

*На правах рукописи*

Щербакова Елена Михайловна

**Матричные и тензорные разложения с условием  
неотрицательности и их применение**

Специальность 1.2.2 – Математическое моделирование, численные методы  
и комплексы программ

Диссертация на соискание ученой степени  
кандидата физико-математических наук

Научный руководитель:  
доктор физико-математических наук,  
профессор, академик РАН  
Е. Е. Тыртышников

# Содержание

<b>Введение</b>	<b>5</b>
<b>1 Неотрицательная факторизация матриц</b>	<b>13</b>
1.1 Сепарабельные матрицы . . . . .	13
1.2 Возмущенные сепарабельные матрицы . . . . .	19
1.3 Редуцированный алгоритм для симметричной неотрицательной факторизации . . . . .	28
1.4 Матрицы ранга 2 . . . . .	28
<b>2 Неотрицательная факторизация тензоров</b>	<b>30</b>
2.1 Неотрицательный тензорный поезд . . . . .	30
2.2 Алгоритм построения неотрицательного тензорного поезда NTTF . . . . .	31
2.3 Оценка точности аппроксимации, построенной методом NTTF	35
2.4 Алгоритм построения неотрицательного тензорного поезда NTT-MU . . . . .	36
2.5 Комбинация алгоритма NTT-MU и методики DMRG . . . . .	42
2.6 Построение тензорного поезда с неотрицательными элементами с помощью коррекции элементов . . . . .	45
2.7 Быстрый алгоритм построения неотрицательного канонического разложения на основе тензорного поезда . . . . .	50
2.8 Быстрый алгоритм построения неотрицательного разложения Таккера на основе тензорного поезда . . . . .	57
2.9 Чередующиеся проекции: общая идея . . . . .	59
<b>3 Применение неотрицательной тензорной факторизации для ускорения вычислений меры влияния узлов многомерной сети</b>	<b>61</b>
3.1 Теорема Фробениуса-Перрона для мультиоднородных отображений . . . . .	61

3.2	Пример $\mathcal{F} \in \text{MH}_d$ , определенного через неотрицательный тензор . . . . .	63
3.3	Применение неотрицательной тензорной факторизации . . .	67
3.4	Ошибка приближенного решения . . . . .	67
<b>4</b>	<b>Результаты численных экспериментов</b>	<b>73</b>
4.1	Комплекс программ . . . . .	73
4.2	Результаты численных экспериментов для неотрицательной матричной факторизации . . . . .	74
4.2.1	Проверка сложности редуцированного алгоритма для сепарабельных матриц . . . . .	74
4.2.2	Проверка сложности алгоритма для факторизации матриц ранга 2 . . . . .	75
4.2.3	Проверка сложности редуцированного алгоритма для симметричной неотрицательной факторизации . . . .	77
4.2.4	Редуцированный алгоритм для возмущенных сепарабельных матриц . . . . .	78
4.3	Результаты численных экспериментов для неотрицательной тензорной факторизации . . . . .	83
4.3.1	Результаты NTTF на искусственных данных . . . . .	83
4.3.2	Результаты NTTF и метода с коррекцией отрицательных элементов при аппроксимации ядра для уравнения Смолуховского . . . . .	84
4.3.3	Результаты методов неотрицательной тензорной факторизации для ERP (event related potential) тензора .	86
4.3.4	Анализ алгоритма NTT-MU (сложность, сходимость метода) . . . . .	87
4.3.5	Сравнение NTTF и NTT-MU с DMRG на искусственных данных . . . . .	89
4.3.6	Анализ алгоритма для построения неотрицательного разложения Таккера на основе тензорного поезда . .	90
4.3.7	Анализ алгоритма для построения неотрицательного канонического разложения на основе тензорного поезда	92

4.4	Численные эксперименты по сравнению двух подходов к построению неотрицательных разложения Таккера и и ТТ . . .	93
4.4.1	Синтетические данные . . . . .	94
4.4.2	Гиперспектральное изображение . . . . .	98
4.5	Вычисление меры влиятельности узлов с помощью NTT, искусственные данные . . . . .	101
4.6	Вычисление меры влиятельности узлов с помощью NTT на примере транспортной модели . . . . .	104
	<b>Заключение</b>	<b>106</b>
	<b>Список литературы</b>	<b>109</b>
	<b>Публикации автора по теме диссертации</b>	<b>116</b>

# Введение

Данная работа посвящена разработке и программной реализации эффективных алгоритмов для неотрицательной факторизации матриц и тензоров. В литературе первые статьи по неотрицательной матричной факторизации появились в 1994 году [1], однако популярность эта проблема приобрела после публикации работы [2] в 1999, в которой был предложен алгоритм, получивший название “мультипликативные обновления”. Задача неотрицательной матричной факторизации (Nonnegative Matrix Factorization – NMF) формулируется следующим образом: имея  $m \times n$  матрицу  $V$  с  $V_{ij} \geq 0, i = \overline{1, m}, j = \overline{1, n}$  (в дальнейшем неотрицательность элементов матрицы будет обозначаться как  $V \geq 0$ ) и натуральное число  $r < \min(m, n)$ , требуется найти такие матрицы  $W \in \mathbb{R}^{m \times r}$  и  $H \in \mathbb{R}^{r \times n}$  с неотрицательными элементами, что  $V \approx WH$ . Минимальное  $r$  такое, что  $V = WH$ , называется неотрицательным рангом  $V$ .

Задача построения разложения матрицы является хорошо изученной, однако добавление требования неотрицательности кардинально меняет ситуацию. В общем случае неотрицательная матричная факторизация считается NP-трудной проблемой даже при условии, что неотрицательный ранг матрицы известен [3]. То же самое можно сказать о сложности поиска неотрицательного ранга матрицы. Если  $W, H$  - решение задачи, то пара  $W_1 = WS, H_1 = S^{-1}H$  тоже является решением при условии, что матрица перехода  $S$  невырожденная и сохраняет неотрицательность компонент разложения.

Обычно для поиска решения проблему NMF формулируют как задачу оптимизации:

$$(W^*, H^*) = \arg \min_{W \geq 0, H \geq 0} D(V || WH),$$

где  $D(P || Q) = \sum_{i=1}^m \sum_{j=1}^n d(p_{ij}, q_{ij}) : d(p, q) \geq 0, d(p, q) = 0 \Leftrightarrow p = q$ .

В качестве функции цены  $D$  чаще всего выбирают норму Фробениуса или обобщенную дивергенцию Кульбака-Лейблера. Сформулированная задача оптимизации не является выпуклой и может иметь несколько локальных минимумов.

Матрицы – один из простых способов представления информации, однако во многих прикладных задачах чаще приходится работать с многомерными массивами, или тензорами. И здесь мы сталкиваемся с другой проблемой: совершать численные операции с ними при большом числе измерений представляется невозможным из-за "проклятия размерности". Имеется в виду, что память, требуемая для хранения элементов, и число действий, необходимых для выполнения базовых манипуляций с тензорами, растет экспоненциально с увеличением ранга  $d$ . Таким образом, возникает потребность в неких малопараметрических представлениях, одним из которых стал тензорный поезд -  $TT$ -разложение [4].

Однако часто существует ограничение на неотрицательность элементов тензоров, и это условие должно выполняться и для их приближений. Данное требование возникает в задачах машинного обучения и обработки изображений или при проведении вычислений для физических моделей и позволяет дать интерпретацию полученным результатам. Отсюда появляется необходимость в развитии такой технологии, как неотрицательная тензорная факторизация - NTF (Nonnegative Tensor Factorisation).

Ранее в литературе уже были рассмотрены алгоритмы NTF для канонического разложения и разложения Таккера. Но к моменту начала работы автора над диссертацией почти не было методов для построения неотрицательной аппроксимации исходного тензора тензорным поездом, когда вагоны заполнены только неотрицательными числами. К 2019 г. единственный алгоритм, который удалось найти, был описан в [5]. Метод называется NTT-HALS (Nonnegative Tensor Train – Hierarchical Alternating Least Squares), в нем изначально фиксируются желаемые ранги разложения, далее с помощью метода ALS (Alternating Least Squares) инициализируется тензорный поезд, а затем в итерационном процессе проводится оптимизация приближения. Последовательное построение улучшенной аппроксимации строится по алгоритму HALS. Для NTT-HALS в данный момент не известны никакие оценки точности результата, более того, сложность алгоритма делает его не подходящим для работы с большими данными.

**Основной целью** данной работы является построение эффективных методов для неотрицательной факторизации матриц и тензоров на основе

малоранговых аппроксимаций и быстрых алгоритмов линейной алгебры. Также целью является теоретическое исследование предложенных методов, их реализация в виде программ и рассмотрение приложений. В качестве приложений были рассмотрены гиперспектральные изображения, тензор ERP (event related potential – электрофизиологический отклик на стимул), ядра уравнения Смолуховского и ранжирование вершин многомерного графа.

**Научная новизна.** В работе предложены новые методы для неотрицательной факторизации матриц и тензоров, указаны оценки алгоритмической сложности, приведены теоретические результаты, доказывающие эффективность алгоритмов для ряда задач. Разработанные методы реализованы программно и протестированы на ряде примеров в сравнении с другими алгоритмами.

#### **Теоретическая ценность.**

В данной работе предлагается использовать для неотрицательной факторизации матриц и тензоров двухэтапный подход, где на первом шаге строится малоранговая аппроксимация, далее алгоритм неотрицательного разложения работает с вычисленной аппроксимацией вместо исходного объекта, что при работе с большими данными позволяет получить ускорение в тысячи раз.

Для определенных классов матриц, а именно неотрицательных сепарабельных матриц и неотрицательных матриц ранга 2, предлагаются методы решения задачи неотрицательной факторизации за время, линейно зависящее от размеров матриц. Исследуется случай возмущенных сепарабельных матриц и доказывается теорема о том, как зависит точность аппроксимации от величины возмущений.

Предлагаются алгоритмы для построения неотрицательного тензорного произведения, канонического полиадического разложения и разложения Таккера на основе двухэтапного подхода, позволяющие снизить сложность итерации с  $O(n^d)$  до линейной зависимости от  $n$ . Дополнительно рассматривается возможность комбинации алгоритма факторизации в неотрицательный тензорный продукт с методом DMRG, чтобы обеспечить возможность выбора рангов динамически.

Предлагается подход точечной коррекции элементов разложения неотрицательного тензора в формате тензорного произведения с целью получения неотрицательной факторизации.

Рассмотрено применение модели неотрицательного тензорного произведения при вычислении собственных векторов Перрона, которые используются, в частности, для моделирования степени важности узлов многомерной сети.

**Практическая ценность** заключается в программной реализации предложенных алгоритмов, позволяющих ускорить расчеты за счет использования малоранговых приближений и быстрых методов линейной алгебры, что подтверждается результатами численных экспериментов для ряда задач.

Полученные теоретические результаты и разработанный комплекс программ позволяют качественно расширить круг задач, доступных для детального изучения методами математического моделирования.

На данный момент алгоритмы, полученные в диссертации, уже успешно применяются для вычисления неотрицательного решения уравнения Смолуховского, сжатия видео и изображений с запуском на нескольких процессорах.

