искусственный нейрон, индуктивность, метод конечных элементов.

DOI: 10.31857/S0044466921050021

1. ВВЕДЕНИЕ

Одним из центральных вопросов, возникающих при разработке аналоговых и цифровых сверхпроводниковых микросхем, является создание численных алгоритмов, решающих задачу вычисления распределения сверхпроводящих токов в таких структурах, а также значений собственных и взаимных индуктивностей их отдельных частей. Это связано с тем, что работоспособность и производительность устройств на основе сверхпроводниковых квантовых интерферометров (СКВИДов) находятся в критической зависимости от индуктивностей отдельных компонентов таких устройств.

[1)]Работа выполнена при финансовой поддержке гранта РНФ 20-12-00130 (разработка численных алгоритмов решения задачи экстракции собственных и взаимных индуктивностей для сверхпроводниковых микросхем); гранта Президента РФ МД-186.2020.8 (разработка модели сверхпроводящего нейрона).

Расчет реальных индуктивностей особенно важен для устройств, преобразующих магнитный поток в ток или напряжение. Примером таких технических решений являются сверхпроводниковые антенны на основе би-СКВИДов и Д-СКВИДов. В этом случае преобразование потока в напряжение должно оставаться линейным с высокой точностью без использования цепей обратной связи [1]–[3].

В настоящее время особый интерес представляют нелинейные элементы сверхпроводниковых искусственных нейронных сетей с магнитным представлением информации. В этом случае преобразование магнитного потока в ток (функция активации) для нейрона должна с хорошей точностью совпадать с сигмоидальной функцией, либо с гиперболическим тангенсом [4].

Целью данной работы является разработка усовершенствованного вычислительного алгоритма и программы экстракции (расчета) рабочих и паразитных индуктивных коэффициентов сверхпроводниковых микросхем, в частности методики расчета индуктивностей для СКВИДов и устройств на основе СКВИДов. Эффективность разработанного подхода показана на примере одного из таких устройств, искусственного нейрона с сигмовидной активационной функцией [5]–[7]. В отличие от полупроводниковых аналогов сверхпроводниковые нейроны оказываются существенно менее сложными, и могут быть относительно просто изготовлены с использованием существующей микроэлектронной технологии. Однако их работоспособность требует высокой точности реализации нужного соотношения между значениями индуктивностей элементов схемы. В данной работе мы демонстрируем возможности и пределы применимости нашего вычислительного алгоритма и программы для вычисления индуктивностей [8] и [9] на примере решения актуальной проблемы — проектирования упомянутого сверхпроводникового нейрона.

Математическая модель, лежащая в основе расчетов пространственного распределения токов в сверхпроводниковых микроэлектронных структурах, основана на совместном решении уравнений Лондонов и Максвелла [10]. В стационарном случае такой подход приводит к интегральному уравнению относительно векторного потенциала [11], содержащему трехмерные интегралы.

Наш подход содержит ряд модификаций этого интегрального уравнения. Во-первых, используя то обстоятельство, что в рассматриваемых структурах толщина слоев невелика по сравнению с лондоновской глубиной проникновения магнитного поля в сверхпроводник, мы преобразуем объемные токи в листовые токи. Во-вторых, мы учитываем конечную толщину слоев в функции Грина, которая является ядром интегрального уравнения. В-третьих, для точного и простого учета межслойных соединений и джозефсоновских переходов мы разделяем ток на потенциальную и вихревую составляющие и представляем вихревую часть тока через функцию тока. Это сводит проблему к решению гиперсингулярного интегрального уравнения с хорошими математическими свойствами, что значительно упрощает численное решение. При таком подходе точно учитываются как магнитная, так и кинетическая части индуктивности фрагментов микросхем.

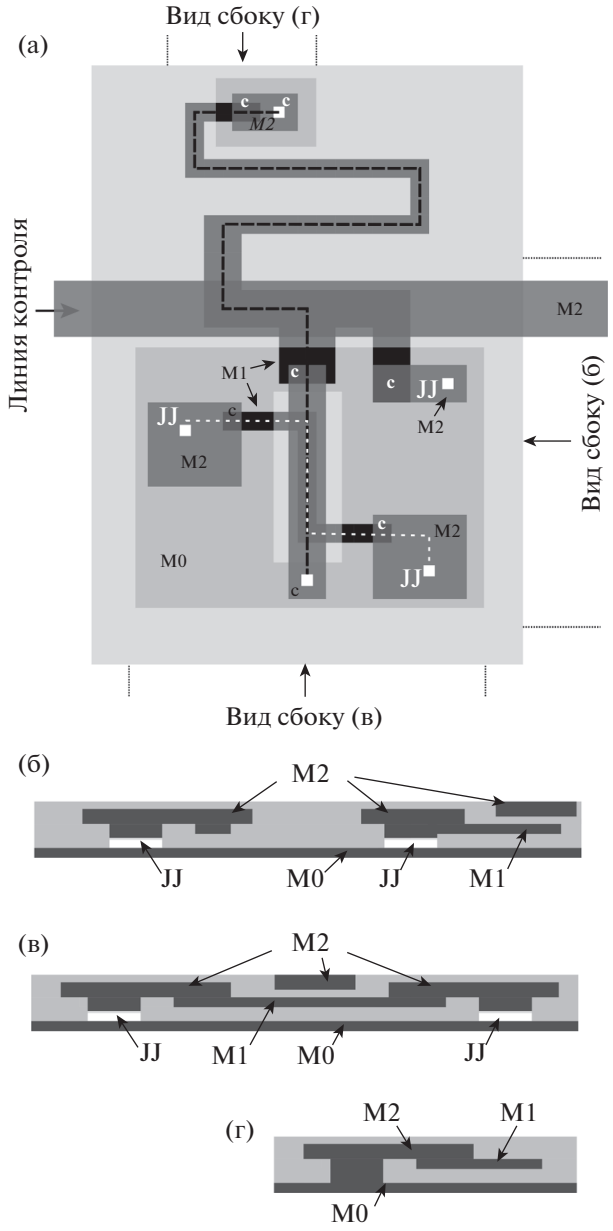
Для численного приближения и решения полученных уравнений мы используем метод конечных элементов с треугольными сетками и линейными конечными элементами. Такой подход приводит к двум плотным матрицам большой размерности. Первая матрица симметрична и имеет размерность, равную количеству ячеек в сетке. Она предназначена для хранения промежуточных результатов расчетов и для расчета полной энергии. Элементы этой матрицы Галеркина являются четырехкратными интегралами от функции Грина по ячейкам сетки. Вторая матрица является симметричной и положительно определенной. Это матрица системы линейных уравнений метода конечных элементов. Для решения этой системы линейных уравнений мы используем разложение Холецкого для плотно заполненных матриц.

Вычислительный алгоритм реализован в нашей программе 3D-MLSI [9].

В последнем разделе статьи возможности, преимущества и ограничения разработанного подхода иллюстрируются расчетом пространственного распределения сверхпроводящего тока для модели искусственного нейрона. Вычислительный алгоритм, разработанный в настоящей работе, может служить в качестве основы для применения алгоритмов скелетонной малоранговой аппроксимации плотных матриц в последующих исследованиях.

2. МОДЕЛЬ НЕЙРОНА

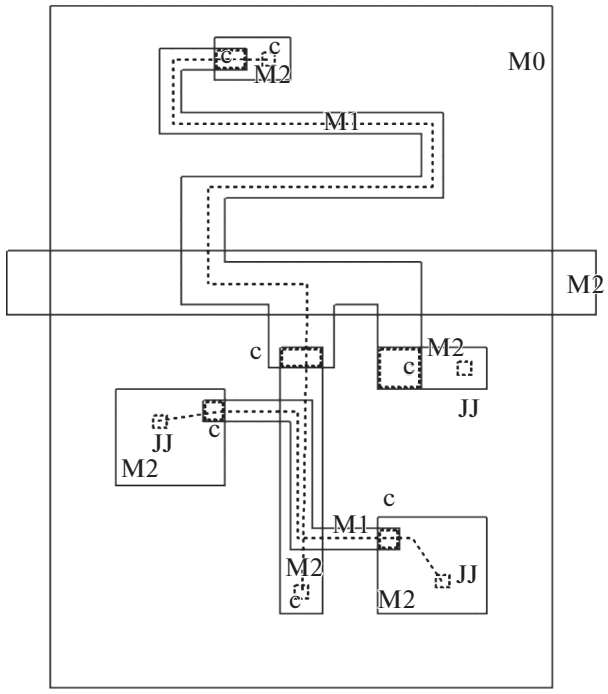
Искусственные свехпроводниковые нейроны [5]–[7] могут быть спроектированы на базе структур с двумя сверхпроводящими слоями над сверхпроводящим же экраном. Результат проектирования сверхпроводящего нейрона в виде упрощенной схемы показан на фиг. 1. Он включает в себя три разделенных диэлектриками металлических слоя: плоскость заземления M0, слой разводки M1 и слой с верхним покрытием M2. Верхний слой M2 в разных точках характеризует-



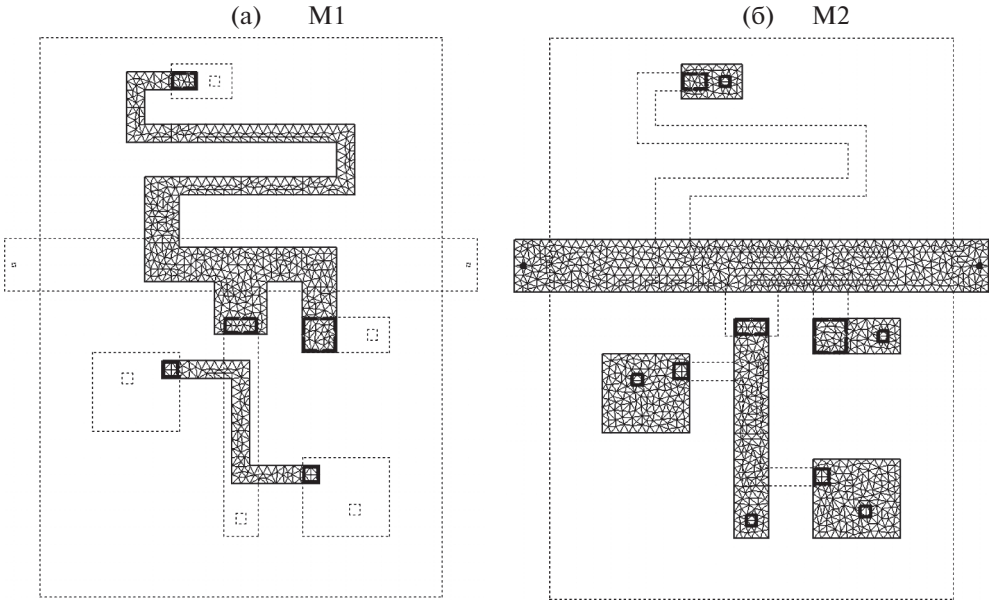
Фиг. 1. Модель нейрона, вид сверху (а) и боковые проекции (б)–(г). Показаны три сверхпроводящих слоя: слой заземления M0, средний слой M1 и верхний M2. Требуется вычислить индуктивности для двух токовых петель, показанных прерывистыми линиями и замыкающимися через слой M0. Джозефсоновские контакты между слоями M1 и M2 обозначены буквами JJ.

ся разными расстояниями до подложки (разными высотами) и частично замыкается на M1, если M1 и M2 перекрываются. В наших расчетах мы используем упрощенную модель, показанную на фиг. 2. В этой модели слой M2 имеет постоянную высоту. Замыкания слоев M1–M2 представлены в виде пары токовых контактов прямоугольной формы.

При последующих расчетах мы будем считать, что лондоновская глубина проникновения λ [10] для всех слоев одинакова и составляет 0.075 мкм. Нижний слой M0 является базовой заземляющей плоскостью. Он собирает все возвратные токи из более высоких слоев. Толщину этого слоя будем считать равной 0.3 мкм (4λ). Следующий слой – M1. Он не является сплошным и содержит несколько участков, по которым протекает ток. Полагаем, что его высота относительно верхней части нижнего слоя составляет 0.3 мкм, а толщина – 0.11 мкм (менее 2λ). Верхний слой M2 находится на высоте 0.61 мкм и имеет толщину 0.45 мкм. Между верхним и нижним слоями



Фиг. 2. Упрощенная модель нейрона. На фигуре показаны три слоя, M0 (нижний проводник, слой заземления), M1 (средний слой) и M2 (верхний слой), джозефсоновские переходы (JJ) и искусственные контакты между слоями (C). Первая токовая петля показана вертикальной пунктирной линией с возвратным током через слой M0. Вторая токовая петля, представляющая измерительный СКВИД, также показана пунктирной линией. Центральная горизонтальная полосковая линия является управляющим элементом. Размер шага сетки на фигуре 10 мкм.



Фиг. 3. Проводники в слоях M1 и M2 над замыкающей плоскостью M0 и конечно-элементная треугольная сетка.