**На защиту выносятся следующие результаты и положения.** Основные результаты:

- Обоснован эффективный редуцированный двухэтапный алгоритм неотрицательной факторизации сепарабельных матриц и матриц ранга 2, исследована применимость метода к возмущенным сепарабельным матрицам, выведена оценка зависимости точности приближения от величины возмущения;
- Разработаны методы построения неотрицательного тензорного произведения для аппроксимации исходного тензора с неотрицательными элементами с использованием двухэтапного подхода. Рассмотрена комбинация одного из алгоритмов с техникой DMRG, что позволяет подбирать также ранги аппроксимации, которые в большинстве алгоритмов считаются входными данными;
- Предложен подход к построению неотрицательного ТТ-приближения

(без требования неотрицательности элементов ядер), основанный на точечной коррекции элементов, данный метод успешно применяется для решения мультикомпонентного уравнения Смолуховского (математической модели процессов коагуляции при неупругих соударениях огромного числа частиц) с сохранением неотрицательности;

- Разработанные методы реализованы в виде комплекса программ, проведен ряд численных экспериментов, иллюстрирующих эффективность и точность предложенных методов, описано применение модели неотрицательного ТТ при ранжировании узлов многомерного графа.

**Апробация работы.** Основные результаты работы докладывались автором и обсуждались на следующих конференциях:

- “The Sixth China-Russia Conference on Numerical Algebra with Applications” (Москва, 2017)
- “Ломоносов” (МГУ имени М. В. Ломоносова, Москва, 2018)
- “Тихоновские чтения” (МГУ имени М. В. Ломоносова, Москва, 2018)
- “Large-Scale Scientific Computations” (Созополь, Болгария, 2019)
- “SIAM Conference on Applied Algebraic Geometry” (Берн, Швейцария, 2019)
- “Методы вычислений и математическая физика” (Сочи, НТУ Сириус, 2020)
- “The 6th international conference on matrix methods and machine learning in mathematics and applications” (Москва, научный парк МГУ Lomonosov Hall и ИВМ РАН, 2023)
- “Матричные методы и интегральные уравнения” (НТУ Сириус, 2023)

### **Публикации**

Основные результаты по теме диссертации изложены в 5 печатных изданиях, изданных в журналах Scopus, WoS, RSCI, а также в изданиях,

рекомендованных для защиты в диссертационном совете МГУ по специальности.

Работа А.1 полностью выполнена автором. В статье А.2 идея редуцированного алгоритма для сепарабельных матриц была предложена автором, автор участвовала в доказательстве лемм и теорем, программная реализация и проведение численных экспериментов были выполнены автором полностью самостоятельно. В работе А.3 автор предложила алгоритм, дала оценку точности аппроксимации, выполнила программную реализацию и провела численные эксперименты. В статье А.4 автор реализовала и развила идею точечной корректировки элементов в аппроксимации тензорным поездом с целью получения неотрицательного приближения, предложила двухэтапный подход к построению неотрицательного канонического разложения и разложения Таккера, снизив сложность одной итерации с  $O(n^d)$  до  $O(dnr^3)$ , реализовала все методы программно и провела численные эксперименты. В работе А.5 автор провела численное исследование двух подходов малоранговой неотрицательной тензорной факторизации, которое на практике демонстрирует превосходство предложенных ей ранее алгоритмов.

Диссертационная работа является самостоятельным и законченным трудом автора.

### **Структура работы.**

Диссертационная работа состоит из введения, четырёх глав, заключения, списка литературы и списка публикаций автора. Общий объем диссертации 117 страниц, включая 13 рисунков, 17 таблиц и список литературы из 73 наименований.

### **Содержание работы.**

В первой главе исследуется возможность построения неотрицательной факторизации матрицы размера  $m \times n$ , зная ее неотрицательный ранг, используя лишь несколько ее строк и столбцов. Предлагаются методы решения этой задачи для определенных классов матриц: неотрицательных сепарабельных матриц – тех, для которых существует конус, натянутый на несколько столбцов исходной матрицы и содержащий все ее столбцы; неотрицательных сепарабельных матриц с возмущениями; неотрицатель-

ных матриц ранга 2. На практике предложенные алгоритмы используют число операций и объем памяти, линейно зависящие от  $m + n$ . Исследован случай возмущенных сепарабельных матриц. В частности, выведена теоретическая оценка точности работы алгоритма в зависимости от возмущения.

Во второй главе вводится модель неотрицательного тензорного поезда. Предлагаются методы построения неотрицательного тензорного поезда для аппроксимации исходного тензора с неотрицательными элементами с использованием двухэтапного подхода. Алгоритм NTTF (Nonnegative Tensor Train Factorization) основан на последовательном применении двухэтапной неотрицательной факторизации матриц к матрицам-разверткам. Исследован вопрос точности разложения. В методе NTT-MU (Nonnegative Tensor Train - Multiplicative Updates) на первом шаге исходный тензор факторизуется в тензорный поезд, все последующие операции проводятся с этим разложением, за счет чего снижается общая сложность алгоритма. Далее формируется начальное неотрицательное приближение в формате тензорного поезда, затем на каждой итерации все вагоны, кроме одного, фиксируются, значения оставшегося ядра заменяются по правилу мультипликативных обновлений, чтобы улучшить аппроксимацию. Рассмотрена комбинация NTT-MU с техникой DMRG, что позволяет подбирать также ранги аппроксимации, которые в большинстве алгоритмов считаются входными данными. Предлагается использование двухэтапного подхода для построения других разложений, в частности неотрицательного канонического разложения и разложения Таккера, применение его позволяет снизить сложность одной итерации известных алгоритмов с  $O(n^d)$  до  $O(dnr^3)$ . В главе также прорабатывается идея точечной корректировки элементов в аппроксимации тензорным поездом с целью получения неотрицательного приближения.

Одной из главных задач специалистов по обработке данных является выявление релевантных компонентов в сети или графе. В третьей главе рассматривается применение неотрицательной тензорной факторизации для ускорения вычислений меры влияния, или центральности, узлов многомерной сети. Мера центральности - это вещественнозначная функция набора узлов, которая является инвариантной при перемаркировке узлов

и, таким образом, может быть использована для их ранжирования в соответствии с их важностью. В [6] авторы используют собственный вектор Перрона простого мультиоднородного отображения для ранжирования узлов важности многоуровневой сети, однако алгоритм вычисления векторов включает операцию умножения исходного тензора, описывающего многоуровневую сеть, на вектор. В главе предлагается использовать неотрицательную аппроксимацию в формате тензорного произведения с целью ускорения вычислений, выводится оценка приближенного решения.

Четвёртая глава посвящена описанию численных экспериментов и их результатов. На практике подтверждаются теоретические оценки сложности и точности предложенных алгоритмов, проводится сравнение с существующими методами, демонстрирующее ускорение вычислений при сравнимой точности решения. Объекты исследования - синтетические и реальные данные, такие как изображения, данные отклика мозга в ответ на сенсорное, когнитивное либо двигательное явление, ядра уравнения Смолуховского, транспортная модель.

**Благодарности.** Автор выражает благодарность академику РАН Тыртышникову Евгению Евгеньевичу за научное руководство и постоянную поддержку в исследованиях. Отдельно автор хотела бы поблагодарить родителей, Щербакову Татьяну Геннадьевну и Щербакова Михаила Юрьевича, без чьих заботы и внимания проведение данного исследования было бы невозможным.

# 1. Неотрицательная факторизация матриц

## 1.1. Сепарабельные матрицы

**Определение 1.** Матрица  $M \in R^{m \times n}$  называется  $r$ -сепарабельной, если все ее столбцы принадлежат конусу, натянутому на некоторые  $r$  столбцов той же матрицы. Соответствующая система столбцов называется определяющей.

Очевидно, что первые  $r$  столбцов могут быть сделаны определяющими с помощью перестановок. Тогда матрица  $M$  может быть представлена в виде:

$$M = WH = W[I_r \ H'],$$

где  $W \in R^{m \times r}$ ,  $H \in R_+^{r \times n}$ ,  $I_r$  - единичная матрица порядка  $r$ . В общем случае ранг  $r$ -сепарабельной матрицы необязательно равен  $r$ .

На первый взгляд условие сепарабельности может показаться несколько искусственным, однако существуют приложения, где оно является естественным и разумным. Например, данное условие широко распространено при работе с гиперспектральными изображениями. В данном случае каждый столбец  $M$  интерпретируется как “спектральная подпись” смеси материалов для одного элемента изображения, и условие сепарабельности означает, что для каждого материала существует элемент изображения, содержащий исключительно его. При работе с текстами столбцы  $M$  могут описывать слова. Тогда условие сепарабельности будет означать, что для каждой темы существует слово, относящееся только к ней.

В дальнейшем нам нужны некоторые дополнительные предположения о матрицах  $W$  и  $H$ .

Во-первых, будем считать, что столбцы матрицы  $W \in R^{m \times r}$  линейно независимы. Тогда  $W$  и  $H$  определены единственным образом с точностью до перестановки и масштабирования столбцов. Другими словами, пусть матрицы  $W$  и  $\widetilde{W}$  имеют по  $r$  линейно независимых столбцов и каждая из матриц  $H$  и  $\widetilde{H}$  содержит единичную подматрицу порядка  $r$ . Тогда, если

невырожденная матрица  $Q$  обеспечивает равенства

$$\widetilde{W} = WQ \geq 0, \quad \widetilde{H} = Q^{-1}H \geq 0,$$

то она обязана быть *мономиальной* – так называются квадратные матрицы, которые в каждой строке и каждом столбце имеют ровно один ненулевой элемент. При сделанных предположениях матрицы  $Q$  и  $Q^{-1}$  обе должны быть неотрицательными и очевидным образом доказывается, что такие матрицы являются мономиальными. Более тонкие вопросы единственности неотрицательных разложений обсуждаются в [15], [22].

Во-вторых, предположим, что сумма элементов в каждом столбце матрицы  $H \in R_+^{r \times n}$  не превышает 1. Такую матрицу будем называть *нормированной  $r$ -сепарабельной*. В случае неотрицательных сепарабельных матриц этого можно легко добиться с помощью нормировки, делающей сумму элементов каждого ненулевого столбца равной 1. В самом деле,

$$M = WH \Leftrightarrow MD_M^{-1} = WD_W^{-1}(D_WHD_M^{-1}),$$

где матрица  $D_X$  определяется как диагональная матрица, в которой  $i$ -й элемент диагонали равен сумме элементов  $i$ -го столбца матрицы  $X$ , если он ненулевой, а в противном случае этот элемент равен 1. Суммы элементов в ненулевых столбцах матриц  $MD_M^{-1}$  и  $WD_W^{-1}$  равны 1, а позиции нулевых столбцов соответствуют нулевым столбцам матрицы  $M$ . То же верно для матрицы  $D_WHD_M^{-1}$ , при этом все ее элементы будут неотрицательными. Заметим однако, что нормировка требует просмотра всех  $mn$  элементов матрицы  $M$ .