имеется три джозефсоновских контакта и множество межслойных контактов. Оставшиеся после изготовления нейрона участки слоев представлены на фиг. 2. Средний и верхний слои отдельно представлены на фиг. 3. Там же приведена и конечно-элементная сетка. Ширина полос в M2 составляет 10 мкм. Верхний и нижний слои довольно толстые по сравнению с λ. Это ограничивает точность расчетов в нашей математической модели на уровне 5% [8].

Физическая постановка задачи наряду с описанием проводящей структуры должна содержать определение токов, для которых будут вычислены индуктивности. Эти токи, как правило, образуют токовые петли, замыкающиеся через заземляющий слой и проходящие через определенные контакты, в том числе через джозефсоновские переходы. Последовательность таких контактов определяет ток, для которого должны быть вычислены индуктивные коэффициенты. Два таких токовых контура представлены на фиг. 2. В свою очередь заданные процессы в контактах позволяют формулировать краевые и иные условия для постановки математической задачи. В результате каждый токовый контур дает одну строку (столбец) в матрице индуктивности. Диагональные элементы этой матрицы являются собственными индуктивностями, а внедиагональные элементы являются взаимными индуктивностями.

Для широких проводников, расположенных над плоскостью заземления, можно оценить собственные индуктивности, которые присутствуют в принципиальной схеме устройства. Однако невозможно достаточно точно оценить взаимные индуктивности, среди которых имеются паразитные связи, каковые невозможно заранее учесть при проектировании устройства. Единственным средством достаточно точного вычисления схемных и паразитных индуктивностей являются вычислительные методы.

3. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ДЛЯ ТОКОВ И ИНДУКТИВНОСТЕЙ

Вычислительный алгоритм этой работы является улучшенной и усовершенствованной версией алгоритмов, предложенных в [8] и [9].

Введем некоторые обозначения.

Пусть N_c – число односвязных проводников. Проводник расположен в некотором слое металлизации. Каждый проводник занимает трехмерную область:

$$V_m = S_m \times [h_m^0, h_m^1], \quad m = 1, \ldots, N_c, \quad V = \bigcup_m V_m,$$

здесь S_m есть 2D-проекция V_m на плоскость.

Пусть $\partial S_{ext,m}$ – граница проводника m, исключая границы возможно существующих отверстий. Толщина проводника m составляет $t_m = h_m^1 - h_m^0$. Пусть N_h – общее количество всех отверстий во всех проводниках. Будем называть $\partial S_{h,k}$ границей отверстия k.

Всего имеется N_T токовых контактов (терминалов) во всех проводниках, не менее двух терминалов для каждого проводника. Джозефсоновские переходы образуют пару терминалов на разных проводниках. Терминалы на $\partial S_{ext,m}$ являются внешними терминалами. Терминалы внутри S_m, например, в форме прямоугольников, являются внутренними терминалами. Джозефсоновский переход образует два внутренних терминала.

Обозначим через N_t число замкнутых или открытых путей для интересующих нас полных токов. Каждый путь содержит некоторую цепочку терминалов. Ток может переходить от проводника к проводнику, а также между слоями через терминалы. Всего имеем $N = N_t + N_h$ независимых полных токов, так как существуют замкнутые токи вокруг отверстий.

В качестве результата нас интересует вычисление $N \times N$ матрицы индуктивности L [8].

Математическая модель основана на стационарных уравнениях Лондонов и формулы Био–Савара для магнитного поля $\mathbf{H}$, а также выражения для полной энергии E:

$$\begin{gathered} \lambda^2 \nabla \times \mathbf{j} + \mathbf{H} = 0, \quad \nabla \times \mathbf{H} = \mathbf{j}, \\ E = \frac{\mu_0}{2} \iiint_V (\lambda^2 j^2 + \mathbf{j} \cdot \mathbf{A}) dv, \quad \mathbf{A}(r) = \frac{1}{4\pi} \iiint_V \frac{\mathbf{j}(r')}{|r - r'|} dv, \end{gathered} \tag{1}$$

где λ является лондоновской глубиной проникновения магнитного поля в сверхпроводник [10], [11]. Из этих выражений следует объемное интегральное уравнение для фазы $\varphi_n(r)$ сверхпроводящего параметра порядка и плотности тока $\mathbf{j}_n(r)$:

$$\lambda^2 \mathbf{j}_n(r) + \frac{1}{4\pi} \sum_{m=1}^{N_c} \iiint_{V_m} \frac{\mathbf{j}_m(r')}{|r - r'|} dv' = \nabla \varphi_n(r), \quad n = 1, \ldots, N_c. \tag{2}$$

Неизвестными являются фаза и плотность тока. Объемное интегральное уравнение, дополненное условием $\nabla \mathbf{j}_n = 0$, является основой численных методов для расчета токов $\mathbf{j}_n(r)$, а затем индуктивностей [12]–[15]. В нашем подходе мы используем некоторые преобразования (2), позволяющие ввести токи вокруг отверстий и существенно упростить численное решение.

Существует ряд специфических трудностей в моделировании сверхпроводящих токов. Первой трудностью является сложная топология распределения тока в объемной структуре. Известно [10], [11], что уравнение Лондонов приводит к концентрации тока вблизи поверхности сверхпроводников. Точный учет этого эффекта довольно сложен алгоритмически и приводит к продолжительным вычислениям, однако при некоторых условиях (тонкий проводник) его общее влияние на значения индуктивности является умеренным.

Другой трудностью является достаточно точное моделирование контактов и джозефсоновских переходов между слоями, так как контакт или переход занимают некоторую конечную область. Поэтому возникает необходимость использования физически релевантных моделей для растекания тока на терминалах.

Наконец, токи вокруг отверстий не имеют терминалов, так что их невозможно задать с помощью краевых условий для тока и фазы.

В нашем подходе мы используем возможность упростить задачу для случая сверхпроводящих слоев относительно небольшой, но конечной толщины. Обычно толщина сверхпроводящих слоев составляет порядка 1–3 лондоновской глубины проникновения. В этом случае для проводника m мы можем использовать модель листового тока $\mathbf{J}_m(x, y) = (J_{m,x}(x, y), J_{m,y}(x, y))$. Полагая $j_z \approx 0$, получаем

$$\mathbf{J}_{m,x}(x, y) = \int_{h_m^0}^{h_m^1} \mathbf{j}_{m,x}(x, y, z)dz, \quad \mathbf{J}_{m,y}(x, y) = \int_{h_m^0}^{h_m^1} \mathbf{j}_{m,y}(x, y, z)dz. \tag{3}$$

Конечная толщина проводящих слоев эффективно учитывается путем усреднения уравнения (2) по толщине слоя и введения некоторой простой функции Грина вместо потенциала простого слоя с ядром $1/|r - r'|$, $r = (x, y)$:

$$\lambda_s^n \mathbf{J}_n(r) + \frac{1}{4\pi} \sum_{m=1}^{N_c} \iint_{s_m} \mathbf{J}_m(r') G_{mn}(r, r')ds' = \nabla \varphi_n(r), \tag{4}$$

$$G_{mn}(r, r') = \frac{1}{4} \sum_{k=0}^{1} \sum_{l=0}^{1} \left(|r - r'|^2 + (h_m^k - h_n^l)^2 \right)^{-1/2}. \tag{5}$$

В этом выражении $\lambda_m^s = \lambda^2/t_m$ – лондоновская глубина проникновения для тонких сверхпроводящих пленок.

Затем с помощью оператора ротора $\nabla_\perp = (\partial_y, -\partial_x)$ интегральное уравнение для листовых токов преобразуется в гиперсингулярное интегральное уравнение. Гиперсингулярная форма уравнения предпочтительнее для численного решения, так как его интегральный оператор является симметричным и положительно-определенным.

Более того, в (2) мы исключаем член $\nabla \varphi_n(r)$. Для этого плотность тока сверхпроводимости представляется в виде суммы двух компонент:

$$\mathbf{J}_m(r) = \mathbf{J}_m^{\text{pot}}(r) + \mathbf{J}_m^{\text{rot}}(r).$$

Полагаем

$$\mathbf{J}_m^{\text{pot}}(r) = (\lambda_m^s)^{-1} \nabla \varphi_m(r), \quad \mathbf{J}_m^{\text{rot}}(r) = \nabla_\perp \psi_m(r). \tag{6}$$

Будем полагать $\mathbf{J}_m^{\text{pot}}(r)$ терминальным током возбуждения, $\mathbf{J}_m^{\text{rot}}(r)$ представляет собой ток экранирования магнитного поля и ток вокруг отверстий.

Окончательно приходим к следующему интегральному уравнению для функции тока $\psi_n(r)$, $n = 1, \ldots, N_c$:

$$-\lambda_s^n \Delta\psi_n(r) + \frac{1}{4\pi}\sum_{m=1}^{N_c}\iint_{s_m}(\nabla\psi_m(r'), \nabla' G_{mn}(r,r'))ds' + F_n(r) = 0, \tag{7}$$

$$F_n(r) = \frac{1}{4\pi}\sum_{m=1}^{N_c}\iint_{s_m}\frac{1}{\lambda_s^m}(\nabla\varphi_m(r'), \nabla_\perp G_{mn}(r,r'))ds'. \tag{8}$$

Пусть $I_{h,k}$ – полный ток, циркулирующий вокруг отверстия k. Тогда граничными условиями для функции тока являются [8]:

$$\psi_m(r) = I_{h,k}, \quad r \in C_{h,k}, \quad \psi_m(r) = 0, \quad r \in C_{ext,m}. \tag{9}$$

Здесь $C_{h,k}$ – двумерная граница дырки k. В настоящее время отверстия в физическом дизайне рассматриваемых моделей нейронов отсутствуют. Далее, $C_{ext,m}$ – внешняя граница проводника m. Подробно краевые условия для функции тока представлены в [8]. Если в схеме нет отверстий, то на всех границах $\psi_m(r) = 0$.

Для потенциала $\varphi_n(r)$ в (8) ставится вторая краевая задача

$$\nabla\cdot\frac{1}{\lambda_s^n}\nabla\varphi_n(r) = f_n(r), \tag{10}$$

$$\frac{1}{\lambda_s^n}\frac{\partial\varphi_n(r)}{\partial\mathbf{n}} = 0, \quad r \in \partial S_{ext,n} \cup \partial S_{h,k}, \quad n = 1, \ldots, N_c, \tag{11}$$

где $\mathbf{n}$ – нормаль. Функция $f_n(r)$ определяет модель терминала как распределенный источник или сток. Пусть I_k – полный ток через терминал k с областью $T_k \in S_m$ и площадью $|T_k|$, тогда:

$$f_m(r) = \frac{I_k}{|T_k|}, \quad r \in T_k, \quad f_m(r) = 0, \quad r \notin T_k. \tag{12}$$

Форма терминалов может быть произвольной. Для контактов большой площади можно моделировать лондоновскую неоднородность тока с помощью кольцевых терминалов [9].

Таким образом, для достижения общей цели необходимо последовательно решить ряд задач. Сначала решаются N_c задач (10), (11) с заданным I_k для $\mathbf{J}_m^{\text{pot}}(r)$. После этого решаются задачи (7)–(9). Наконец, полная энергия рассчитывается с использованием выражения:

$$E = \frac{\mu_0}{2}\sum_{n=1}^{N_c}\iint_{S_n}\left(\lambda_s^n J_n^2(r) + \frac{1}{4\pi}\sum_{m=1}^{N_c}\iint_{S_m}\mathbf{J}_n(r)\cdot\mathbf{J}_m(r')G_{mn}(r,r')ds'\right)ds. \tag{13}$$

Матрица индуктивности содержит все индуктивные коэффициенты и рассчитывается с помощью выражения для полной энергии [8], [9].

Для однослойной структуры ядра $1/|r-r'|$ и одного проводника уравнение для функции тока имеет вид

$$-\lambda_s\Delta\psi(r) + \frac{1}{4\pi}\iint_S\left(\nabla\psi(r'), \nabla'\frac{1}{|r-r'|}\right)ds' + F(r) = 0. \tag{14}$$

Для оператора в смысле главного значения имеем

$$L\psi(r) = \frac{1}{4\pi}\iint_S\left(\nabla\psi(r'), \nabla'\frac{1}{|r-r'|}\right)ds' = -\frac{1}{4\pi}\iint_S\frac{\psi(r')}{|r-r'|^3}ds'. \tag{15}$$

В [16] доказано, что для функций, принимающих нулевое значение на границе, оператор L положителен.

Формула (15) важна для доказательства свойств интегрального оператора и корректности метода конечных элементов. Кроме того, представление интегрального оператора в гиперсингу-

лярной форме может быть использовано для быстрой оценки и вычисления матричных элементов в методе конечных элементов для удаленных друг от друга точек сетки (см. [17]).

Если геометрические параметры задачи и лондоновская глубина проникновения представлены в микронах, то можно считать, что все уравнения приведены к безразмерной форме, а размерность индуктивностей составляет пикогенри.