Предлагаемый алгоритм представляет собой модификацию алгоритма Гиллиса–Вавасиса (Gillis–Vavasis), которая работает с некоторой факторизацией исходной матрицы без предположения о неотрицательности факторов. Таким образом, полный массив элементов исходной матрицы не требуется и за счет этого новый алгоритм обладает существенно меньшей вычислительной сложностью. Прежде чем описывать новый алгоритм, поясним идеи, лежащие в основе метода Гиллиса–Вавасиса.

Задача алгоритма – найти определяющую систему столбцов неотри-

цательной  $r$ -сепарабельной матрицы  $M$ . Предполагается, что матрица  $M$  нормирована так, как было указано выше. В этом случае столбец максимальной длины заведомо входит в определяющую систему столбцов.

Последнее вытекает из следующего общего наблюдения. Пусть имеются векторы  $w_1, \dots, w_r$ , среди них есть хотя бы один ненулевой вектор и все ненулевые векторы попарно различны. Пусть числа  $\alpha_1, \dots, \alpha_r$  неотрицательны, их сумма не больше 1 и ни одно из них не равно 1. Тогда

$$\left\| \sum_{1 \leq i \leq r} \alpha_i w_i \right\| < \max_{1 \leq i \leq r} \|w_i\|.$$

В самом деле, не ограничивая общности, можно считать, что все векторы ненулевые. Если  $r = 2$ , то неравенство имеет ясный геометрический смысл: в треугольнике  $ABC$  для любой точки на отрезке  $BC$ , отличной от вершин, длина отрезка  $AP$  строго меньше длины наибольшей из сторон  $AB$  и  $AC$ . В вырожденном случае, когда  $AB$  и  $AC$  коллинеарны, точка  $P$  будет лежать между точками  $B$  и  $C$  на отрезке  $BC$ , значит, длина  $AP$  строго меньше длины наибольшего из векторов  $AB$  и  $AC$ . При  $r \geq 3$  очевидным образом применяется индукция.

Итак, столбец максимальной длины матрицы  $M$  входит в искомую определяющую систему. Чтобы получить остальные столбцы, вычтем из каждого столбца его ортогональную проекцию на линейную оболочку найденного вектора. Преобразованная таким образом матрица, которую обозначим через  $M'$ , в общем случае уже не будет неотрицательной. Но она останется  $r$ -сепарабельной матрицей с разложением вида  $M' = W'H$ , в котором один из столбцов матрицы  $W'$  нулевой, а остальные линейно независимы. Действительно, пусть  $j$ -й столбец матрицы  $M$  равен

$$m_j = \sum_{i=1}^r h_{ij} w_i$$

и для каждого столбца  $w_i$  записано разложение  $w_i = u_i + v_i$  по одной и той же паре ортогональных подпространств. Тогда в аналогичном разложении

вектора  $m_j = z_j + p_j$  получаем

$$z_j = \sum_{i=1}^r h_{ij} u_i.$$

Заметим, что номера определяющих столбцов матрицы  $M'$  те же, что и для матрицы  $M$ . При этом уже найденный нами определяющий столбец матрицы  $M$  соответствует нулевому столбцу матрицы  $M'$ . Если  $r \geq 2$ , то столбец максимальной длины в матрице  $M'$  не может быть нулевым. Но известно, что он входит в определяющую систему для  $M'$ , а значит и в определяющую систему для  $M$ . Таким образом, имеется уже два определяющих столбца матрицы  $M$ . Продолжая в том же духе, будут найдены все столбцы искомой определяющей системы.

Краткое формальное описание изложенных преобразований приводится ниже в виде псевдокода. Это и есть алгоритм Гиллиса–Вавасиса [15]. На его вход подается произвольная нормированная  $r$ -сепарабельная матрица ранга  $r$ , ее неотрицательность не требуется. Выше было продемонстрировано, что в случае неотрицательной матрицы необходимую нормировку можно выполнить достаточно просто. Это должно быть сделано перед применением алгоритма.

**Дано:**  $M$  – нормированная  $r$ -сепарабельная  $m \times n$ -матрица ранга  $r$ .

**Найти:** Множество  $J$ , содержащее номера столбцов определяющей системы матрицы  $M$ .

```

1  $R = M, J = \{\}, j = 1;$ 
2 while  $R \neq 0$  &  $j \leq r$  do
3    $j^* = \arg \max_j \|R_{:j}\|_2^2;$ 
4    $u_j = R_{:j^*};$ 
5    $R \leftarrow (I - \frac{u_j u_j^T}{\|u_j\|_2^2}) R;$ 
6    $J = J \cup \{j^*\};$ 
7    $j = j + 1;$ 
8 end
```

**Алгоритм 1.** Алгоритм Гиллиса–Вавасиса [15].

В алгоритме Гиллиса–Вавасиса выполняется  $O(mnr)$  арифметических

операций. В работах [7]-[10] было показано, что для матриц ранга  $r$  скелетные разложения могут быть построены с использованием специально выбранных  $r$  столбцов и  $r$  строк заданной матрицы. Соответствующие методы обычно называются *крестовыми*. В них для получения разложений достаточно памяти для хранения порядка  $(m+n)r$  чисел, и в случае  $m = n$  сложность зависит от  $n$  линейно, а не квадратично. Более точно вычислительные затраты составляют порядка  $(m+n)r^2$  операций.

Будем считать, что у нас уже есть разложение вида  $M = UV$ , где матрицы  $U$  и  $V$  имеют  $r$  столбцов и  $r$  строк соответственно, но в этих матрицах допускается наличие отрицательных элементов. Получив  $U$  и  $V$  и далее используя только их, предлагается *редуцированный алгоритм* вычисления неотрицательной факторизации матрицы  $M$  существенно меньшей сложности, чем алгоритм Гиллиса–Вавасиса.

Пусть известно, что произведение  $M = UV$  является неотрицательной  $r$ -сепарабельной матрицей ранга  $r$ . Тогда вектор-строка

$$d = (eU)V, \quad e = [1, \dots, 1],$$

содержит  $n$  элементов и может быть найдена с затратами  $2(m+n)r$  арифметических операций (сначала вычисляем строку  $eU$ , а затем умножаем результат на матрицу  $V$ ). Очевидно, что  $d_j$  есть сумма элементов  $j$ -го столбца матрицы  $M$ . Составим из них диагональную матрицу порядка  $n$  и каждый нулевой элемент главной диагонали, если таковые есть, заменим на 1. Полученную диагональную матрицу обозначим через  $D$ . Очевидно, матрица  $\tilde{M} = MD^{-1}$  будет нормированной  $r$ -сепарабельной.

**Лемма 1.** *Если  $M$  является неотрицательной  $r$ -сепарабельной матрицей ранга  $r$ , то  $\tilde{V} = VD^{-1}$  также будет нормированной  $r$ -сепарабельной матрицей ранга  $r$ .*

**Доказательство.** Для матрицы  $\tilde{M} = MD^{-1}$  имеет место разложение  $WH$ , где матрица  $W$  имеет  $r$  линейно независимых столбцов, а матрица  $H$  содержит единичную подматрицу порядка  $r$ , все ее элементы неотрицательны и сумма элементов в каждом столбце не больше 1. Из равенства  $U\tilde{V} = WH$  получаем  $\tilde{V} = \tilde{W}H$ , где  $\tilde{W} = (U^T U)^{-1} U^T W$ .

**Лемма 2.** *Определяющие столбцы матрицы  $\tilde{V}$  соответствуют определяющим столбцам матрицы  $M$ .*

**Доказательство.** Достаточно принять во внимание, что нормировка никак не влияет на порядок столбцов, поэтому у  $M$  и  $\tilde{M} = MD^{-1}$  совпадают множества индексов определяющих столбцов. Остается заметить, что из равенств  $\tilde{M} = U\tilde{V}$  и  $\tilde{V} = \tilde{W}H$  сразу же следует, что  $\tilde{M} = (U\tilde{W})H$ .

Объединяя приведенные рассуждения, можно сформулировать следующую теорему.

**Теорема 1.** *Пусть матрица  $M$  является неотрицательной сепарабельной матрицей ранга  $r$ . Если известно некоторое разложение  $M = UV$ ,  $U \in R^{m \times r}$ ,  $V \in R^{n \times r}$ , то число действий для построения неотрицательной матричной факторизации будет линейно зависеть от размеров матрицы.*

**Доказательство.** Из лемм 1 и 2 следует, что матрица  $\tilde{V} = VD^{-1}$  является нормированной  $r$ -сепарабельной матрицей ранга  $r$ , причем ее определяющие столбцы соответствуют определяющим столбцам матрицы  $M$ . Значит, чтобы найти определяющие столбцы матрицы  $M$ , достаточно применить алгоритм 1 не к исходной матрице размера  $m \times n$ , а к  $\tilde{V} = VD^{-1}$ , у которой  $r \times n$  элементов. Число требуемых арифметических операций оценивается как  $O(r^2n)$ .

Ранее обсуждалось, как найти матрицу  $D$ , а точнее вектор-строку  $d$  на ее диагонали, за  $2(m+n)r$  операций. Чтобы вычислить  $\tilde{V} = VD^{-1}$ , нужно поделить каждый столбец  $V$  на соответствующий элемент  $d$ , что потребует еще  $rn$  арифметических операций. В итоге, просуммировав число действий на поиск определяющих столбцов  $M$ , получена линейная зависимость от суммы размеров матрицы.

Ниже представлен предлагаемый редуцированный алгоритм.

**Дано:** вещественные матрицы  $U$  и  $V$  размеров  $m \times r$  и  $n \times r$  соответственно. Предполагается известным, что произведение  $M = UV$  является неотрицательной  $r$ -сепарабельной матрицей ранга  $r$ .

**Найти:** Множество индексов  $J$ , соответствующих определяющей системе  $M$ .

```

1  $d = eU, \quad e = [1, \dots, 1]$ 
2  $d = dV$ 
3 for (  $j = 1; j \leq n; i = i + 1$  ) {
4   | for (  $i = 1; i \leq r; i = i + 1$  ) {
5   |   |  $\tilde{v}_{ij} = v_{ij}/d_j$ 
6   |   }
7   }
8 Алгоритм 1 применяется к матрице  $\tilde{V}$ .
```

**Алгоритм 2.** Редуцированный алгоритм.

При работе с большими объемами данных имеет смысл использовать описанный метод в целях ускорения вычислений, его преимущество по сравнению с алгоритмом 1 наглядно продемонстрировано в главе, посвященной численным экспериментам.

## 1.2. Возмущенные сепарабельные матрицы

Пусть  $A$  – нормированная сепарабельная матрица ранга  $r$ , а предложенный нами редуцированный алгоритм применяется к факторизации возмущенной матрицы  $\tilde{A} = A + E$ , где евклидова длина каждого столбца матрицы  $E$  не выше  $\varepsilon > 0$ . Очевидно, что при достаточно малых возмущениях алгоритм Гиллиса–Вавасиса найдет возмущенные векторы определяющей системы исходной матрицы  $M$ . Существенно более трудный вопрос – выяснить, насколько малыми должны быть возмущения и как именно зависит от них точность приближений. В [15] получены некоторые конкретные, хотя и завышенные оценки в ситуации, когда максимизируется не длина, а произвольно взятая сильно выпуклая функция вектора. Практического

смысла в таком обобщении, как признают сами авторы, им выявить не удалось. В этом разделе приводится более простой вывод оценок того же типа в случае максимизации именно длины и применим их для анализа нашего редуцированного алгоритма в условиях возмущений.