4. ВЫЧИСЛИТЕЛЬНЫЙ АЛГОРИТМ И ПРОГРАММА

Для решения обеих краевых задач (10), (11) и (7), (8) используется метод конечных элементов. Мы используем треугольные сетки и линейные лагранжевы конечные элементы [18]. Такой выбор приводит к простому и быстрому алгоритму сборки общей матрицы метода конечных элементов.

Для уравнения Пуассона (10), (11) используется аналогичный метод конечных элементов. Этот алгоритм описан в литературе.

Для краткости рассмотрим задачу с одним проводником. Тогда билинейная форма задачи для функции тока имеет вид

$$a(u,v) = -\lambda_s \iint\limits_S (\nabla u, \nabla v)ds + \frac{1}{4\pi}\iint\limits_S ds \iint\limits_S (\nabla u(r), \nabla v(r'))G(r,r')ds'. \tag{16}$$

Сеточная аппроксимация функции тока с главными краевыми условиями имеет вид

$$\psi(r) \approx \psi^h(r) = \sum_{j\in J} \psi_j^h u_j^h(r), \tag{17}$$

где суммирование проводится по всем точкам сетки J, включая граничные точки, ψ_j^h – приближенные значения функции тока в точках сетки, $u_j^h(r)$ – базисные функции интерполяции метода конечных элементов.

В нашем вычислительном алгоритме используются линейные лагранжевы конечные элементы (P1). Эти элементы обладают достаточной гладкостью и удобны для аппроксимации гиперсингулярного оператора (15).

Такой метод конечных элементов приводит к системе линейных уравнений для ψ_i^h, $i \in I$, I – внутренние точки сетки:

$$\sum_{j\in J} a(u_i^h, u_j^h)\cdot \psi_j^h = 0, \quad i \in I. \tag{18}$$

Матрица в (18) плотная, симметричная и при достаточно малом шаге сетки положительно-определенная. Размерность этой матрицы равна числу внутренних точек сетки N_I. Граничные значения $\psi(r)$ формируют правую часть как главное краевое условие.

Таким образом, матричные элементы конечно-элементной матрицы K имеют вид

$$(K)_{ij} = a(u_i^h, u_j^h) = \lambda_s \iint\limits_{S_i\cap S_j} (\nabla u_i^h, \nabla u_j^h)ds + \frac{1}{4\pi}\iint\limits_{S_i} ds \iint\limits_{S_j} (\nabla u_j^h(r'), \nabla u_i^h(r))G(r,r')ds', \tag{19}$$

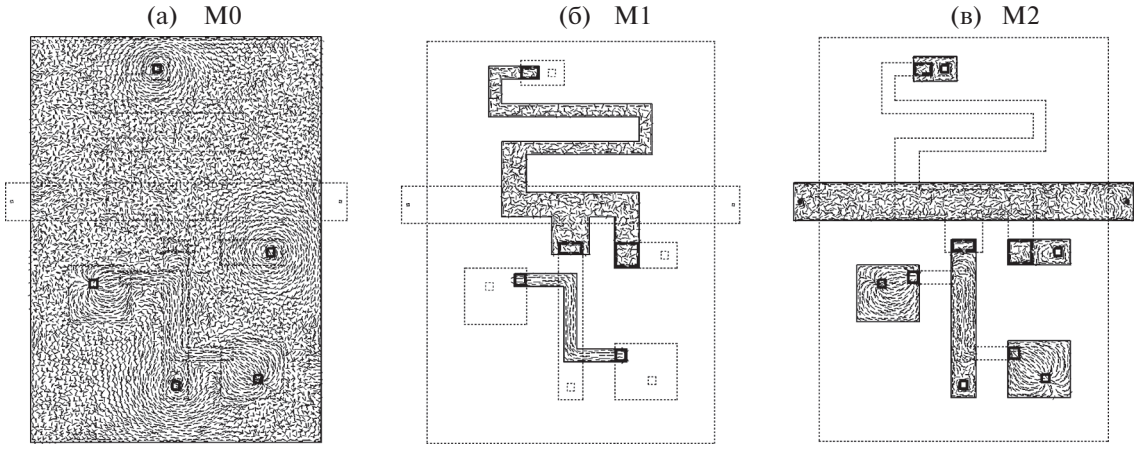
где S_i, S_j – носители базисных функций конечно-элементной аппроксимации u_i^h, u_j^h. Эффективный метод вычисления этих элементов описан в [8] и [9], [17].

Обоснование описанного метода конечных элементов для гиперсингулярного уравнения с ядром $1/|r-r'|^3$ было дано в [16].

Для решения линейных уравнений описанного метода конечных элементов мы используем разложение Холецкого для плотных матриц. Метод конечных элементов для (10), (11) основан на применении разреженных матриц.

Для построения треугольных сеток наша программа содержит собственный генератор. Также можно использовать известный генератор треугольных сеток Triangle [19].

В алгоритме есть несколько вычислительных процедур, занимающих много времени процессора. Первой процедурой является вычисление матрицы K (19) и правой части $F(r)$ в (8). Второй процедурой является расчет полной энергии, определяемой выражением (13). Чтобы ускорить эти вычисления, мы вводим матрицу Галеркина T взаимодействий между треугольниками на



Фиг. 4. Линии тока в слоях M0, M1 и M2 для петли измерительного СКВИДа.

сетке метода конечных элементов, которая является общей частью обоих вычислений. Пусть Δ_i и Δ_j – треугольные ячейки, тогда элементы матрицы взаимодействия T являются четырехкратными интегралами:

$$T_{ij} = \frac{1}{4\pi}\iint\limits_{\Delta_i} ds' \iint\limits_{\Delta_j} G(r, r')ds. \tag{20}$$

Таким образом, вычислительный алгоритм основан на двух матрицах, T и K. Матрица T имеет размерность, равную числу ячеек в сетке N_t. Вторая матрица, K, является матрицей системы линейных уравнений метода конечных элементов, размерность этой матрицы – число внутренних узлов сетки N_I. N_I и N_t в практически интересных случаях могут быть большими (несколько тысяч) или очень большими (десятки тысяч). В настоящее время мы вычисляем эти плотно заполненные матрицы и храним в явной форме. В данной работе мы демонстрируем решение одной практической задачи с помощью этого подхода. В то же время мы рассматриваем разработанный вычислительный подход как основу для разработки и реализации алгоритмов скелетонного типа и малоранговой аппроксимации для работы с плотно заполненными матрицами большой размерности.

Программная реализация численного алгоритма содержит вычислительные модули, написанные на C++ и работающие в Linux или Windows. Также нами разработан графический Windows интерфейс для визуализации входных данных и распределения тока [9].

5. РЕЗУЛЬТАТЫ РАСЧЕТОВ

Рассмотрим модель нейрона для многослойного персептрона, см. фиг. 2. Исходные данные для расчетов первоначально были подготовлены в САПР в формате DXC (DXF), а затем в полуавтоматическом режиме преобразованы во входные данные для программы 3D-MLSI.

Схема содержит множество возможных токовых структур, которые можно использовать для прямого или косвенного вычисления схемных и паразитных индуктивностей. Для краткости рассмотрим пример с двумя замкнутыми токами. Во-первых, это горизонтальная замкнутая токовая петля с собственной индуктивностью L_{11}. Эта петля моделирует измерительный СКВИД с двумя джозефсоновскими контактами и возвратным током через заземленную поверхность. Второй контур с собственной индуктивностью L_{22} – это вертикальная (на фигуре) петля с одним джозефсоновским переходом и обратным током через плоскость заземления. Обе петли показаны на фиг. 3.

Для расчета описанных индуктивностей с точностью 5–6% решалась задача с размерностями матриц $N_t = 12\,722$ и $N_I = 5770$. Всего система уравнений с матрицей K решалась 3 раза. Были получены значения $L_{11} = 9.05$, $L_{12} = 1.8$ и $L_{22} = 26.3$ пикогенри. Линии тока для расчета L_{11} представлены на фиг. 4. Ток через первый контур создает близкие к прямым линии тока. Токи экранирования на других частях конструкции образуют вихревые структуры. Полученные результаты были далее использованы при моделировании процессов в таком нейроне, а также для дальнейшей оптимизации его структуры.

Описанные вычисления оказались достаточно быстрыми и заняли несколько минут на процессоре Intel i7 в одноядерном режиме. Те же вычисления с более плотной сеткой и $N_I = 18\,694$ и $N_t = 42\,842$ заняли всю 16-гигабайтную память и потребовали несколько часов счета.

5. ВЫВОДЫ

Математическая модель настоящей работы является достаточно точной и эффективной для сверхпроводниковых схем с одним или несколькими слоями металлизации и толщиной слоев, сопоставимой с лондоновской глубиной проникновения. Алгоритм и программа применимы как к задачам с элементами схемы, несущими весь возвратный ток, так и к задачам без таких элементов, например, однослойным.

Вычислительный метод настоящей работы, программа и методология расчетов могут быть довольно просто использованы для вычисления индуктивностей схем небольших и умеренных размеров. Для моделирования больших схем требуется дополнить программу средствами автоматической подготовки входных данных, использующими данные систем автоматизированного проектирования и разработки (САПР) микросхем.

Другим необходимым и важным шагом является использование современных методов работы с плотно заполненными матрицами большой размерности с целью уменьшения требований к необходимой памяти и ускорения вычислений.

Авторы благодарят В. Больгинова за плодотворные обсуждения возможных реализаций моделей сверхпроводниковых нейронов.

СПИСОК ЛИТЕРАТУРЫ

1. *Soloviev I.I., Klenov N.V., Schegolev A.E., Bakurskiy S.V., Kupriyanov M.Yu.* Analytical derivation of DC SQUID response // Superconductor Sci. and Technology. 2016. V. 29. № 9. P. 094005.
2. *Kornev V.K., Kolotinskiy N.V., Bazulin D.E., Mukhanov O.A.* High linearity bi-SQUID: Design map // IEEE Transactions on Appl. Superconductivity. 2018. V. 28. № 7. P. 1–5.
3. *Soloviev I.I., Ruzhickiy V.I., Klenov N.V., Bakurskiy S.V., Kupriyanov M.Yu.* A linear magnetic flux-to-voltage transfer function of a differential DC SQUID // Superconductor Sci. and Technology. 2019. V. 32. № 7. P. 074005.
4. *Katayama H., Fujii T., Hatakenaka N.* Theoretical basis of SQUID-based artificial neurons // J. of Applied Physics. 2018. V. 124. № 15. P. 152106.
5. *Soloviev I.I., Schegolev A.E., Klenov N.V., Bakurskiy S.V., Kupriyanov M.Y., Tereshonok M.V., Shadrin A.V., Stolyarov V.S., Golubov A.A.* Adiabatic superconducting artificial neural network: Basic cells // J. of Applied Physics. 2018. V. 124. № 15. P. 152113.
6. *Klenov N.V., Schegolev A.E., Soloviev I.I., Bakurskiy S.V., Tereshonok M.V.* Energy efficient superconducting neural networks for high-speed intellectual data processing systems // IEEE Transactions on Appl. Superconductivity. 2018. V. 28. № 7. P. 1–6.
7. *Schegolev A.E., Klenov N.V., Soloviev I.I., Tereshonok M.V.* Adiabatic superconducting cells for ultra-low-power artificial neural networks // Beilstein J. Nanotechnol. 2016. V. 7. P. 1397–1403.
8. *Khapaev M.M.* Inductance extraction of multilayer finite-thickness superconductor circuits // IEEE Transactions on Microwave Theory and Techniques. 2001. V. 49. P. 217–220.
9. *Khapaev M.M., Kupriyanov M.Y.* Inductance extraction of superconductor structures with internal current sources // Superconductor Sci. and Technology. 2015. V. 28. № 5. P. 055013.
10. *Schmidt V.V.* The Physics of Superconductors: Introduction to Fundamentals and Applications. Berlin: Springer, 2010.
11. *Orlando T.P., Delin K.A.* Foundations of Applied Superconductivity. Addison–Wesley, 1991.
12. *Kamon M., Tsuk M.J., White J.K.* FASTHENRY: a multipole-accelerated 3D inductance extraction program // IEEE Transactions on Microwave Theory and Techniques. 1994. V. 42. № 9. P. 1750–1758.
13. *Yucel A.C., Georgakis I.P., Polimeridis A.G., Bagci H., White J.K.* VoxHenry: FFT-Accelerated Inductance Extraction for Voxelized Geometries // IEEE Transactions on Microwave Theory and Techniques. 2018. V. 66. № 4. P. 1723–1735.
14. *Whiteley S.R.* Fasthenry 3.0wr. http://www.wrcad.com
15. *Fourie C.J., Jackman K.* Software tools for flux trapping and magnetic field analysis in superconducting circuits // IEEE Transactions on Appl. Superconductivity. 2019. V. 29. 1301004.
16. *Ervin V.J., Stephan E.P.* A boundary element Galerkin method for a hypersingular integral equation on open surfaces // Mathematical Methods in the Applied Sciences. 1990. V. 13. P. 281–289.
17. *Khapaev M.M., Kupriyanov M.Y.* Sparse approximation of FEM matrix for sheet current integro-differential equation // Matrix Methods: Theory, Algorithms and Applications. Dedicated to the Memory of Gene Golub. 2010. P. 510–522.
18. *Jin J.M.* The Finite Element Method in Electromagnetics. John Wiley, 2015.
19. *Shewchuk J.R.* Delaunay refinement algorithms for triangular mesh generation // Computational Geometry: Theory and Applications. 2002. V. 22. I. 1–3. P. 21–74.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 895

МАТЕМАТИЧЕСКАЯ ФИЗИКА

УДК 519.63

ПЕРСПЕКТИВЫ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ ТЕНЗОРНЫХ РАЗЛОЖЕНИЙ ДЛЯ МОДЕЛИРОВАНИЯ КОЛЛЕКТИВНОЙ ЭЛЕКТРОСТАТИКИ В МНОГОЧАСТИЧНЫХ СИСТЕМАХ[1)]

© 2021 г. В. Х. Хоромская[1,*], Б. Н. Хоромский[1,2,]**

[1] *D-04103 Leipzig, Inselstr. 22–26, Max Planck Institute for Mathematics in the Sciences, Germany*

[2] *Magdeburg, Max Planck Institute for Dynamics of Complex Technical Systems, Germany*

**e-mail: vekh@mis.mpg.de*

***e-mail: bokh@mis.mpg.de*

Поступила в редакцию 24.12.2020 г.
Переработанный вариант 24.12.2020 г.
Принята к публикации 14.01.2021 г.