Прежде всего нам нужно исследовать ситуацию, когда при поиске вектора наибольшей длины в возмущенной матрице будет выбран столбец, соответствующий нетривиальной выпуклой комбинации определяющих столбцов исходной (невозмущенной) сепарабельной матрицы. При некотором ограничении на возмущение мы покажем, что найденный столбец будет, тем не менее, близок к какому-то из столбцов определяющей системы исходной матрицы. Чтобы это сделать, предлагается использовать следующую лемму.

**Лемма 3.** *Для произвольных векторов  $a_1, \dots, a_k$  и их выпуклой комбинации с коэффициентами  $\alpha_1, \dots, \alpha_k$  справедливо тождество*

$$\left\| \sum_{i=1}^k \alpha_i a_i \right\|^2 = \sum_{i=1}^k \alpha_i \|a_i\|^2 - \sum_{1 \leq i < j \leq k} \alpha_i \alpha_j \|a_i - a_j\|^2.$$

**Доказательство.** При  $k = 2$  получаем легко проверяемое равенство

$$\|\alpha_1 a_1 + \alpha_2 a_2\|^2 = \alpha_1 \|a_1\|^2 + \alpha_2 \|a_2\|^2 - \alpha_1 \alpha_2 \|a_1 - a_2\|^2.$$

При  $k > 2$  применяем индукцию

$$\begin{aligned} \left\| \sum_{i=1}^k \alpha_i a_i \right\|^2 &= \left\| \alpha_1 a_1 + (1 - \alpha_1) \sum_{i=2}^k \frac{\alpha_i}{1 - \alpha_1} a_i \right\|^2 = \\ &= \alpha_1 \|a_1\|^2 + (1 - \alpha_1) \left\| \sum_{i=2}^k \frac{\alpha_i}{1 - \alpha_1} a_i \right\|^2 - \alpha_1 (1 - \alpha_1) \left\| \sum_{i=2}^k \frac{\alpha_i}{1 - \alpha_1} (a_i - a_1) \right\|^2 = \\ &= \alpha_1 \|a_1\|^2 + (1 - \alpha_1) \left( \sum_{i=2}^k \frac{\alpha_i}{1 - \alpha_1} \|a_i\|^2 - \sum_{2 \leq i < j \leq k} \frac{\alpha_i \alpha_j}{(1 - \alpha_1)^2} \|a_i - a_j\|^2 \right) - \end{aligned}$$

$$\begin{aligned}
& -\alpha_1(1 - \alpha_1) \left( \sum_{i=2}^k \frac{\alpha_i}{1 - \alpha_1} \|a_i - a_1\|^2 - \sum_{2 \leq i < j \leq k} \frac{\alpha_i \alpha_j}{(1 - \alpha_1)^2} \|a_i - a_j\|^2 \right) = \\
& = \sum_{i=1}^k \alpha_i \|a_i\|^2 - \sum_{j=2}^k \alpha_1 \alpha_j \|a_1 - a_j\|^2 - \sum_{2 \leq i < j \leq k} \alpha_i \alpha_j \|a_i - a_j\|^2 = \\
& = \sum_{i=1}^k \alpha_i \|a_i\|^2 - \sum_{1 \leq i < j \leq k} \alpha_i \alpha_j \|a_i - a_j\|^2.
\end{aligned}$$

**Следствие 1.** Для произвольной выпуклой комбинации заданных векторов имеет место неравенство

$$\left\| \sum_{i=1}^k \alpha_i a_i \right\|^2 \leq \max_{1 \leq i \leq k} \|a_i\|^2 - \left( \sum_{1 \leq i < j \leq k} \alpha_i \alpha_j \right) \min_{1 \leq i < j \leq k} \|a_i - a_j\|^2.$$

Отметим еще одно интересное следствие, которое в данной работе не используется, но может быть полезно при работе с сильно выпуклыми функциями. Так называются выпуклые функции  $f$ , для которых выполняется неравенство

$$f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y) - \gamma t(1 - t)\|x - y\|^2, \quad 0 \leq t \leq 1,$$

где  $\gamma$  – положительное число, называемое параметром сильной выпуклости, а  $x$  и  $y$  – произвольные векторы, принадлежащие выпуклой области определения функции  $f$ .

**Следствие 2.** Для произвольной сильно выпуклой функции  $f$  с параметром сильной выпуклости  $\gamma$  и произвольной выпуклой комбинации векторов  $a_1, \dots, a_k$  имеет место неравенство

$$f\left(\sum_{i=1}^k \alpha_i a_i\right) \leq \sum_{i=1}^k \alpha_i f(a_i) - \gamma \sum_{1 \leq i < j \leq k} \alpha_i \alpha_j \|a_i - a_j\|^2.$$

Доказательство проводится по индукции в полной аналогии с доказательством леммы 3 и с использованием установленного в ней тождества.

В литературе автором было найдено эквивалентное данному неравенство только для случая, когда  $a_1, \dots, a_k$  – вещественные числа (см. [23]).

Для вывода оценок нам нужны некоторые величины, связанные с системой столбцов  $a_1, \dots, a_k$  или составленной из них матрицы  $A$ :

$$\mu = \mu(A) = \max_{1 \leq i \leq k} \|a_i\|, \quad \gamma = \gamma(A) = \min_{1 \leq i \leq k} \|a_i\|, \quad \omega = \omega(A) = \min_{1 \leq i < j \leq k} \|a_i - a_j\|.$$

Полученную выше оценку длины выпуклой комбинации вида

$$b = \sum_{i=1}^k \alpha_i a_i$$

можно записать в виде неравенства

$$\|b\|^2 \leq \mu^2 - \left( \sum_{1 \leq i < j \leq k} \alpha_i \alpha_j \right) \omega^2.$$

**Лемма 4.** *Предположим, что в  $\varepsilon$ -окрестности вектора  $b$  имеется вектор, длина которого не меньше  $\mu$ , и пусть  $a_j$  – вектор с наибольшим коэффициентом  $\alpha_j$ . Тогда для всех достаточно малых  $\varepsilon$  справедливы оценки*

$$1 - \alpha_j = O\left(\frac{\mu}{\omega^2}\varepsilon\right), \quad (1)$$

$$\|b - a_j\| = O\left(\frac{\mu^2}{\omega^2}\varepsilon\right). \quad (2)$$

**Доказательство.** Если  $\alpha_j = 1$ , то утверждение очевидно. Полагаем далее, что  $\alpha_j < 1$ . Пусть вектор  $e$  имеет длину  $\|e\| \leq \varepsilon \leq \mu$  и при этом  $\|b + e\| \geq \mu$ . Тогда, согласно следствию 1, имеем

$$\left( \sum_{1 \leq i < j \leq k} \alpha_i \alpha_j \right) \omega^2 \leq \mu - \|b\|^2 \leq \|b + e\|^2 - \|b\|^2 \leq (2\|b\| + \|e\|)\varepsilon \leq 3\mu\varepsilon.$$

Принимая во внимание равенство

$$\sum_{1 \leq i < j \leq k} \alpha_i \alpha_j = \frac{1}{2} \sum_{i=1}^k \alpha_i (1 - \alpha_i),$$

находим

$$\alpha_j (1 - \alpha_j) \leq c\varepsilon, \quad c = \frac{6\mu}{\omega^2}.$$

Будем считать, что  $4c\varepsilon \leq 1/k$ . Тогда заведомо  $\sqrt{1 - 4c\varepsilon} \geq 1 - 4c\varepsilon$  и, следовательно, имеются две возможности:

$$\alpha_j \geq \frac{1 + \sqrt{1 - 4c\varepsilon}}{2} \geq 1 - 2c\varepsilon \quad (3)$$

или

$$\alpha_j \leq \frac{1 - \sqrt{1 - 4c\varepsilon}}{2} \leq 2c\varepsilon. \quad (4)$$

Случай (4) приводит к противоречию между условием  $4c\varepsilon \leq 1/k$  и максимальнойностью коэффициента  $\alpha_j$ :

$$1 = \sum_{i=1}^k \alpha_i \leq k \cdot \max_{1 \leq i \leq k} \alpha_i \leq k \cdot 2c\varepsilon \quad \Rightarrow \quad 4c\varepsilon > 1/k.$$

Значит, справедливы неравенства  $\alpha_j \geq 1 - 2c\varepsilon$ ,  $1 - \alpha_j \leq 2c\varepsilon$  и, далее,

$$\|b - a_j\| = \left\| \sum_{i \neq j} \alpha_i (a_i - a_j) \right\| \leq 2(1 - \alpha_j)\mu \leq 4c\mu\varepsilon.$$

**Замечание.** Условие малости  $\varepsilon$  можно записать в виде  $\varepsilon \leq \min\{c_1 \frac{\sigma_r^2}{r\mu}, \mu\}$ , а его следствие – в виде неравенства  $\|b - a_j\| \leq c_2 \frac{\mu^2}{\omega^2} \varepsilon$ , если взять, например,  $c_1 = \frac{1}{12}$  и  $c_2 = 12$ .

Следующая теорема – это ключевой результат анализа возмущений. Его прототипом служит теорема из [15], полученная там для алгоритма, в котором на каждом шаге выбирается столбец, максимизирующий заданную сильно выпуклую функцию. В частности, такой функцией является квадрат длины вектора.

**Теорема 2.** Пусть  $A \in \mathbb{R}^{m \times n}$  – нормированная сепарабельная матрица ранга  $r$ , и предположим, что алгоритм Гиллеса–Вавасиса применяется к возмущенной матрице  $\tilde{A} = A + E$ , где длина каждого столбца матрицы  $E$  не превышает  $\varepsilon$ , и находит столбцы  $\tilde{a}_{i_1}, \dots, \tilde{a}_{i_r}$ . Тогда определяющие столбцы матрицы  $A$  можно занумеровать таким образом, что при всех достаточно малых  $\varepsilon$  имеют место неравенства

$$\|\tilde{a}_{i_k} - a_{j_k}\| \leq \left(1 + c \frac{\mu^2}{\sigma_r^2}\right) \varepsilon, \quad k = 1, \dots, r,$$

где  $\mu$  – наибольшая длина столбцов матрицы  $A$ ,  $\sigma_r$  – минимальное сингулярное число  $m \times r$ -матрицы, составленной из столбцов определяющей системы матрицы  $A$ ,  $c$  – некоторая положительная константа.