В настоящее время использование структурированных малоранговых тензорных методов привело к прогрессу в задачах численного исследования электростатистических задач многочастичных систем с дальнодействующими взаимодействиями и соответствующими энергиями и силами. В данной статье предлагается обзор перспектив численного моделирования коллективного электростатического потенциала на решетках и в многочастичных системах общего типа с использованием тензорных разложений. Данный подход, исходно предложенный для структурированных по рангу сеточных вычислений потенциалов взаимодействия на трехмерных решетках, обобщается в этой работе для случая многочастичных систем с различными зарядами, расположенными на решетках в многомерных областях вида $L^{\otimes d}$, дискретизированных на мелких декартовых сетках вида $n^{\otimes d}$ для произвольных значений размерности d. В результате потенциал взаимодействия представляется в параметрическом малоранговом каноническом формате со сложностью $O(dLn)$. Полная энергия взаимодействия далее может быть вычислена за $O(dL)$ операций. Электростатика для больших биомолекулярных систем дискретизируется на мелкой сетке $n^{\otimes 3}$ с использованием нового тензорного формата с разделением по диапазонам (RS) [3], который поддерживает дальнодействующую часть трехмерного коллективного потенциала многочастичной системы в параметрической малоранговой форме сложности порядка $O(n)$. Демонстрируется, что поле сил можно легко восстановить с использованием предварительно вычисленного электрического поля в малоранговом RS-формате. RS-представление дискретизированной дельты Дирака [4] позволяет построить эффективную консервативную по энергии схему регуляризации для решения трехмерных эллиптических уравнений в частных производных с сильно сингулярными правыми частями, возникающими при научных вычислениях. Основной вывод состоит в том, что методы аппроксимации на основе тензоров с ранговой структурой предоставляют многообещающие численные инструменты для приложений к динамике многих тел в бионауках, докингу белков и задачам классификации, для малопараметрической интерполяции разрозненных данных в науках о данных, а также в машинном обучении во многих измерениях. Библ. 76. Фиг. 9. Табл. 3.

Ключевые слова: потенциал Кулона, потенциал Слейтера, дальние многочастичные взаимодействия, малоранговые тензорные разложения, тензорные форматы с разделением по диапазонам, суммирование электростатических потенциалов, вычисление энергии и сил.

DOI: 10.31857/S0044466921050112

[1)]Полный текст статьи печатается в английской версии журнала.

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 896–910

ИНФОРМАТИКА

УДК 519.65

ОБЗОР МЕТОДОВ ВИЗУАЛИЗАЦИИ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ[1)]

© 2021 г. С. А. Матвеев[1,2,*], И. В. Оселедец[1,2,], Е. С. Пономарев[1,***], А. В. Чертков[1]**

[1] 121205 Москва, Большой бульвар, 30, стр. 1, Сколковский институт науки и технологий, Россия

[2] 119333 Москва, ул. Губкина, 8, Институт вычислительной математики им. Г.И. Марчука Российской академии наук, Россия

**e-mail: s.matveev@skoltech.ru*

***e-mail: evgenii.ponomarev@skoltech.ru*

****e-mail: andrei.chertkov@skolkovotech.ru*

Поступила в редакцию 24.11.2020 г.
Переработанный вариант 24.11.2020 г.
Принята к публикации 11.12.2020 г.

Современные алгоритмы, основанные на искусственных нейронных сетях, крайне полезны при решении множества сложных задач компьютерного зрения, робастного управления, анализа звука и текстов на естественном языке в приложениях обработки данных, робототехники и т.д. Однако для успешного внедрения нейросетевого подхода в критически значимые системы, например, в медицине или в судебной практике, необходима понятная человеку интерпретация внутренней архитектуры и процесса принятия решений сетью. В последние годы особую распространенность для создания интерпретируемых моделей глубокого обучения приобрели методы анализа, основанные на различных техниках визуализации, применяемых к графу вычислений, профилю функции потерь, к параметрам отдельных слоев сети и даже к отдельным нейронам. В данном обзоре систематизируются существующие математические методы анализа и объяснения поведения соответствующих алгоритмов и приводятся постановки соответствующих задач вычислительной математики. Исследование и визуализация глубоких нейронных сетей являются новыми, малоизученными, и в то же время бурно развивающимися областями. Рассмотренные методы позволяют заглянуть вглубь и лучше понять работу нейросетевых алгоритмов. Библ. 57. Фиг. 5. Табл. 2.

Ключевые слова: искусственная нейронная сеть, интеллектуальный анализ данных, машинное обучение, глубокое обучение, визуализация искусственной нейронной сети.

DOI: 10.31857/S0044466921050148

1. ВВЕДЕНИЕ

Современные вычислительные технологии и алгоритмы все чаще используют нейронные сети. Искусственные нейронные сети (далее ИНС) и глубокое обучение (Deep Learning, далее DL) [1] стали практически незаменимыми в приложениях анализа больших объемов данных, машинного зрения, автоматической обработки естественного языка и др. На сегодняшний день ИНС уже находят применение в автономных роботизированных системах на производстве, в автоматизированных биомедицинских системах, в системах автономного вождения автомобилей и в широком спектре иных робототехнических приложений (см., например, [2]–[4]). Однако для дальнейшего развития нейросетевого подхода и возможности полноценного использования ИНС в критически значимых практических областях, например, в медицине, в судебной или финансовой системах, где цена ошибки очень высока, необходима возможность создания интерпретируемых DL моделей (Explainable Deep Learning, далее EDL) [5]–[7]. На сегодняшний день ИНС в подобных приложениях используются преимущественно лишь в качестве систем поддержки принятия решений, т.е. конечное решение, которое принимается с учетом мнения ИНС, остается все-таки за человеком – специалистом в соответствующей области знания.

[1)]Работа выполнена при финансовой поддержке Минобрнауки РФ (проект № 075-15-2020-801).

Таким образом, основной целью данной работы является обзор математических методов и технологий анализа работы широкого класса вычислительных алгоритмов – искусственных нейронных сетей.

В широком смысле EDL может рассматриваться как понятное человеку объяснение, почему конкретное решение было принято конкретной искусственной нейросетевой моделью. Подобное объяснение может быть полезным в трех важных направлениях.

1. *Понимание модели*, связанное с нахождением зависимостей между конкретной реализацией ИНС и механизмами ее внутреннего функционирования с одной стороны, и даваемыми ИНС предсказаниями с другой стороны.

2. *Отладка модели*, связанная с поиском дефектов структуры ИНС или артефактов обучения при возникновении проблемы со сходимостью процесса обучения или наличия не оптимального режима функционирования.

3. *Улучшение модели*, связанное с динамическим внесением модификаций в ИНС на основе экспертных оценок и конкретных знаний из моделируемой предметной области.

В современной литературе часто вводятся два термина: “explainability” и “interpretability”. Также может отдельно рассматриваться вопрос, связанный с тем, какому именно человеку объяснение понятно – объяснение модели, наглядное для специалиста в области машинного обучения (Machine Learning, далее ML), может оказаться совершенно не понятным финансисту, врачу и т.п. Мы не будем углубляться в эти вопросы (подробное обсуждение можно найти, например, в книге [8]), и далее в работе употребляем единый термин EDL, предполагая, что его смысл понятен из контекста.

Зрение является для человека основным инструментом изучения окружающего мира, в этой связи одним из наиболее перспективных подходов к разработке EDL моделей, в контексте всех трех обозначенных выше направлений, оказывается *визуализация*. Визуальное представление может строиться для графа вычислений или профиля функции потерь, а также для параметров отдельных слоев ИНС, и даже для отдельных нейронов.

Развитие методов визуализации ИНС может привести к появлению полезных инструментов для использования учеными в фундаментальных исследованиях в области наук о мозге. И, наоборот, современные исследования в науках о мозге могут послужить катализатором для соответствующих разработок в области ML и непосредственно визуализации ИНС. В частности, они позволяют поставить важный вопрос о формировании в ИНС аналогов памяти и специализации нейронов, присутствующих в естественных нейронных сетях [9]–[11]. Интересным направлением исследований также является анализ способов получения максимального отклика от заданных групп искусственных нейронов на определенные типы “раздражителей” [12] по аналогии с соответствующими результатами из нейронаук. Так, например, в работе [13] в рамках экспериментов с мышами исследователи обнаружили особые (“главные”) нейроны, ответственные за конкретный приобретенный поведенческий навык. Безусловно, современные ИНС имеют ряд существенных отличий от естественных нейронных сетей, однако применение схожих подходов для исследования естественных и искусственных нейронных сетей представляется уместным и перспективным.

Отметим, что систематический интегральный подход к задаче визуализации ИНС начал формироваться лишь в последние пять лет, однако на сегодняшний день в литературе уже представлен ряд обзоров, затрагивающих в той или иной степени эту тематику. После обсуждения перспективных методов и программного обеспечения для визуализации ИНС нами будет проведен краткий анализ этих работ. Актуальность предлагаемого обзора связана с необходимостью совместного рассмотрения новейших алгоритмов и программных решений в области визуализации глубоких нейросетевых структур, имеющих различные архитектуры и назначения, а также систематизации уже существующих обзоров по данной теме.

2. МЕТОДЫ ВИЗУАЛИЗАЦИИ

Исследование и визуализация глубоких ИНС являются новыми, малоизученными и в то же время бурно развивающимися областями, позволяющими заглянуть вглубь и лучше понять работу нейросетевых алгоритмов. На сегодняшний день существует большое разнообразие методов визуализации, относящихся к различным сущностям: архитектура сети, процесс обучения, функционал потерь, поведение отдельных слоев и отдельных нейронов. В данном разделе мы рассмотрим наиболее распространенные на сегодняшний день подходы, а в последующих разде-

лах мы обсудим соответствующее программное обеспечение и ряд обзорных статей, более подробно затрагивающих определенные группы методов.

2.1. Максимизация активации

Разберем постановку задачи и методы, связанные с обнаружением и изучением стимулов, активирующих конкретные единичные нейроны или их малые группы [13]. Под стимулами в приложениях компьютерного зрения чаще всего понимаются изображения, а соответствующая задача ставится как поиск такого из них, которое бы максимизировало реакцию анализируемого нейрона, и известна в литературе под общим названием *максимизация активации* [14] (Activation Maximization, далее AM). Отметим, что в последнее время методы, решающие задачу AM, стали активно применяться в задаче визуализации ИНС.

В случае изучения зрения живых существ стимулом является видимое изображение, а активацией – электрический сигнал, снимаемый с нейрона в мозге. Одной из первых соответствующих работ биологов была работа, в которой было обнаружено, что отдельный нейрон в мозге кошки сильнее всего реагирует на изображение наклонных линий [15]. Подобный подход может быть применен также и для ИНС.

Простейшим способом проанализировать, на что реагирует тот или иной нейрон, является поиск изображения (например, из тренировочной выборки), которое максимизирует функцию активации исследуемого нейрона. Однако в таком подходе кроется ряд недостатков. Во-первых, в этом случае требуется произвести поиск по всей обучающей выборке для всех интересующих исследователя нейронов, что ведет к значительным затратам вычислительных ресурсов. Во-вторых, выборка может не содержать изображение, которое бы максимизировало выход исследуемого нейрона, так как пространство изображений обычно сильно больше размера выборки. И, наоборот, нейрон может активироваться совершенно разными изображениями приблизительно одинаково сильно, что усложняет интерпретируемость. Обычно в таких случаях рассматривают 9 наиболее сильных стимулов, но и они могут быть разрознены. И, наконец, для случая настоящих изображений, часто возникает неоднозначность – какие именно визуальные особенности заставляют нейрон реагировать. Например, если нейрон активируется изображением птицы на ветке дерева, то не понятно, важна ли здесь птица или ветка.

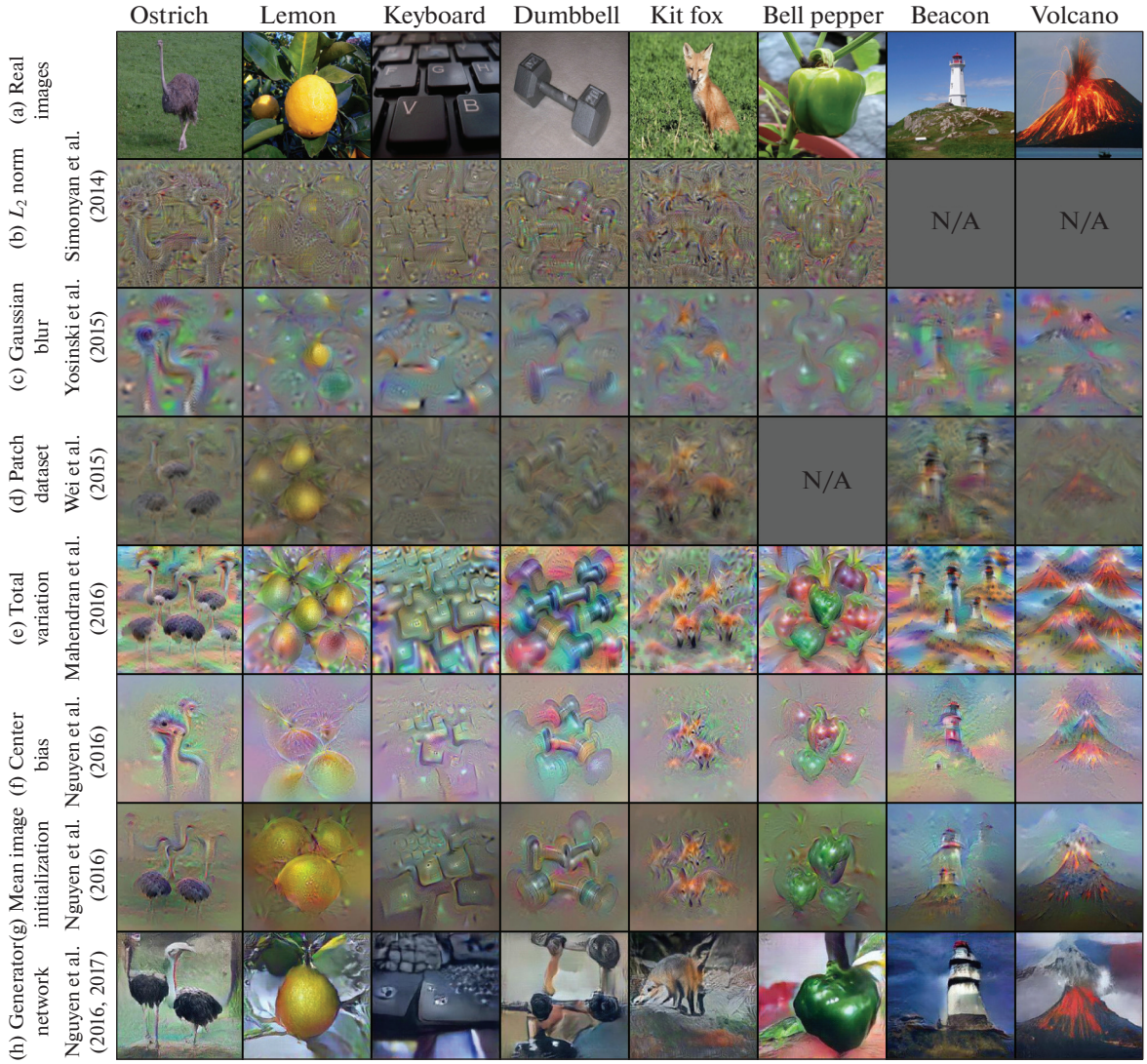
Для компенсации части обозначенных выше недостатков возможен синтез изображений. Такой подход позволяет отделять стимулы друг от друга и обладает большей репрезентативностью. Так, можно реконструировать стимулы без необходимости доступа к обучающей выборке целевой модели, которая может быть недоступна на практике, также можно контролировать количество объектов и их вид на изображении, например, создавать изображения только птиц или только веток. Отметим, что современные методы в основном используют именно синтетические изображения.

Рассмотрим задачу классификации изображений. Пусть Θ – параметры классификатора, отображающего картинку из C каналов размера $H \times W$ точек $x \in \mathbb{R}^{H \times W \times C}$, в распределение вероятностей по выходным классам. Тогда мы можем сформулировать задачу максимизации активации нейрона с индексом i из слоя l как задачу нахождения входного изображения x, которое максимизирует функцию активации $a_i^l(\Theta, x)$. Таким образом, оптимизационная задача запишется следующим образом:

$$x^* = \arg\max_x a_i^l(\Theta, x). \tag{1}$$

Поставленную задачу назовем задачей *максимизации активации* (AM) или *визуализации признаков* (Feature Visualization). Подобная постановка задачи (1) впервые была предложена в работе [14] и в дальнейшем широко использовалась для визуализации ИНС [17]. Однако ее прямое решение зачастую ведет к неинтерпретируемым входным изображениям, состоящим преимущественно из высокочастотного шума [18], поэтому исследователями были предложены модификации – добавление регуляризации и ограничений в задачу для получения более интерпретируемых результатов.

2.1.1. Регуляризация в задаче максимизации активации. Для того, чтобы решить задачу в пространстве “реалистичных изображений”, могут применяться регуляризация по L^2 норме [18], размытие с гауссовым ядром [19], создание датасета с кусочками реальных изображений (patch dataset) [20], ограничения на вариацию функции (total variation, TV) [21], использование цен-



Фиг. 1. Примеры изображений, максимизирующих активации соответствующих классов, изображения взяты из обзора методов AM [16].

трального смещения (center bias) [22], инициализация средним изображением (mean image initialization) [22] и другие.

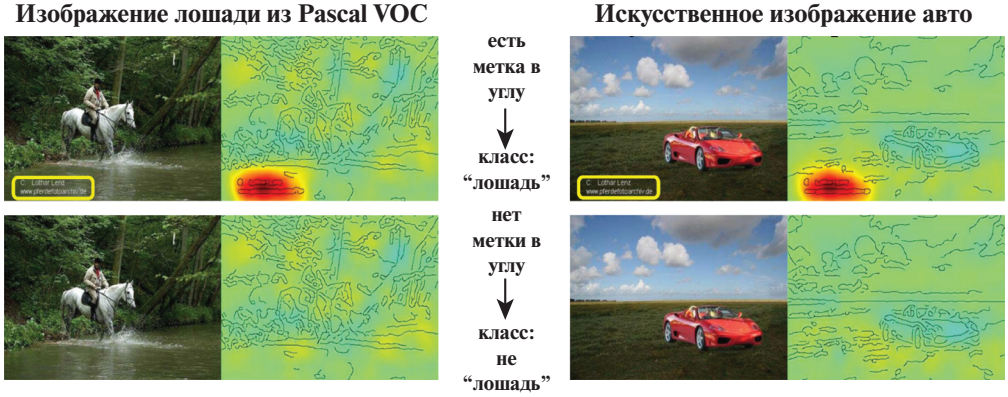
Таким образом, после добавления регуляризационного члена $R(x)$ задача максимизации активации имеет вид

$$x^* = \arg\max_x (a(x) - R(x)). \tag{2}$$

2.1.2. Генеративные сети в задаче максимизации активации. Генеративные сети [23] позволяют создавать изображения из переменных латентного пространства (Latent Space). Эти изображения можно использовать для поиска реалистичных изображений, максимизирующих активацию. Такой метод рассмотрен в работе [17], где генератор G ищет код в скрытом подпространстве $h \in R^{4096}$ такой, что изображение $G(h)$ максимизирует активацию ИНС $a(G(h))$. Соответственно задача AM из формы, записанной в уравнении 2, преобразуется в следующую:

$$h^* = \arg\max_h (a(G(h)) - R(h)). \tag{3}$$

Отметим, что подход с использованием генеративных сетей действительно позволяет во многих случаях находить реалистичные входные изображения. Найденные решения задачи AM для ряда методов представлены на фиг. 1.



Фиг. 2. Пример атрибуции тепловых карт из работы SpRAy [24].

2.1.3. Области применения метода максимизации активации. Метод AM допускает широкий спектр возможных приложений в области DL и EDL (см., например, [16]), включая следующие:

- визуализация выходных параметров для новых задач;
- визуализация внутренних параметров ИНС;
- синтезирование изображений, активирующих несколько нейронов;
- наблюдение эволюции нейрона во время обучения;
- синтезирование видео;
- использование максимизации активации как инструмента отладки;
- синтезирование изображений на основе описания;
- синтезирование изображений на основе маски семантической сегментации;
- синтезирование preferred stimuli для реального, биологического мозга.

2.2. Атрибуция

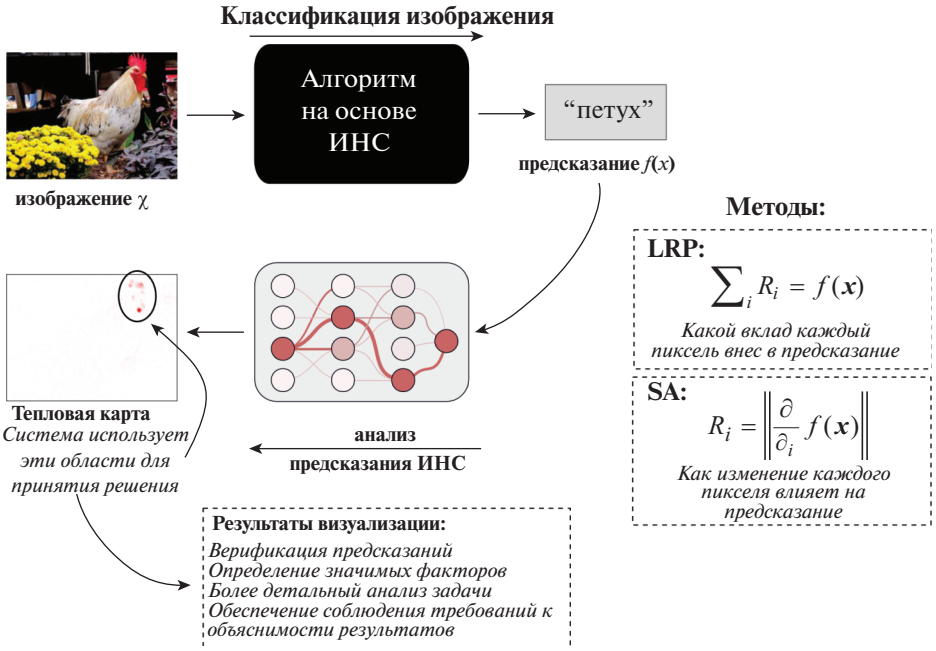
Атрибуция (Attribution, Heatmapping, Spectral Relevance Analysis) изучает, какая часть входного тензора ИНС, чаще всего какая область изображения, отвечает за активацию определенного нейрона сети. Так, например, для задачи детектирования объектов под атрибуцией обычно понимается построение тепловой карты вклада в детектирование данного объекта каждой точки на входном изображении (см. пример на фиг. 2).

Построение подобных тепловых карт проливает свет на процесс принятия решения алгоритмом компьютерного зрения. Так, в работе SpRAy [24] приводятся примеры, когда ИНС на наборе данных PASCAL VOC [25] обучилась находить подпись внизу картинки. По стечению обстоятельств подписи присутствовали на многих картинках с лошадьми, в итоге подобные примеры в литературе обрели название “Умный Ганс” (Clever Hans) в честь знаменитой лошади начала прошлого века.

Классическим методом построения тепловой карты атрибуции является аддитивный метод – Layer-wise Relevance Propagation (далее LRP) [26]. Пусть $x = (x_1, x_2, \ldots, x_d)$- входной вектор, а как $f(x)$ – выход ИНС. Тогда метод LRP заключается в нахождении такого вектора $R = (R_1, R_2, \ldots, R_d)$, имеющего такую же длину, как и входной вектор x, что верно следующее:

$$\sum_{p=1}^{d} R_p = f(x).$$

Схожий вид визуализации ИНС – анализ чувствительности (Sensitivity Analysis) [27], [28]. В рамках данного метода предполагается решение задачи сопоставления входному изображению тепловой карты, отображающей для каждого пикселя меру того, насколько изменится выход заданного нейрона при изменении значения в данной точке изображения. В отличие от методов анализа чувствительности, в методах атрибуции и разобранном выше методе LRP нас интересует само значение выхода, а не его локальное изменение (см. рис. 3 из работы [27]). Пусть индексы i



Фиг. 3. Описание процесса анализа и визуализации работы ANN на основе метода атрибуции LRP и на основе метода анализа чувствительности. Видно различие в данных двух задачах. Основано на статье [27].