**Доказательство.** Пусть  $W = [w_1, \dots, w_r]$  – матрица, составленная из определяющих столбцов матрицы  $A$  и  $\sigma_r$  – ее минимальное сингулярное число. Тогда  $\|Wx\| \geq \sigma_r \|x\|$  для любого  $x \in \mathbb{R}^r$  и, следовательно,

$$\|w_i\| \geq \sigma_r, \quad \|w_i - w_j\| \geq \sqrt{2}\sigma_r.$$

Чтобы вывести первое неравенство, в качестве  $x$  надо взять  $i$ -й столбец единичной матрицы порядка  $r$ . Чтобы получить второе, в качестве  $x$  надо взять разность  $i$ -го и  $j$ -го столбцов единичной матрицы. Таким образом,

$$\mu = \mu(W) \geq \sigma_r, \quad \omega = \omega(W) \geq \sqrt{2}\sigma_r,$$

а оценку леммы 4 для первого шага алгоритма можно записать в виде

$$\|a_{i_1} - a_{j_1}\| = O\left(\frac{\mu^2}{\sigma_r^2}\varepsilon\right).$$

После выбора вектора  $\tilde{a}_{i_1}$  из каждого столбца матрицы  $\tilde{A}$  вычитается ортогональная проекция на выбранный столбец. В результате получается матрица

$$\tilde{A}_1 = Q_1 \tilde{A}, \quad Q_1 = I - q_1 q_1^T, \quad q_1 = \frac{\tilde{a}_{i_1}}{\|\tilde{a}_{i_1}\|},$$

близкая к нормированной сепарабельной матрице  $A_1 = Q_1 A$ . Пусть  $A =$

$WH$ . Тогда  $A_1 = W_1H$ , где  $W_1 = Q_1W$ , и, следовательно, коэффициенты выпуклых комбинаций в разложении столбцов матрицы  $A_1$  те же, что и для матрицы  $A$ . На втором шаге в матрице  $\tilde{A}_1$  будет выбран столбец с номером  $i_2 \neq i_1$ . Если столбец  $a_{i_2}$  входит в определяющую систему матрицы  $A$ , то  $j_2 = i_2$  и  $\|\tilde{a}_{i_2} - a_{i_2}\| \leq \varepsilon$ . В противном случае пусть номер  $j_2$  соответствует наибольшему элементу столбца  $i_2$  матрицы  $H$ . Обозначим этот элемент через  $\alpha_{j_2}$ . Тогда, согласно лемме 4, получим

$$1 - \alpha_{j_2} = O\left(\frac{\mu(W_1)}{\omega^2(W_1)}\varepsilon\right).$$

Очевидно, что  $\mu(W_1) \leq \mu = \mu(W)$ . Для определенности будем считать, что  $w_1 = a_{j_1}$ , и рассмотрим подматрицу  $\widehat{W} = [w_2, \dots, w_r]$  в  $W$  и соответствующую ей подматрицу  $\widehat{W}_1 = Q_1\widehat{W}$  в матрице  $W_1$ . Первый столбец матрицы  $W_1$  равен  $Q_1a_{j_1} \approx Q_1\tilde{a}_{i_1} = 0$  и, следовательно, его длина достаточно мала при малых  $\varepsilon$ . Значит, при достаточно малых  $\varepsilon$  находим

$$\omega(W_1) \geq \min\{\gamma(\widehat{W}_1) - \|Q_1w_1\|, \omega(\widehat{W}_1)\} \geq c_1\sigma_{r-1}(\widehat{W}_1),$$

где  $\sigma_{r-1}(\widehat{W}_1)$  – минимальное сингулярное число матрицы  $\widehat{W}_1$  и  $c_1$  – некоторая положительная константа. Пусть  $V_1$  получается из  $W_1$  заменой первого столбца на нулевой столбец. Тогда

$$\sigma_{r-1}(\widehat{W}_1) = \sigma_{r-1}(V_1) \geq \sigma_{r-1}(W_1) - \|Q_1w_1\|.$$

Теперь заметим, что  $\sigma_{r-1}(W_1) \geq \sigma_r(W)$ . Это следует из соотношений разделения для собственных значений эрмитовой матрицы  $W^TW$  и матрицы  $W_1^TW_1$ , которая получается из нее вычитанием эрмитовой положительно определенной матрицы ранга 1. Действительно,

$$W_1^TW_1 = W^T(I - q_1q_1^T)^2W = W^T(I - q_1q_1^T)W = W^TW - (W^Tq_1)(W^Tq_1)^T.$$

Нумеруя собственные значения в порядке невозрастания, получаем (см.,

например, теорему 5.9 из [24])

$$\lambda_{i-1}(W_1^T W_1) \geq \lambda_i(W^T W) \geq \lambda_i(W_1^T W_1), \quad 2 \leq i \leq r.$$

Итак, имеем

$$\sigma_{r-1}(\widehat{W}_1) \geq \sigma_r(W) - \|Q_1 w_1\|.$$

Следовательно, при малых  $\varepsilon$  получаем

$$1 - \alpha_{j_2} = O\left(\frac{\mu(W)}{\sigma_r^2(W)}\varepsilon\right) \Rightarrow \|\tilde{a}_{i_2} - a_{j_2}\| \leq \varepsilon + O\left(\frac{\mu^2(W)}{\sigma_r^2(W)}\varepsilon\right).$$

Таким образом, проведен анализ первого и второго шага алгоритма Гиллиса–Вавасиса. Те же построения дают оценки для последующих шагов.

В следующей теореме исследуется применение редуцированного алгоритма к факторизации, полученной каким-либо вариантом крестового метода [8]. Ниже используется норма Фробениуса  $\|E\|_F$ , определяемая как корень квадратный из суммы квадратов всех элементов матрицы.

**Теорема 3.** Пусть  $A \in \mathbb{R}^{m \times n}$  – нормированная сепарабельная матрица ранга  $r$  и рассматривается возмущенная матрица  $\tilde{A} = A + E$ , где  $\|E\|_F \leq \varepsilon$ . Тогда существуют такие  $r$  столбцов  $C$  и  $r$  строк  $R$  матрицы  $\tilde{A}$ , дающие в пересечении  $r \times r$ -матрицу  $G$ , что применение редуцированного алгоритма к факторизации  $CG^{-1}R$  находит столбцы  $\tilde{a}_{i_1}, \dots, \tilde{a}_{i_r}$ , которые при некоторой нумерации определяющих столбцов матрицы  $A$  при всех достаточно малых  $\varepsilon$  удовлетворяют неравенствам

$$\|\tilde{a}_{i_k} - a_{j_k}\| \leq O\left(r \frac{\mu^2}{\sigma_r^2} \varepsilon\right), \quad k = 1, \dots, r,$$

где  $\mu$  – наибольшая длина столбцов матрицы  $A$ ,  $\sigma_r$  – минимальное сингулярное число  $m \times r$ -матрицы, составленной из столбцов определяющей системы матрицы  $A$ .

**Доказательство.** В работе [25] доказано существование  $r$  столбцов и  $r$  строк, гарантирующих неравенство  $\|CG^{-1}R - \tilde{A}\|_F \leq (r + 1)\varepsilon$ . Остается принять во внимание оценки теоремы 2.

Заметим также, что аппроксимации ранга  $r$  могут строиться на основе большего, чем  $r$ , числа столбцов и строк (см. [26], [27]). Выбирая, например,  $2r$  столбцов и строк, можно гарантировать поэлементную оценку погрешности, в которой уже нет зависимости от  $r$ .

Интересным представляется вопрос, насколько точны полученные теоретические оценки и какова вероятность, что алгоритм выберет на некотором шаге вектор не из определяющей системы. Был проведен следующий эксперимент. Были сгенерированы матрицы  $W \in R^{10 \times 2}$  и  $H \in R^{2 \times 10000}$ , элементы которых равномерно распределенные на отрезке  $[0; 1]$  числа. Затем столбцы матрицы  $H$  были нормированы, чтобы сумма каждого была равна 1. Далее было подсчитано, как указано в замечании к 4, максимальное значение  $\varepsilon$ . После этого были рассмотрены 100 значений  $\tilde{\varepsilon}$  от 0 до  $\varepsilon$ , к каждому столбцу  $M = WH$  прибавлялся вектор  $\frac{\tilde{\varepsilon}}{\sqrt{n}}e$ , норма которого равна  $\tilde{\varepsilon}$ . Было подсчитано, сколько векторов в возмущенной матрице  $M = WH$  по норме превзошли  $\mu(W)$ . Результаты представлены на рис. 1. Даже при максимальном возмущении доля нетривиальных выпуклых комбинаций определяющих векторов, норма которых при возмущении больше  $\mu(W)$ , меньше 1.5%.

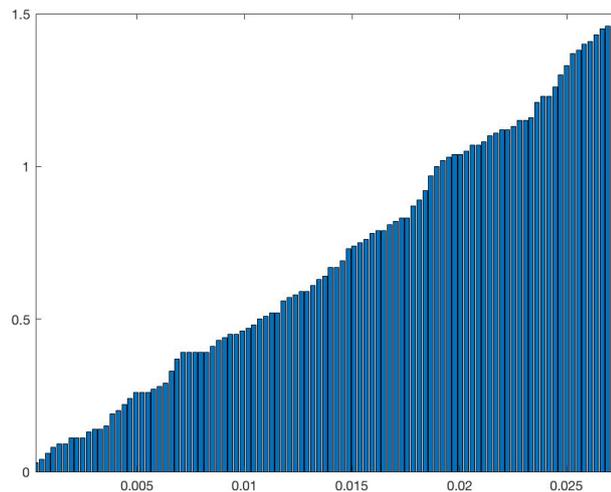


Рис. 1. Процент (из 10 тысяч) нетривиальных выпуклых комбинаций определяющих векторов, норма которых при возмущении больше  $\mu(W)$ , в зависимости от возмущения.

### 1.3. Редуцированный алгоритм для симметричной неотрицательной факторизации

В [20] предложен метод построения симметричной неотрицательной факторизации  $M = WW^T$  для симметричных неотрицательных матриц  $M \in \mathbb{R}^{m \times m}$  ранга  $r$ , содержащих диагональную главную подматрицу ранга  $r$ . Напомним, что главная подматрица получается вычеркиванием строк и столбцов с общими номерами. Матрица  $W$  определяется однозначно с точностью до перестановки столбцов (см. [20]). Время работы алгоритма, названного авторами IREVA (Identification and Rotation of Extreme Vectors Algorithm), оценивается ими как  $O(m^2r)$  для неразрезанных матриц.

Однако эту задачу можно решить гораздо быстрее. Для этого нужно заметить, что симметричные неотрицательные матрицы ранга  $r$  с диагональной главной подматрицей того же ранга принадлежат классу сепарабельных матриц. Если главную диагональную подматрицу обозначить  $D$ , а содержащие ее столбцы  $S$ , то справедливо разложение  $M = SD^{-1}S^T = S(D^{-1}S^T)$ . Очевидно, матрица  $D^{-1}S^T$  неотрицательна и содержит единичную подматрицу порядка  $r$ . Таким образом,  $M$  сепарабельна, а  $S$  состоит из ее определяющих столбцов. Чтобы их найти, можно применить алгоритм крестовой аппроксимации, а затем редуцированный алгоритм 2. Вместо  $O(m^2r)$  потребуется всего  $O(mr^2)$  операций.

### 1.4. Матрицы ранга 2

Очевидно, что для произвольной неотрицательной  $m \times n$ -матрицы имеют место неравенства  $\text{rank}(A) \leq \text{rank}_+(A) \leq \min(m, n)$ . Как показано в [21], если  $\text{rank}(A) \leq 2$ , то  $\text{rank}_+(A) = \text{rank}(A)$ .