и j кодируют номер нейронов на двух последовательных слоях, тогда $a_j = \sigma\left(\sum_i (a_i w_{ij} + b)\right)$ – активация на j слое. Здесь $\sum_i$ означает суммирование по всем нейронам i-го слоя, а $\sum_j$ – суммирование по всем нейронам j-го слоя. В этом случае метод распространения ошибки (propagation of LRP) можно записать в виде

$$R_i = \sum_j \frac{z_{ij}}{\sum_i z_{ij}} R_j, \tag{4}$$

где z_{ij} – вклад нейрона i в активацию a_j. Чаще всего этот вклад зависит от активации a_i и веса w_{ij}. Последовательно применяя данный метод обратного распространения ошибки, начиная с выхода ИНС и продолжая до входного изображения, можно получить веса для каждой компоненты входного ветора-картинки и отобразить результат в виде тепловой карты. Пример такой тепловой карты можно видеть на фиг. 3. Как можно видеть, подобная визуализация понятна и легко поддается анализу.

Кратко отметим также другие алгоритмы, решающие задачу атрибуции.

• DeepLIFT [29] – в рамках данного метода сравнивается активация каждого нейрона с некоторым “референсным” значением и присваивается значение “вклада” в активацию исследуемого выхода на основе их разницы.

• Guided Backpropagation (Guided BackProp) [30] – метод, основанный на методе обратного распространения ошибки (backpropagation) с той разницей, что отрицательные значения заменяются на 0, а также не используется информация от нейронов с отрицательным выходным значением.

• Integrated Gradients [31] – метод, основанный на интегрировании всех градиентов (из стандартного метода обратного распространения ошибки) вдоль “пути” от входного значения до выходного.

• Smooth Grad [32] – данный подход представляет модернизацию методов атрибуции, основанных на подсчете градиента, при помощи добавления сглаживания.

• Class Activation Mapping (CAM) [33] – метод, основанный на применении слоев глобального среднего (global average pooling, GAP) в сверточных нейронных сетях (Convolutional Neural Networks, далее CNN) для построения тепловой карты значимости пикселей входного изображения.

• Gradient-Weighted Class Activation Mapping (Grad-CAM) [34] – метод, основанный на идее CAM с добавлением информации о градиенте для потока информации, связанного с активацией нейрона предсказанного класса.

• Score-Weighted Class Activation Mapping (Score-CAM) [35] – модернизация метода Grad-CAM – вместо информации о градиентах используется специально введенная мера "увеличения уверенности", наподобие той, что используется в DeepLIFT.

• SHAPley Additive exPlanations (SHAP) [36] – метод, расширяющий идею аддитивной атрибуции признаков при помощи теоретико-игрового анализа. Может применяться для анализа важности признаков в широком спектре ML задач.

• Saliency Map (SM) [18] – метод, вычисляющий локальную чувствительность на основе частных производных. Позволяет, например, оценивать, для каких входных пикселей возмущения влияют на изменение финальной категории для изображения. Данный метод применим для достаточно общих типов архитектур нейронных сетей с дифференцируемыми входами.

• Deconvolution (DeConv) [37] – метод, основанный на построении CNN g с выходами в виде другой CNN f. Сеть g конструируется так, чтобы "обратить" операции, выполняемые исходной сетью f. Например, для операции свертки применяются транспонированные версии исходных фильтров с некоторыми оговорками. При этом деконволюционная сеть использует ReLU в качестве активационной функции каждый раз, тем самым приравнивая к нулю все возникающие отрицательные значения.

2.3. Визуализация функционала потерь

В наиболее общей форме ИНС может быть представлена как векторная функция $f(x,\theta)$, где x – это входной вектор, θ – набор параметров сети, а значение функции f – это соответствующее предсказание ИНС. Подстройка параметров θ производится на обучающем наборе данных (x_i, y_i), $i = 1,2,\ldots,N$, посредством минимизации функционала потерь:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} l(x_i, y_i, \theta),$$

где функционал потерь отдельного обучающего примера может, например, иметь простейшую квадратичную форму:

$$l(\theta) = \|f(x,\theta) - y\|^2.$$

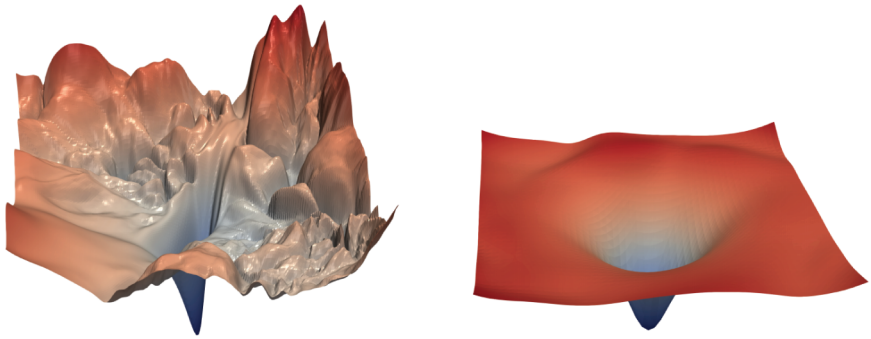
В общем случае функционал потерь является невыпуклой функцией, зависящей от огромного числа переменных (параметров ИНС), а его минимизация представляет сложнейшую вычислительную задачу [38]. Классический стохастический метод градиентного спуска позволяет с минимальными вычислительными затратами осуществлять итерационный процесс поиска локального минимума функционала потерь. Принципиальную важность здесь представляет форма функционала потерь, определяемая выбором гиперпараметров ИНС. Чем более гладким оказывается функционал потерь, тем быстрее будет происходить обучение и тем лучше окажется обобщающая способность ИНС.

Перспективным средством анализа профиля функционала потерь может быть визуализация. Удачная визуализация позволяет оценить общие характерные черты выбранного функционала, а также их изменение при соответствующей модификации гиперпараметров сети (см. пример на фиг. 4). Однако здесь возникают сложности, характерные для задачи представления существенно многомерных функций на одно- и двумерных графиках.

В литературе наибольшее распространение получили два способа визуализации функционала потерь – графики одномерной линейной интерполяции [40], [41] и контурные графики по случайным направлениям [42], [43]. В первом случае выбираются два различающихся значения вектора параметров ИНС: θ и θ', между которыми производится интерполяция функционала потерь по линейной формуле:

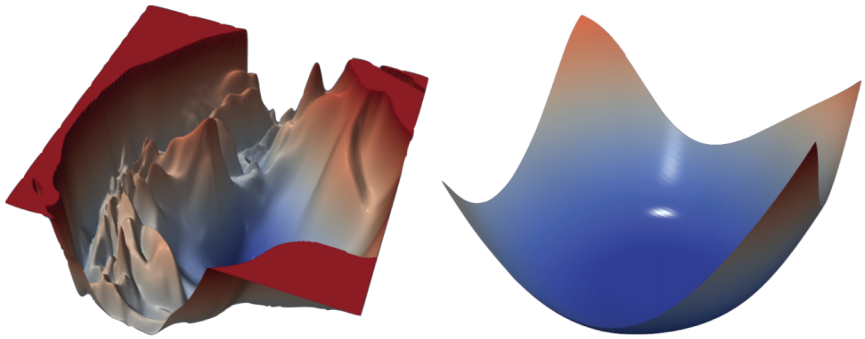
$$\phi(\alpha) = L(\theta(\alpha)), \quad \theta(\alpha) = (1-\alpha)\theta + \alpha\theta', \quad 0 \le \alpha \le 1,$$

(а) без skip connections (б) со skip connections



Фиг. 4. Профиль функционала потерь для ИНС типа ResNet-56, построенный в работе [39] с использованием оригинальной схемы нормализации.

(а) ResNet-110 без skip connections (б) DenseNet, 121 слой



Фиг. 5. Профиль функционала потерь для ИНС типа ResNet-110 и DenseNet на наборе данных CIFAR-10, построенный в работе [39] с использованием оригинальной схемы нормализации.

с последующим построением графика одномерной функции $\phi(\alpha)$. Во втором случае выбираются некоторая центральная точка θ^* и два вектора направлений δ и η, а затем строится одномерная функция:

$$\phi(\alpha) = L(\theta^* + \alpha\delta),$$

либо двумерная функция:

$$\phi(\alpha, \beta) = L(\theta^* + \alpha\delta + \beta\eta).$$

На фиг. 4 и 5 приводятся результаты по визуализации функционала потерь в окрестности обнаруженного при обучении локального минимума, полученные в работе [39] на основе обобщения метода контурных графиков. Как можно видеть, данный метод представляет эффективный инструмент для оценки итогового качества обучения ИНС и сравнения эффекта от выбора гиперпараметров и различных эвристик обучения.

3. ПРОГРАММНЫЕ РЕАЛИЗАЦИИ МЕТОДОВ ВИЗУАЛИЗАЦИИ

Рассмотрим наиболее популярные программные реализации алгоритмов визуализации ИНС. Для выявления релевантных программных продуктов авторами осуществлялись различные формы предметных запросов в поисковой системе Google, а также проводился поиск непосредственно по тегам репозиториев на Github, при этом в результаты поиска мы не добавляли программные продукты, с момента последнего обновления которых прошло более 12 мес. В итоге нами было отобрано 16 программных решений (библиотек), которые представлены в табл. 1. Для каждой библиотеки мы указываем поддерживаемый фреймворк ML (в рамках библиотеки осуществляется визуализация моделей, построенных с использованием только соответствующего

Таблица 1. Программные продукты для визуализации ИНС

Название программы	Фреймворк ML	Звезды Github	Последнее обновление
OpenAI Microscope	–	закрытый код	веб-интерфейс
SHAP	TensorFlow, PyTorch	11000	2020, ноябрь
Playground-TensorFlow	TensorFlow	9500	2020, апрель
TensorboardX	PyTorch	6700	2020, июль
Tensorboard	TensorFlow	5100	2020, ноябрь
PyTorch-CNN-visualizations	PyTorch	4900	2020, сентябрь
Lucid	TensorFlow	4000	2020, ноябрь
Keras-vis	TensorFlow (Keras)	2800	2020, апрель
Captum	PyTorch	1900	2020, ноябрь
PyTorch-grad-cam	PyTorch	1700	2020, апрель
Hiddenlayer	TensorFlow, PyTorch	1300	2020, апрель
TF-explain	TensorFlow	740	2020, июль
INNvestigate	TensorFlow	726	2020, октябрь
Saliency	TensorFlow	600	2020, октябрь
FlashTorch	PyTorch	507	2020, май
TCAV	TensorFlow	414	2020, июль

фреймворка; при этом, как можно видеть из полученных результатов, выбор в итоге осуществляется между двумя популярными фреймворками – TensorFlow [44] и PyTorch [45]), количество звезд на Guthub и дату (год и месяц) последнего обновления репозитория проекта. Для простоты и компактности мы приводим округленное количество звезд на Guthub в качестве меры интереса научного сообщества к соответствующему программному продукту. Безусловно, для более точной оценки необходим также учет даты создания репозитория, отдельное ранжирование программных продуктов, использующих различные фреймворки ML и т.д. Однако подобное рассмотрение выходит за рамки задач данного обзора. Отметим, что последняя актуализация данных осуществлялась нами в начале декабря 2020 г.

Программный продукт OpenAI Microscope (https://openai.com/blog/microscope) позволяет осуществлять визуализацию (признаков отдельных слоев и набора обучающих данных) для восьми популярных в задачах машинного зрения архитектур CNN (AlexNet, AlexNet (Places), Inception v1, Inception v1 (Places), VGG 19, Inception v3, Inception v4 и ResNet v2 50) на специализированном интерактивном веб-сайте. Данный продукт имеет закрытый исходный код (известно лишь, что визуализации подготавливались с использованием библиотеки Lucid, которая будет рассмотрена ниже) и, на наш взгляд, может быть использован лишь в учебных или иллюстративных целях.

Популярная библиотека SHAP (https://github.com/slundberg/SHAP) (SHAPley Additive exPlanations) реализует универсальный подход [36], основанный на теоретико-игровом анализе и расширяющий идею аддитивной атрибуции признаков, для интерпретации и последующей визуализации широкого класса моделей ML. Помимо оригинального алгоритма SHAP, в данной библиотеке реализованы также методы DeepLIFT и Smooth-Grad.