**Теорема 4.** *Любая неотрицательная матрица ранга не выше 2 является сепарабельной матрицей того же ранга.*

**Доказательство.** Если  $\text{rank}(A) = 1$ , то  $A = uv^T = (\gamma|u|)(\gamma^{-1}|v|)^T$ , где  $\gamma = \max_{1 \leq j \leq n} |v_j|$ , а  $|u|$  и  $|v|$  обозначают векторы, координаты которых равны абсолютным величинам координат векторов  $u$  и  $v$ . Если  $\text{rank}(A) = 2$ , то,

разделив каждый столбец матрицы  $A$  на сумму его элементов, получим стохастическую матрицу с тем же неотрицательным рангом. Поэтому без ограничения общности можно считать  $A$  стохастической матрицей. Если  $a$  и  $b$  – какие-то линейно независимые столбцы, то  $j$ -й столбец записывается в виде линейной комбинации  $a_j = \alpha_j a + \beta_j b$ , где в силу стохастичности  $\alpha_j + \beta_j = 1$ . Пусть  $\beta_k$  и  $\beta_l$  – минимальное и максимальное среди чисел  $\beta_1, \dots, \beta_n$ . Тогда, поскольку  $\beta_k \leq \beta_j \leq \beta_l$ , можно найти числа  $\delta_j$  такие, что

$$\beta_j = (1 - \delta_j)\beta_k + \delta_j\beta_l, \quad \alpha_j = (1 - \delta_j)\alpha_k + \delta_j\alpha_l, \quad 0 \leq \delta_j \leq 1.$$

Отсюда

$$\begin{aligned} a_j &= \alpha_j a + \beta_j b = ((1 - \delta_j)\alpha_k + \delta_j\alpha_l)a + ((1 - \delta_j)\beta_k + \delta_j\beta_l)b = \\ &= (1 - \delta_j)(\alpha_k a + \beta_k b) + \delta_j(\alpha_l a + \beta_l b) = (1 - \delta_j)a_k + \delta_j a_l. \end{aligned}$$

Значит, любой столбец матрицы  $A$  является выпуклой комбинацией ее столбцов  $a_k$  и  $a_l$ .

**Следствие 3.** Если  $A \geq 0$  и  $\text{rank}_+(A) \leq 3$ , то  $\text{rank}_+(A) = \text{rank}(A)$ .

*Доказательство.* Пусть  $\text{rank}_+(A) = 3$ . Как установлено выше, если  $\text{rank}(A) \leq 2$ , то  $\text{rank}_+(A) = \text{rank}(A)$ . Следовательно,  $\text{rank}(A) = 3$ . Остальные случаи разбираются аналогично.

**Теорема 5.** Если для неотрицательной  $m \times n$ -матрицы  $A$  ранга 2 известно скелетное разложение  $A = UV$ , где  $U \in R^{m \times 2}$  и  $V \in R^{2 \times n}$ , то для получения неотрицательной факторизации ранга 2 достаточно выполнить число операций, линейно зависящее от суммы размеров матрицы  $A$ .

*Доказательство.* Линейного числа операций достаточно для того, чтобы получить разложение  $A = UV$ , в котором  $V$  содержит единичную матрицу порядка 2. Тогда  $U = [a, b]$  состоит из двух линейно независимых столбцов матрицы  $A$ . Матрицы  $A, U, V$  можно считать стохастическими (чтобы провести соответствующую нормировку, достаточно линейного

числа операций). Пусть  $\beta_1, \dots, \beta_n$  – элементы второй строки матрицы  $V$ . Согласно описанным выше построениям, найдем среди них минимальный  $\beta_{j^*}$  и максимальный  $\beta_{j^*}$ . Тогда пара столбцов матрицы  $A$  с номерами  $j^*$  и  $j^*$  составляют  $W$ , для определения  $H$  можно воспользоваться крестовым разложением, достаточно найти в  $W$  невырожденную подматрицу размера  $2 \times 2$ . Все эти действия требуют линейного числа операций.

Ниже приведен алгоритм для построения факторизации матриц ранга 2. Напомним, что  $D_X$  – диагональная матрица, в которой  $i$ -й элемент диагонали равен сумме элементов  $i$ -го столбца матрицы  $X$ , если он ненулевой, а в противном случае этот элемент равен 1.

**Дано:**  $A \in R_+^{m \times n}$ ,  $\text{rank}(A) = 2$ .

**Найти:** Матрицы  $W \in R_{\geq 0}^{m \times 2}$ ,  $H \in R_{\geq 0}^{2 \times n}$ :  $A = WH$ .

1 Строится крестовое разложение  $A$ :  $A = UV$ .

2  $\tilde{V} = D_U V$

3  $\tilde{V} = \tilde{V} D_{\tilde{V}}^{-1}$

4  $i_* = i : \tilde{v}_{2i} = \min_{1 \leq i \leq n} \tilde{v}_{2i}$

5  $i^* = i : \tilde{v}_{2i} = \max_{1 \leq i \leq n} \tilde{v}_{2i}$

6  $W = [a_{i_*} \ a_{i^*}]$

7 Найти в  $W$  строки  $p$  и  $s$ : подматрица  $\begin{bmatrix} w_{p1} & w_{p2} \\ w_{s1} & w_{s2} \end{bmatrix}$  невырожденная.

8  $H = \begin{bmatrix} w_{p1} & w_{p2} \\ w_{s1} & w_{s2} \end{bmatrix}^{-1} \begin{bmatrix} a_{p1} & \dots & a_{pn} \\ a_{s1} & \dots & a_{sn} \end{bmatrix}$

**Алгоритм 3.** Неотрицательная факторизация матриц ранга 2.

## 2. Неотрицательная факторизация тензоров

### 2.1. Неотрицательный тензорный поезд

Разложение в тензорный поезд используется для компактного представления и аппроксимации многомерных массивов. Элементы неотрицательного тензорного поезда записываются как:

$$a(i_1, \dots, i_d) = \sum_{\alpha_1, \dots, \alpha_{d-1}} G_1(i_1, \alpha_1) G_2(\alpha_1, i_2, \alpha_2) \dots$$

$$\dots G_{d-1}(\alpha_{d-2}, i_{d-1}, \alpha_{d-1}) G_d(\alpha_{d-1}, i_d),$$

с вагонами  $G_1, \dots, G_d$ , среди которых любые два соседних имеют общий индекс суммирования. Индексы суммирования  $\alpha_k$  принимают значения от 1 до  $r_k$  и называются вспомогательными индексами. Величины  $r_k$  называются рангами тензорного поезда (ТТ ранги). Легче предполагать, что  $G_1, G_d$  не двумерные, а трехмерные с дополнительными вспомогательными индексами  $\alpha_0 = \alpha_d = 1$  и ТТ рангами  $r_0 = r_d = 1$ . Это предположение помогает упростить некоторые алгоритмы. Тензорные вагоны  $G_1, \dots, G_d \in \mathbb{R}_+^{r_{k-1} \times n_k \times r_k}$ .

## 2.2. Алгоритм построения неотрицательного тензорного поезда NTTF

Для того чтобы добиться неотрицательности элементов ТТ-разложения, предлагается в алгоритме ТТ-SVD применять не сингулярное разложение матриц-разверток, а один из методов решения задачи неотрицательной матричной факторизации (NMF).

Что касается неотрицательного ранга разложения, то справедлива следующая теорема:

**Теорема 6.** [21] Пусть все элементы матрицы  $V$  размерности  $m \times n$  неотрицательные:  $V \in \mathbb{R}_+^{m \times n}$ . Тогда неотрицательный ранг  $V$  ( $\text{rank}_+(V)$ ) удовлетворяет следующему неравенству:

$$\text{rank}(V) \leq \text{rank}_+(V) \leq \min(m, n).$$

Доказательство представляется очевидным. Действительно, если  $\text{rank}_+(V) < \text{rank}(V)$ , то из определения неотрицательного ранга следует, что  $\text{rank}_+(V) = \text{rank}(V)$ . Таким образом, получаем противоречие, и наше предположение было неверным. Оценка сверху выполняется, поскольку в качестве факторов для разложения неотрицательной матрицы  $V$  всегда можно взять единичную матрицу и саму  $V$ .

Обычно для поиска решения проблему NMF формулируют как задачу

оптимизации:  $\min_{W \geq 0, H \geq 0} \|V - WH\|_F^2$ , где  $\|\cdot\|_F$  - норма Фробениуса. Она не является выпуклой, то есть может существовать несколько локальных минимумов.

Для NTTF можно выбрать любой метод неотрицательной матричной факторизации, предлагается применять lraNMF, предложенный в [36]. В этой работе авторы проанализировали два известных алгоритма NMF (мультипликативные обновления и HALS) и заметили, что сложность каждой итерации больше  $O(mn)$ . Операция, требующая больше всего времени, – умножение матриц на исходную большую матрицу  $V$ . Поэтому авторы предложили заменить ее малоранговой аппроксимацией, чтобы повысить эффективность метода. Они рассмотрели следующую задачу оптимизации:

$$\min_{\widetilde{W}, \widetilde{H}, W, H} \|V - \widetilde{W}\widetilde{H}\|_F^2 + \|\widetilde{W}\widetilde{H} - WH\|_F^2,$$

где  $\widetilde{W} \in \mathbb{R}^{m \times s}$ ,  $\widetilde{H} \in \mathbb{R}^{s \times n}$ ,  $W \in \mathbb{R}^{m \times r} \geq 0$ ,  $H \in \mathbb{R}^{r \times n} \geq 0$ .

Авторы [36] использовали двухэтапный метод, чтобы решить ее. Сначала строится малоранговое разложение  $V$  с маленьким рангом  $s$ . Затем решается задача NMF  $\min_{W, H} \|\widetilde{W}\widetilde{H} - WH\|_F^2$ . Один из алгоритмов, предложенных в [36], называется lraNMF\_HALS. Сложность каждой итерации  $O((m+n)r^2)$ , где  $r$  - ранг неотрицательной факторизации. Авторы утверждают, что lraNMF\_HALS сходится к стационарной точке. В [36] также приводится некоторый анализ того, как ошибка малоранговой аппроксимации влияет на итоговую ошибку lraNMF.

В диссертации предлагается применять методы крестовой малоранговой аппроксимации [7]-[10], которые имеют сложность  $O((m+n)s^2)$  и используют для построения  $O((m+n)s)$  элементов исходной матрицы  $V$ . Поэтому применение крестового метода с lraNMF\_HALS позволяет значи-

тельно уменьшить число операций.

**Дано:** Тензор  $A(i_1, \dots, i_d)$ , точность  $\varepsilon$ , максимальное число итераций  $maxIter$ .

**Найти:** Неотрицательный ТТ  $B(i_1, \dots, i_d)$  с вагонами  $G_1, \dots, G_d$ .

```

1  $C = A; r_0 = 1;$ 
2 for  $i = 1:(d-1)$ 
3    $C = \text{reshape}(C, [r_{i-1}n_i, \frac{\text{numel}(C)}{r_{i-1}n_i}]);$ 
4   Первая оценка неотрицательного ранга:  $k = \text{rank}(C);$ 
5   while  $k < \min(\text{size}(C))$  do
6      $iter = 1;$ 
7     Найти факторизацию ранга  $k$ :  $C = WH + E, W \geq 0, H \geq 0.$ 
8     while  $\|E\| > \frac{\varepsilon}{d-1} \wedge iter \leq maxIter$  do
9       Найти факторизацию ранга  $k$ :
10       $C = WH + E, W \geq 0, H \geq 0.$ 
11       $iter = iter + 1;$ 
12    end
13    if  $\|E\| < \frac{\varepsilon}{d-1}$  then
14       $break;$ 
15    end
16     $k = k + 1;$ 
17  end
18  if  $\|E\| > \frac{\varepsilon}{d-1}$  then
19    if  $(\text{size}(C, 1) > \text{size}(C, 2))$  then
20       $W = C; H = \text{eye}(k);$ 
21    else
22       $W = \text{eye}(k); H = C;$ 
23    end
24  end
25   $r_{i+1} = k;$ 
26   $H = \|W\|_F H;$ 
27   $W = \frac{W}{\|W\|_F};$ 

```

```

27
28 | New core:  $G_i = \text{reshape}(W, [r_{i-1}, n_i, r_i]);$ 
29 |  $C = H;$ 
30 end
31  $G_d = C;$ 

```

#### Алгоритм 4. Псевдокод NTTF.

Алгоритм 4 – псевдокод метода NTTF (nonnegative tensor train factorization). Для описания используются стандартные функции Matlab. Функция  $\text{reshape}(A, [n_1, n_2, \dots, n_t])$  возвращает массив размера  $n_1 \times n_2 \times \dots \times n_t$ , элементы которого берутся по столбцам  $A$ . Команда  $\text{eye}(m)$  возвращает  $m \times m$  единичную матрицу, а  $\text{rank}(C)$  - ранг  $C$ .

В алгоритме 4 итеративно используется некоторый NMF метод для факторизации матриц-разверток, и это гарантирует поэлементную неотрицательность вагонов тензорного поезда. Но NMF алгоритмы требуют, чтобы неотрицательный ранг факторизации подавался на вход. Так как единственные известные оценки для него дает теорема 6, сначала переменная  $k$  приравнивается рангу матрицы-развертки. Затем делается попытка найти неотрицательную факторизацию ранга  $k$  с точностью  $\frac{\varepsilon}{d-1}$ . Если это удалось, то найден новый неотрицательный  $TT$ -ранг. В противном случае  $k$  увеличивается на 1 и процесс повторяется. Таким образом, проверяются значения для неотрицательного ранга одно за другим. Другой способ - использовать бинарный поиск, что и было сделано для численных экспериментов. Здесь алгоритм 4 формулируется для любого метода NMF, если использовать комбинацию lraNMF с крестовым методом, то в качестве начального значения для  $k$  можно взять ранг малоранговой аппроксимации.

В теории NMF алгоритмы сходятся к локальному минимуму. Это означает, что результаты могут значительно отличаться при изменении матриц-факторов для инициализации. Чтобы избежать возможной проблемы с “плохой” инициализацией, NMF метод может запускаться несколько раз (в алгоритме 4 максимальное число итераций определяется переменной  $\text{maxIter}$ ).

### 2.3. Оценка точности аппроксимации, построенной методом NTTF

**Теорема 7.** Алгоритм 4 строит тензорный поезд  $B$  так, что

$$\|A - B\|_F \leq \varepsilon.$$

**Доказательство.** При  $d = 2$  имеем случай матриц, и утверждение следует из алгоритма построения  $B$ .

Пусть  $d > 2$ . Тогда первая матрица-развертка  $A_1$  будет представлена  $A_1 = W_1 H_1 + E_1$ , где  $\|E_1\| \leq \frac{\varepsilon}{d-1}$ ,  $W_1 \in \mathbb{R}^{n_1 \times r_1}$  и  $\|W_1\|_F = 1$ . Матрица  $H_1$  естественным образом ассоциируется с тензором с элементами  $H_1(\alpha_1 i_2, i_3, \dots, i_d)$ , который далее будет разложен по алгоритму. Значит,  $H_1$  будет приближена произведением некоторых  $W_2$  и  $H_2$ , причем  $\|H_1 - W_2 H_2\|_F \leq \frac{\varepsilon}{d-1}$ ,  $\|W_2\|_F = 1$ . Тогда на второй итерации алгоритма получаем следующую оценку:

$$\begin{aligned} \|A_1 - W_1 W_2 H_2\|_F &= \|A_1 - W_1(W_2 H_2 - H_1 + H_1)\|_F \leq \\ &\leq \|A_1 - W_1 H_1\|_F + \|W_1(H_1 - W_2 H_2)\|_F \leq \frac{\varepsilon}{d-1} + \|W_1\|_F \|W_2 H_2 - H_1\|_F \leq \\ &\leq 2 \frac{\varepsilon}{d-1}. \end{aligned}$$

Можно воспользоваться для доказательства методом индукции. Пусть на  $n$ -ой итерации ( $n < d - 1$ ) выполнено:

$$\|A_1 - W_1 \dots W_n H_n\|_F \leq n \frac{\varepsilon}{d-1}.$$

Значит, на  $(n+1)$ -ом шаге для  $H_n$  будет получена аппроксимация:  $\|H_n - W_{n+1} H_{n+1}\|_F \leq \frac{\varepsilon}{d-1}$ ,  $\|W_{n+1}\|_F = 1$ . Отсюда следует, что:

$$\begin{aligned} \|A_1 - W_1 \dots W_{n+1} H_{n+1}\|_F &= \|A_1 - W_1 \dots W_n (W_{n+1} H_{n+1} - H_n + H_n)\|_F \leq \\ &\leq \|A_1 - W_1 \dots W_n H_n\|_F + \|W_1 \dots W_n (W_{n+1} H_{n+1} - H_n)\|_F \leq n \frac{\varepsilon}{d-1} + \end{aligned}$$

$$+\|W_1\|_F \dots \|W_n\|_F \|H_n - W_{n+1}H_{n+1}\|_F \leq (n+1) \frac{\varepsilon}{d-1}.$$

Таким образом на  $(d-1)$ -ой (последней итерации) будет получен неотрицательный ТТ  $B(i_1, \dots, i_d)$ , для которого  $\|A - B\|_F \leq \varepsilon$ .

## 2.4. Алгоритм построения неотрицательного тензорного поезда NTT-MU

Чтобы построить неотрицательный тензорный поезд, предлагается использовать методы неотрицательной факторизации матриц. Напомним, как формулируется задача неотрицательной матричной факторизации (NMF) : дана  $m \times n$  матрица  $V$  с  $V_{ij} \geq 0, i = \overline{1, m}, j = \overline{1, n}$  (далее неотрицательность матричных элементов будет обозначаться  $V \geq 0$ ) и натуральное число  $r < \min(m, n)$ , требуется найти матрицы  $W \in \mathbb{R}^{m \times r} \geq 0$  и  $H \in \mathbb{R}^{r \times n} \geq 0$  такие, что  $V \approx WH$ . Минимальное такое  $r$ , что  $V = WH$  называется неотрицательным рангом  $V$ . Обычно ищут  $W$  и  $H$ , минимизируя расстояние между  $V$  и  $WH$ :

$$\min_{W \geq 0, H \geq 0} \|V - WH\|_F^2, \quad (5)$$

где  $\|\cdot\|_F$  - норма Фробенуса.

Один из наиболее известных подходов к решению - метод мультипликативных обновлений [2]. Он очень популярен благодаря своей простоте.

Из условия Каруша-Куна-Таккера (W;H) является стационарной точкой задачи 5 тогда и только тогда, когда

$$W \geq 0, H \geq 0, \quad (6)$$

$$\nabla_W \|V - WH\|_F^2 \geq 0, \nabla_H \|V - WH\|_F^2 \geq 0, \quad (7)$$

$$W \circ \nabla_W \|V - WH\|_F^2 = 0, H \circ \nabla_H \|V - WH\|_F^2 = 0. \quad (8)$$

Знак "  $\circ$  " обозначает произведение Адамара, и

$$\nabla_W \|V - WH\|_F^2 = -2(V - WH)H^T, \nabla_H \|V - WH\|_F^2 = -2W^T(V - WH). \quad (9)$$

Если подставить (9) в (8), то получим:

$$W \circ (WHH^T) = W \circ (VH^T), \quad (10)$$

$$H \circ (W^TWH) = H \circ (W^TV). \quad (11)$$

Из этих уравнений Ли и Сеунг выписали простые правила обновления  $W$  и  $H$  на каждом шаге ( $\frac{[\cdot]}{[\cdot]}$  - деление Адамара, т.е. покомпонентное):

$$W \leftarrow W \circ \frac{[VH^T]}{[WHH^T]}, H \leftarrow H \circ \frac{[W^TV]}{[W^TWH]}, \quad (12)$$

для которых они доказали свойство монотонности:

**Теорема 8.** [35] Норма Фробениуса  $\|V - WH\|_F$  не возрастает при обновлении матриц  $W, H$  по правилам (12).

Описанный метод называется методом мультипликативных обновлений. Он прост в реализации и на практике дает хорошие результаты.

Предлагается метод для вычисления неотрицательного ТТ разложения, имеющий в основе некоторые идеи из [5]. В этой работе авторы представили метод NTT-HALS. Он используется для построения аппроксимации данного тензора неотрицательным тензорным поездом. ТТ-ранги считаются входными параметрами, сперва тензор инициализируется с помощью метода чередующихся наименьших квадратов, а затем улучшается с помощью итеративной процедуры с использованием иерархических наименьших квадратов (HALS). У этого алгоритма есть два основных недостатка. ТТ-ранги должны быть известны заранее, а их оценка - очень сложная задача. Также сложность NTT-HALS может быть очень высокой, поэтому он не подходит для работы с большими данными. На каждой итерации  $k$ ,  $k = \overline{1, d}$  нужно умножать 2 матрицы размером  $n_k \times n_1 \dots n_{k-1} n_{k+1} \dots n_d$  и  $n_1 \dots n_{k-1} n_{k+1} \dots n_d \times r_{k-1} r_k$ , что имеет сложность  $O(n^d)$ , где  $n = \max_i(n_i)$ ,  $i = \overline{1, d}$ . В новом алгоритме предлагается способ избежать одной из этих проблем.

Пусть  $B$  в виде неотрицательного ТТ - аппроксимация для исходного

тензора  $A \in \mathbb{R}_+^{n_1 \times n_2 \times \dots \times n_d}$ . Наша цель - минимизировать

$$\|A - B\|_F^2. \quad (13)$$

Для решения этой задачи, имея некое начальное приближение, предлагается последовательно обновлять вагоны  $B$ , предполагая, что остальные вагоны зафиксированы.