Интерактивная браузерная среда Playground-Tensor Flow (https://github.com/tensorflow/playground) предполагает использование в образовательных целях для визуализации процесса построения и обучения ИНС. Пользователь имеет возможность выбрать ряд гиперпараметров полносвязной сети (количество слоев, количество нейронов в слое, тип функции активации и параметры регуляризации), тип и степень зашумленности набора данных, а также используемые входные признаки из ограниченного набора возможных вариантов. После запуска процесса обучения сети производится интерактивная демонстрация эволюции весов связей нейронов и точности предсказания.

Библиотека Tensorboard (https://github.com/tensorflow/Tensorboard) для TensorFlow и ее адаптация TensorboardX (https://github.com/lanpa/TensorboardX) для PyTorch представляют популярный инструмент визуализации архитектуры ИНС и процесса обучения в режиме реального времени.

Библиотека PyTorch-CNN-visualizations (https://github.com/utkuozbulak/PyTorch-CNN-visualizations) предназначена для визуализации глубоких CNN, построенных и обученных с использованием фреймворка PyTorch. В рамках данной библиотеки реализован широкий спектр алгоритмов EDL, включая SM, CAM, Grad-CAM, Score-CAM, Guided BackProp, DeConv, Deep-dream, Smooth-Grad и др.

Библиотека Lucid (https://github.com/tensorflow/Lucid) содержит обширную подборку EDL методов, включая различные методы визуализации признаков, сетки активации, метод пространственной атрибуции и др. Отметим, что для работы библиотеки необходим фреймворк TensorFlow, причем поддержка современной 2-й версии пока что отсутствует.

Отметим также библиотеку Keras-vis (https://github.com/raghakot/Keras-vis), которая предоставляет набор инструментов для визуализации сверточных и полносвязных слоев ИНС на основе методов AM, SM и CAM.

Активно развивающаяся библиотека Captum (https://github.com/pytorch/Captum) реализует широкий набор методов атрибуции для интерпретации нейросетевых моделей, построенных с PyTorch, включая такие методы, как SM, DLIFT, SHAP, Grad-CAM, Guided BackProp, DeConv, Integrated gradients и многие другие. Отметим, что возможна также установка интерактивного браузерного интерфейса “Captum Insights” для визуализации результатов.

Библиотека PyTorch-grad-cam (https://github.com/jacobgil/PyTorch-grad-cam) реализует метод атрибуции Grad-CAM для моделей, созданных с PyTorch. Отметим, что существует версия данной библиотеки для TensorFlow, однако она несколько лет не обновлялась и в этой связи не включена в наш перечень.

Библиотека Hiddenlayer (https://github.com/waleedka/Hiddenlayer) реализует функционал, близкий к продукту Tensorboard и предоставляет набор инструментов для визуализации как графа ИНС с возможностью кастомизации, так и процесса ее обучения, включая эволюцию функционала стоимости, весов и активаций нейронов слоев сети и т.п. Особо отметим, что данная библиотека может использоваться и для моделей, обученных с TensorFlow, и с PyTorch.

Библиотека TF-explain (https://github.com/sicara/TF-explain) реализует ряд методов визуализации для моделей, построенных с использованием TensorFlow, включая Grad-CAM, Integrated gradients, Smooth-Grad, а также ряд базовых методов визуализации активаций и градиентов.

Библиотека iNNvestigate (https://github.com/albermax/innvestigate) содержит реализации ряда методов визуализации ИНС, включая Smooth-Grad, Guided BackProp, DeConv, LRP и Integrated gradients, DLIFT. Для работы библиотеки необходим фреймворк TensorFlow первой версии.

Библиотека Saliency (https://github.com/PAIR-code/saliency) реализует множество методов из класса SM, включая Smooth-Grad, Guided BackProp, Integrated gradients, Grad-CAM и XRAI [46] и осуществляет визуализацию карт значимости для моделей, обученных с использованием TensorFlow.

Библиотека FlashTorch (https://github.com/MisaOgura/FlashTorch) позволяет осуществлять визуализацию ИНС, созданных с использованием фреймворка PyTorch, посредством методов AM и SM. Отметим, что данная небольшая библиотека имеет простой понятный интерфейс и подробные инструкции в различных форматах (текст, видео, демонстрационные примеры).

Специализированная библиотека TCAV (https://github.com/tensorflow/TCAV) (Testing with Concept Activation Vectors, см. также работу [47]) позволяет выявить наиболее важные сложные признаки (не значения отдельных пикселей, ацвет, пол, раса и т.п.), влияющие на выбор класса при предсказании ИНС. Данная библиотека работает с уже обученными TensorFlow моделями и требует для своего функционирования набор примеров, демонстрирующих сложные признаки. В качестве вывода отображается график, иллюстрирующий степень важности выбранных сложных признаков для рассматриваемого варианта предсказания сети (например, на сколько “полосатость” и “зигзагообразность” влияют на выбор сетью класса “зебра”).

4. АНАЛИЗ СУЩЕСТВУЮЩИХ ОБЗОРОВ

На сегодняшний день в литературе представлен ограниченный набор обзоров, затрагивающих в той или иной степени рассматриваемую нами тематику визуализации ИНС. В табл. 2 приведены выявленные авторами релевантные обзорные работы, а ниже кратко приведено обсуждение каждой из этих работ с указанием характерных особенностей и степени соответствия теме.

Таблица 2. Обзоры по теме визуализации ИНС

Ссылка	Год	Заголовок	Журнал/Книга
[48]	2017	Towards better analysis of machine learning models: A visual analytics perspective	Visual Informatics
[49]	2017	Visualizations of deep neural networks in computer vision: A survey	Transparent Data Mining for Big and Small Data
[50]	2018	A user-based taxonomy for deep learning visualization	Visual Informatics
[51]	2018	Visual interpretability for deep learning: a survey	Frontiers of Informat. Technology & Electronic Engineering
[52]	2018	How convolutional neural network see the world – A survey of convolutional neural network visualization methods	ArXiv Preprint
[53]	2018	Visual analytics for explainable deep learning	IEEE Computer Graphics and Applications
[54]	2018	Visual analytics in deep learning: An interrogative survey for the next frontiers	IEEE Transactions on Visualization and Computer Graphics
[55]	2018	A task-and-technique centered survey on visual analytics for deep learning model engineering	Computers & Graphics
[16]	2019	Understanding neural networks via feature visualization: A survey	Explainable AI: Interpreting, Explaining and Visualizing Deep Learning
[56]	2020	A survey of visual analytics techniques for machine learning	Computational Visual Media
[57]	2020	A survey of surveys on the use of visualization for interpreting machine learning models	Informat. Visualization

В статье [48] приводится краткий обзор методов и перспектив визуализации нейросетевых структур. Данная работа не может рассматриваться как полноценный обзор, однако она представляет определенный интерес для исследователей и в этой связи включена в перечень.

В работе [49] впервые, на наш взгляд, предлагается системный подход к задаче визуализации ИНС, формулируется собственная многоуровневая схема классификации, включающая цель и метод визуализации, архитектуру ИНС и область ее применения в ML, а также набор данных, на котором обучалась ИНС. Для каждого из этих пяти критериев вводится набор возможных значений, что позволяет авторам эффективно типизировать все отобранные научные работы, а также провести соответствующий сравнительный анализ популярности различных направлений и т.п. Данная работа безусловно обладает высокой научной ценностью, однако представленные в ней методы визуализации и перечень обсуждаемых публикаций на сегодняшний день являются не вполне актуальными.

В обзоре [50] формулируется система классификации методов визуализации ИНС с точки зрения конечного заинтересованного лица (“начинающие”, “практики”, “разработчики”, “эксперты”). В зависимости от типа заинтересованного лица уточняется специализация инструмента визуализации и далее в рамках этой специализации осуществляется краткое обсуждение нескольких практических реализаций подобных систем. В данной работе содержится описание ряда практически значимых программных продуктов, однако малый объем работы и отсутствие в ней описания алгоритмов и методов визуализации ИНС не позволяют рассматривать ее как полноценный масштабный обзор.

В обзоре [51] рассматриваются общие вопросы EDL в контексте CNN и обсуждаются различные методы для интерпретации нейросетевых моделей, включая их представление в форме графов и решающих деревьев. Таким образом, данный обзор не вполне соответствует рассматриваемой нами тематике, хотя и содержит ряд важных положений по общей задаче построения интерпретируемых моделей.

В обзоре [52], посвященном визуализации непосредственно CNN, формулируется ряд методов, включая AM, обращение сети (Network Inversion), деконволюционные нейронные сети (DeConv) и метод “рассечения” сети (Network Dissection). Для каждого метода обсуждается структура, алгоритм, соответствующие операции и результаты из научных публикаций, при этом

особый акцент делается на EDL. Основной вывод, к которому приходят на основе проведенного анализа – это иерархический характер организации CNN (каждый последующий слой отвечает за распознавание все более сложных признаков), имеющий определенную аналогию с механизмом действия зрительной коры человека. Отметим, что данная работа отличается высоким уровнем описания математических постановок методов визуализации. Однако рассмотрение в этом обзоре сосредоточено на одном типе нейросетевых архитектур, также с момента публикации данного обзора появился ряд новых релевантных публикаций.

Работа [53] может лишь условно рассматриваться как обзорная. В данной публикации обсуждаются общие аспекты EDL и особая роль визуализации в этой глобальной задаче. В работе приводится перечень основных задач EDL и визуализации ИНС, а также демонстрируются некоторые программные реализации систем визуализации ИНС. Таким образом, в данной публикации отражен ряд важных задач и прикладных направлений визуализации, однако отсутствует детальный обзор методов визуализации и соответствующих результатов.

В обзоре [54] предлагается оригинальный подход к классификации работ по теме визуализации ИНС, в основе которого лежит идея интегрального рассмотрения связанных вопросов: *почему* проводится визуализация (*why*); *кто* хочет осуществлять визуализацию (*who*); *что* визуализируется (*what*); *как* реализуется визуализация (*how*); *когда*, т.е. на каком этапе работы модели ML, производится визуализация (*when*); *где* используется визуализация (*where*). Для каждого из представленных вопросов в работе предлагаются варианты ответов с соответствующими подробными комментариями. Построенная таким образом система позволила осуществить аккуратную визуальную классификацию анализируемых ими научных работ. На наш взгляд, данный обзор является одним из наиболее обширных и в то же время глубоких из представленных в табл. 2. Однако предложенная система классификации в определенных аспектах представляется несколько искусственной (в отличие, например, от классификации, предложенной в работе [49]). Также данный обзор не охватывает новые научные результаты, полученные за 2019 и 2020 г.

В обзоре [55] для формализации задачи классификации методов и публикаций по визуализации ИНС вводится система из трех категорий, соответствующих цели визуализации: понимание нейросетевой архитектуры; улучшение процесса тренировки сети; выявление значимых входных признаков. На наш взгляд, данная классификация не является универсальной, однако в этом обзоре обсуждается ряд практических реализаций систем визуализации.

В работе [16] рассматриваются возможные реализации и применения одного конкретного метода визуализации ИНС – метода AM, строятся различные формулировки данного метода и описывается ряд современных работ, в которых используется метод AM для визуализации ИНС. Отметим, что в данной работе также приводятся рассуждения, вскрывающие связь максимизации активации с задачей анализа активности головного мозга в нейронауках. Поскольку авторы сосредоточились на обсуждении лишь одного метода, то данная работа не может рассматриваться как полноценный обзор по методам визуализации ИНС.

В работе [56] выполнен обширный обзор публикаций по теме методов визуализации в ML. Для организации анализируемых научных работ предложена классификация по этапу процесса подготовки и обучения модели ML (соответствует вопросу *when* в контексте рассмотренного выше обзора [54]) и далее по цели визуализации (соответствует вопросу *why* в обзоре [54]). Соответственно рассматривается визуализация: до построения модели (для улучшения качества исходных данных или качества входных признаков); в процессе построения модели (для понимания модели или для диагностики модели, или для “ручного управления” моделью); после построения модели (для понимания статического или динамического распределения результатов работы модели). Авторы приводят кривые тренда публикационной активности по соответствующим направлениям классификации, согласно которым наибольший рост соответствует направлениям диагностики и “ручного управления” моделью, при этом направление, связанное с пониманием модели, характеризуется восходящим по годам трендом с незначительным снижением активности в 2020 г. Проведенный в работе анализ публикаций позволяет судить об общих направлениях развития направления визуализации в области ML, однако в этой работе не приводятся описания алгоритмов и технические подробности реализации соответствующих подходов и систем визуализации.

Особо отметим работу [57], позиционируемую как “обзор обзоров” в области визуализации для EDL. Авторы данной работы детально описывают использованную ими интеллектуальную стратегию поиска научных работ, в результате которой был построен список из 18 публикаций, являющихся обзорами. Затем авторы проводят анализ близости и релевантности отобранных ра-

бот в контексте совпадения внешних ссылок и др., на основе которого производится их последующее краткое обсуждение. Отметим, что в данном "метаобзоре" предложены интересная методология поиска и оценки близости научных работ, однако в нем отсутствует подробное обсуждение содержания работ и методов. Также отобранные авторами научные обзоры соответствуют не только лишь теме визуализации ИНС, но и более общим областям, связанным с предиктивной аналитикой и визуализацией больших объемов данных.

5. ЗАКЛЮЧЕНИЕ

В работе приведен обзор современных методов визуализации искусственных нейронных сетей, включающий методы максимизации активации, атрибуции и визуализации функционала потерь. Кроме непосредственного обзора ключевых алгоритмов для каждой подзадачи визуализации, в работе построен подробный перечень релевантных программных пакетов с практическими реализациями алгоритмов и рассмотрены уже представленные в литературе обзорные работы.

Как следует из проведенного анализа алгоритмов, программного обеспечения и обзорных работ, научное направление, связанное с визуализацией искусственных нейронных сетей, на сегодняшний день является актуальным и бурно развивающимся. При этом существует ряд потенциальных новых приложений данной методологии в современных задачах по исследованию естественных нейронных сетей и формированию в них памяти и специализации нейронов.

СПИСОК ЛИТЕРАТУРЫ

1. *LeCun Y., Bengio Y., Hinton G.* Deep learning // Nature. 2015. V. 521. Issue 7553. P. 436–444.
2. *Shahid N., Rappon T., Berta W.* Applications of artificial neural networks in health care organizational decision-making: A scoping review // PLoS ONE. 2019. V. 14. Issue 2.
3. *Nassif A.B., Shahin I., Attili I., Azzeh M., Shaalan K.* Speech recognition using deep neural networks: A systematic review // IEEE Access. 2019. V. 7. P. 19143–19165.
4. *Alkinani H.H., Al-Hameedi A.T.T., Dunn-Norman S., Flori R.E., Alsaba M.T., Amer A.S.* Applications of artificial neural networks in the petroleun industry: A review // SPE Middle East Oil and Gas Show and Conference, Society of Petroleum Engineers. 2019.
5. *Tjoa E., Guan C.* A survey on explainable artificial intelligence (xai): Toward medical xai // Proc. of the IEEE Transactions on Neural Networks and Learning Systems. 2020.
6. *Xu F., Uszkoreit H., Du Y., Fan W., Zhao D., Zhu J.* Explainable ai: A brief survey on history, research areas, approaches and challenges // CCF Internat. Conference on Natural Language Proc. and Chinese Comput. 2019. P. 563–574.
7. *Samek W., Montavon G., Vedaldi A., Hansen L.K., Müller K.-R.* Explainable AI: interpreting, explaining and visualizing deep learning // Nature. 2019. V. 11700.
8. *Lipton Z.C.* The mythos of model interpretability // Queue. 2018. V. 16. Issue 3. P. 31–57.
9. *Cowan N.* The many faces of working memory and short-term storage // Psychonomic Bulletin Review. 2017. V. 24. Issue 4. P. 1158–1170.
10. *Anokhin K., Ivashkina O., Toropova K., Gruzdeva A., Rogozhnikova O.B., Plushnin V., Fedotov I.* Neuronal encoding of object-type and object-place memories in hippocampus and neocortex of young and old mice // The FASEB Journal. 2020. V. 34. Issue S1. P. 1–1.
11. *Zhigulina P., Ushakov V., Kartashov S., Malakhov D., Orlov V., Novikov K., Korotkova A., Anokhin K., Nourkova V.* The architecture of neural networks for enhanced autobiographical memory access: a functional mri study // Procedia Computer Science. 2020. V. 169. P. 787–794.
12. *Tiunova A.A., Komissarova N.V., Anokhin K.V.* Mapping the neural substrates of recent and remote visual imprinting memory in the chick brain // Frontiers in Physiology. 2019. V. 10. P. 351–351.
13. *Marshel J.H., Kim Y.S., Machado T.A., Quirin S., Benson B., Kadmon J., Raja C., Chibukhchyan A., Ramakrishnan C., Inoue M.* Cortical layer-specific critical dynamics triggering perception // Science. 2019. V. 365. Issue 6453.
14. *Erhan D., Bengio Y., Courville A., Vincent P.* Visualizing higher-layer features of a deep network, Technical Report // ICML 2009 Workshop on Learning Feature Hierarchies, Montréal, Canada. 2009.
15. *Hubel D., Wiesel T.* Receptive fiedls of single neurones in the cat's striate cortex // J. of Physiology. 1959. V. 148.
16. *Nguyen A., Yosinski J., Clune J.* Understanding neural networks via feature visualization: A survey // Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. 2019. P. 55–76.
17. *Nguyen A., Dosovitskiy A., Yosinski J., Brox T., Clune J.* Synthesizing the preferred inputs for neurons in neural networks via deep generator networks // Advances in Neural Informat. Processing Systems. 2016. P. 3395–3403.

18. *Simonyan K., Vedaldi A., Zisserman A.* Deep inside convolutional networks: Visualising image classification models and saliency maps // Workshop at Internat. Conference on Learning Representations. 2014.
19. *Yosinski J., Clune J., Nguyen A., Fuchs T., Lipson H.* Understanding Neural Networks Through Deep Visualization // Deep Learning workshop at ICML 2015. 2015.
20. *Wei D., Zhou B., Torrabla A., Freeman W.* Understanding Intra-Class Knowledge Inside CNN // arXiv:1507.02379, 2015, url: https://arxiv.org/abs/1507.02379
21. *Mahendran A., Vedaldi A.* Visualizing deep convolutional neural networks using natural pre-images // Internat. Journal of Computer Vision. 2016. V. 120. Issue 3. P. 233–255.
22. *Nguyen A., Yosinski J., Clune J.* Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks // arXiv:1602.03616, 2016, url: https://arxiv.org/abs/1602.03616
23. *Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.* Generative adversarial nets // Proc. of the 27th Internat. Conference on Neural Informat. Proc. Systems. 2014. V. 2. P. 2672–2680.
24. *Lapuschkin S., Wäldchen S., Binder A., Montavon G., Samek W., Müller K.-R.* Unmasking clever hans predictors and assessing what machines really learn // Nature Communications. 2019. V. 10. Issue 3.
25. *Everingham M., Eslami S.M.A., Van Gool L., Williams C.K.I., Winn J., Zisserman A.* The pascal visual object classes challenge: A retrospective // Internat. Journal of Computer Vision. 2015. V. 111. Issue 1. P. 98–136.
26. *Lapuschkin S., Binder A., Montavon G., Klauschen F., Müller K.-R., Samek W.* On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation // PLoS ONE. 2015. V. 10.
27. *Samek W., Wiegand T., Müller K.-R.* Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models // ITU Journal: ICT Discoveries. 2019. V. 1. P. 39–48.
28. *Samek W., Binder A., Montavon G., Lapushckin S., Müller K.-R.* Evaluating the visualization of what a deep neural network has learned // IEEE Transactions on Neural Networks and Learning Systems. 2017. V. 28. Issue 11. P. 2660–2673.
29. *Shrikumar A., Greenside P., Kundaje A.* Learning important features through propagating activation differences // Proc. of the 34th Internat. Conference on Machine Learning, PLMR. 2017. P. 3145–3153.
30. *Springenberg J.T., Dosovitskiy A., Brox T., Riedmiller R.* Striving for simplicity: The all convolutional net // arXiv:1412.6806, 2014, url: https://arxiv.org/abs/1412.6806
31. *Sundararajan M., Taly A., Yan Q.* Axiomatic attribution for deep networks // Proc. of the Internat. Conference on Machine Learning, ICML. 2017. P. 3319–3328.
32. *Smilkov D., Thorat N., Kim B., Viégas F., Wattenberg M.* Smoothgrad: removing noise by adding noise // Workshop on Visualization for Deep Learning, ICML. 2017.
33. *Zhou B., Khosla A., Lapedriza A., Oliva A., Torralba A.* Learning deep features for discriminative localization // Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. P. 2921–2929.
34. *Selvaraju R.R., Cogswell M., Das A., Vedantam R., Parikh D., Batra D.* Grad-cam: Visual explanations from deep networks via gradient-based localization // Proc. of the IEEE Internat. Conference on Computer Vision. 2017. P. 618–626.
35. *Wang H., Wang Z., Du M., Yang F., Zhang Z., Ding S., Mardziel P., Hu X.* Score-cam: Score-weighted visual explanations for convolutional neural networks // Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020. P. 24–25.
36. *Lundberg S.M., Lee S.-I.* A unified approach to interpreting model predictions // Advances in Neural Informat. Processing Systems. 2017. P. 4765–4774.
37. *Zeiler M.D., Fergus R.* Visualizing and understanding convolutional networks // European Conference on Computer Vision. Springer. 2014. P. 818–833.
38. *Choromanska A., Henaff M., Mathieu M., Arous G.B., LeCun Y.* The loss surfaces of multilayer networks // Artificial Intelligence and Statistics. 2015. P. 192–204.
39. *Li H., Xu Z., Taylor G., Studer C., Goldstein T.* Visualizing the loss landscape of neural nets // Advances in Neural Informat. Processing Systems. 2018. P. 6389–6399.
40. *Dinh L., Pascanu R., Bengio S., Bengio Y.* Sharp minima can generalize for deep nets // Proc. of the 34th Internat. Conference on Machine Learning, PMLR. 2017. P. 1019–1028.
41. *Keskar N.S., Mudigere D., Nocedal J., Smelyanskiy M., Tang P.T.P.* On large-batch training for deep learning: Generalization gap and sharp minima // 5th Internat. Conference on Learning Representations, ICLR. 2017.
42. *Goodfellow I.J., Vinyals O., Saxe A.M.* Qualitatively characterizing neural network optimization problems // Internat. Conference on Learning Representations. 2015.
43. *Im D.J., Tao M., Branson K.* An empirical analysis of deep network loss surfaces // arXiv:1612.04010, 2016, https://arxiv.org/abs/1612.04010
44. *Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G.S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jia Y., Jozefowicz R., Kaiser L., Kudlur M., Levenberg J., Mané D., Monga R., Moore S., Murray D., Olah C., Schuster M., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viégas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., Zheng X.*

TensorFlow: Large-scale machine learning on heterogeneous systems // arXiv:1603.04467, 2016, url: https://arxiv.org/abs/1603.04467

45. *Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N., Antiga L., Desmaison A., Kopf A., Yang E., DeVito Z., Raison M., Tejani A., Chilamkurthy S., Steiner B., Fang L., Bai J., Chintala S.* Pytorch: An imperative style, high-performance deep learning library // Advances in Neural Informat. Processing Systems. 2019. V. 32. P. 8024–8035.
46. *Kapishnikov A., Bolukbasi T., Viégas F., Terry M.* Xrai: Better attributions through regions // Proc. of the IEEE Internat. Conference on Computer Vision. 2019. P. 4948–4957.
47. *Kim B., Wattenberg M., Gilmer J., Cai C., Wexler J., Viegas F., et al.* Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV) // Internat. conference on machine learning, PMLR. 2018. P. 2668–2677.
48. *Liu S., Wang X., Liu M., Zhu J.* Towards better analysis of machine learning models: A visual analytics perspective // Visual Informatics. 2017. V. 1. Issue 1. P. 48–56.
49. *Seifert C., Aamir A., Balagopalan A., Jain D., Sharma A., Grottel S., Gumhold S.* Visualizations of deep neural networks in computer vision: A survey // Transparent Data Mining for Big and Small Data. 2017. P. 123–144.
50. *Yu R., Shi L.* A user-based taxonomy for deep learning visualization // Visual Informatics. 2018. V. 2. Issue 3. P. 147–154.
51. *Zhang Q.-S., Zhu S.-C.* Visual interpretability for deep learning: a survey // Frontiers of Informat. Technology Electronic Engineering. 2018. V. 19. Issue 1. P. 27–39.
52. *Qin Z., Yu F., Liu C., Chen X.* How convolutional neural network see the world-a survey of convolutional neural network visualization methods // Mathematical Foundations of Computing. 2018. V. 1. Issue 2. P. 149–180.
53. *Choo J., Liu S.* Visual analytics for explainable deep learning // IEEE computer graphics and applications. 2018. V. 38. Issue 4. P. 84–92.
54. *Hohman F., Kahng M., Pienta R., Chau D.H.* Visual analytics in deep learning: An interrogative survey for the next frontiers // IEEE transactions on visualization and computer graphics. 2018. V. 25. Issue 8. P. 2674–2693.
55. *Garcia R., Telea A.C., da Silva B.C., Tørresen J., Comba J.L.D.* A task-and-techniquecentered survey on visual analytics for deep learning model engineering // Computers Graphics. 2018. V. 77. P. 30–49.
56. *Yuan J., Chen C., Yang W., Liu M., Xia J., Liu S.* A survey of visual analytics techniques for machine learning // Computational Visual Media. 2020.
57. *Chatzimparmpas A., Martins R.M., Jusufi I., Kerren A.* A survey of surveys on the use of visualization for interpreting machine learning models // Informat. Visualization. 2020. P. 6572–6583.