Для описания нового алгоритма будут использоваться следующие обозначения:

$$G_{<k} = G_{(1)} \times^1 G_{(2)} \times^1 \dots \times^1 G_{(k-1)} \in \mathbb{R}_+^{n_1 \times \dots \times n_{k-1} \times r_{k-1}},$$

$$G_{>k} = G_{(k+1)} \times^1 G_{(k+2)} \times^1 \dots \times^1 G_{(d)} \in \mathbb{R}_+^{r_k \times n_{k+1} \times \dots \times n_d},$$

и  $G_{<1} = G_{>d} = 1$ . Операция “ $\times^1$ ” обозначает умножение по моде 1 двух тензоров, так что  $(X \times^1 Y)_{i_1, \dots, i_{d-1}, j_2, \dots, j_s} = \sum_{i_d=1}^{n_d} x(i_1, \dots, i_d) y(i_d, j_2, \dots, j_s)$ , где  $X \in \mathbb{R}_+^{n_1 \times n_2 \times \dots \times n_d}$ ,  $Y \in \mathbb{R}_+^{n_d \times m_2 \times \dots \times m_s}$ . Следовательно  $B$  может быть записан как  $B = G_{<k} \times^1 G_k \times^1 G_{>k}$ . Назовем матрицу-развертку по моде  $k$  тензора  $A$  как  $A^{(k)} \in \mathbb{R}_+^{n_k \times n_1 \dots n_{k-1} n_{k+1} \dots n_d}$ , тогда

$$B^{(k)} = G_k^{(2)} (G_{>k}^{(1)} \otimes G_{<k}^{(k)}),$$

где “ $\otimes$ ” отвечает произведению Кронекера двух матриц.

Функция цены 13 может быть переписана как  $\|A^{(k)} - B^{(k)}\|_F^2$ . Тогда обновление вагона  $G_k$  означает, что нужно найти такую новую матрицу  $G_k^{*(2)}$  с неотрицательными элементами, что она минимизирует  $\|A^{(k)} - G_k^{*(2)} (G_{>k}^{(1)} \otimes G_{<k}^{(k)})\|_F^2$ . Это стандартная подзадача в методе чередующихся наименьших квадратов для решения NMF, и решение может быть записано в форме мультипликативного правила обновления для  $W$  из 12:

$$G_k^{(2)} \leftarrow G_k^{(2)} \circ \frac{[A^{(k)} (G_{>k}^{(1)} \otimes G_{<k}^{(k)})^T]}{[G_k^{(2)} ((G_{>k}^{(1)} (G_{>k}^{(1)})^T) \otimes (G_{<k}^{(k)} (G_{<k}^{(k)})^T))]} \quad (14)$$

Однако здесь возникает та же проблема, что и в NTT-HALS из [5]: если тензор  $A$  дан в полном формате, то умножение его матрицы-равертки

на матрицу потребует слишком много ресурсов и времени. Поэтому для больших данных нужно найти некоторое альтернативное решение.

Пусть имеется некое представление  $A$  в виде тензорного поезда (без условия неотрицательности):

$$a(i_1, \dots, i_d) = \sum_{\alpha_0, \dots, \alpha_d} \tilde{G}_1(\alpha_0, i_1, \alpha_1) \tilde{G}_2(\alpha_1, i_2, \alpha_2) \dots \\ \dots \tilde{G}_{d-1}(\alpha_{d-2}, i_{d-1}, \alpha_{d-1}) \tilde{G}_d(\alpha_{d-1}, i_d, \alpha_d),$$

где вагоны (ядра)  $\tilde{G}_i \in \mathbb{R}^{\tilde{r}_{k-1} \times n_k \times \tilde{r}_k}$ ,  $\tilde{r}_0 = \tilde{r}_d = 1$ .

Тогда 14 можно переписать:

$$G_k^{(2)} \leftarrow G_k^{(2)} \circ \frac{[\tilde{G}_k^{(2)} ((\tilde{G}_{>k}^{(1)}(G_{>k}^{(1)})^T) \otimes (\tilde{G}_{<k}^{(k)}(G_{<k}^{(k)})^T))]}{[G_k^{(2)} ((G_{>k}^{(1)}(G_{>k}^{(1)})^T) \otimes (G_{<k}^{(k)}(G_{<k}^{(k)})^T))]} \quad (15)$$

Это правило обновления будет верным, если есть точное представление  $A$  в форме ТТ. В случае аппроксимации, чтобы избежать появления отрицательных элементов, нужно немного изменить 15:

$$G_k^{(2)} \leftarrow G_k^{(2)} \circ \frac{[[\tilde{G}_k^{(2)} ((\tilde{G}_{>k}^{(1)}(G_{>k}^{(1)})^T) \otimes (\tilde{G}_{<k}^{(k)}(G_{<k}^{(k)})^T))]]_+]}{[G_k^{(2)} ((G_{>k}^{(1)}(G_{>k}^{(1)})^T) \otimes (G_{<k}^{(k)}(G_{<k}^{(k)})^T))]} \quad (16)$$

где  $[\cdot]_+$  меняет отрицательные аргументы на некоторую очень маленькую неотрицательную константу  $\rho$ . Такой подход был использован в [36] применительно к матрицам, где авторы также обсудили вопрос сходимости.

Ниже представлен псевдокод метода NTT-MU.

**Дано:** Тензор  $A(i_1, \dots, i_d)$ ,  $\rho \geq 0$ , неотрицательные ТТ-ранги  $r_i$ ,  
 $i = \overline{1, d-1}$ .

**Найти:** Неотрицательный ТТ  $B(i_1, \dots, i_d)$  с вагонами  $G_1, \dots, G_d$ .

```

1 Найти некоторое приближение  $\tilde{A}$  для  $A$  в форме неотрицательного
  ТТ с вагонами  $\tilde{G}_1, \dots, \tilde{G}_d$  и ТТ-рангами  $\tilde{r}_i$ ,  $i = \overline{1, d-1}$ 
2 Инициализировать  $B$  - неотрицательный ТТ с ТТ-рангами  $r_i$ ,
   $i = \overline{1, d-1}$ .
3  $G_{<1} = G_{>d} = 1$ ;
4 while TRUE do
5   for (  $k = (d-1):1$  ) {
6     Вычислить  $Q_{>k} = G_{>k}^{(1)}(G_{>k}^{(1)})^T$ ;
7     Вычислить  $\tilde{Q}_{>k} = \tilde{G}_{>k}^{(1)}(\tilde{G}_{>k}^{(1)})^T$ ;
8   }
9   for (  $k = 1:d$  ) {
10    Вычислить  $Q_{<k} = G_{<k}^{(k)}(G_{<k}^{(k)})^T$ ;
11    Вычислить  $\tilde{Q}_{<k} = \tilde{G}_{<k}^{(k)}(\tilde{G}_{<k}^{(k)})^T$ ;
12     $numerator = \tilde{Q}_{>k} \otimes \tilde{Q}_{<k}$ ;
13     $numerator = [\tilde{G}_k^{(2)} * numerator]_+$ ; //  $[\cdot]_+$  меняет
      отрицательные элементы на  $\rho$ 
14     $denominator = Q_{>k} \otimes Q_{<k}$ ;
15     $denominator = G_k^{(2)} * denominator$ ;
16     $updG = G_k^{(2)} \circ \frac{[numerator]}{[denominator]}$ ; // Матрица-развертка
      обновленного вагона по моде 2
17    Переформировать  $updG$  в  $r_{k-1} \times n_k \times r_k$  и сохранить как
      вагон  $G_k$ ;
18  }
19  if критерий останова выполнен then
20    break;
21  end
22 end

```

**Алгоритм 5.** Псевдокод NTT-MU.

ТТ аппроксимация  $\tilde{A}$  исходного тензора может быть найдена с помо-

щью, например, TT-SVD или TT-CROSS ([34]). И это приближение нужно вычислить лишь раз.

Для инициализации  $B$  можно использовать разные методы. Например, можно применить ALS, как это было сделано в [5]. Отрицательные элементы в получившихся вагонах заменялись их модулем. Другой способ - использовать метод NTTF (Алгоритм 4). Также можно инициализировать  $B$  аппроксимацией  $\tilde{A}$ , найденной на предыдущем шаге, и заменить отрицательные элементы в вагонах  $\tilde{G}_1, \dots, \tilde{G}_d$  на их модули. Этот способ наиболее выгодный с вычислительной точки зрения. Интересное направление для будущего исследования - проверить, как метод инициализации влияет на результат.

Оценим сложность обновления одного ядра. Матрицы  $Q_{<k}$ ,  $Q_{>k}$ ,  $\tilde{Q}_{<k}$ ,  $\tilde{Q}_{>k}$  можно вычислить эффективно, алгоритм аналогичен расчету скалярного произведения (с реализацией можно ознакомиться в [28]). И так как расчет  $(k+1)$ -ой зависит от  $k$ -ой матрицы, можно сэкономить время, вычисляя общие части лишь раз. Поэтому сложность оценивается как  $O(dnr^3)$ ,  $n = \max_i(n_i)$ . Отметим, что можно вычислить все  $Q_{>k}$  и  $\tilde{Q}_{>k}$  заранее, так как обновление  $k$ -ого вагона не меняет  $Q_{>k+1}$  и  $\tilde{Q}_{>k+1}$ .

Сложность произведения Кронекера двух маленьких матриц оценивается как  $O(r^4)$ ,  $r = \max_i(r_i, \tilde{r}_i)$ . Вычисление числителя требует  $O(nr^4)$  операций. Заключаем, что сложность обновления вагона  $O(nr^4)$ , и это очень хорошо, поскольку зависимость от  $n$  линейная.

Процесс обновления всех вагонов назовем проходом. В алгоритме 5 рассматриваются проходы слева направо, но можно переписать формулы для обновления справа налево.

Критерий останова можно определять по-разному. Можно остановить процесс, когда выполнено максимальное число итераций или достигнута желаемая точность ( $\|\tilde{A} - B\|_F^2 < \varepsilon$ ). Или если сходимость значительно замедлилась. Можно комбинировать все эти критерии.

## 2.5. Комбинация алгоритма NTT-MU и методики DMRG

В предыдущей секции говорилось, что один из недостатков NTT-HALS из [5] (а значит и Алгоритма 5) - требование, чтобы неотрицательные ТТ-ранги были известны. В реальности обычно такой информации нет в наличии. Недооценивание рангов ведет к плохой неотрицательной аппроксимации, если они переоценены, то сложность существенно возрастает. Поэтому в наших интересах найти способ определять неотрицательные ТТ-ранги для  $B$  в алгоритме 5 “на лету”.

Для решения этой проблемы предлагается использовать “Density Matrix Renormalization Group” подход из физики твердого тела, который был успешно применен в алгоритмах для тензорных поездов, например ALS [40]. Идея заключается в оптимизации относительно двух вагонов  $G_k, G_{k+1}$  одновременно. Введем обозначение для супервагона

$$W(k, k + 1) = G_k \times^1 G_{k+1}. \quad (17)$$

Тогда  $r_k$  не требуется знать. Когда вычислен  $W(k, k + 1)$ , находятся  $G_k, G_{k+1}$  из 17 с помощью какого-либо NMF метода. Так неотрицательный ТТ-ранг  $r_k$  определяется адаптивно.

Псевдокод для комбинации NTT-MU с техникой DMRG представлен

как алгоритм 6 .

**Дано:** Тензор  $A(i_1, \dots, i_d)$ ,  $\rho \geq 0$ ,  $\tilde{\varepsilon} > 0$ , начальные неотрицательные ТТ-ранги  $r_i$ ,  $i = \overline{1, d-1}$ .

**Найти:** Неотрицательный ТТ  $B(i_1, \dots, i_d)$  с вагонами  $G_1, \dots, G_d$ .

- 1 Найти некоторое приближение  $\tilde{A}$  для  $A$  в форме неотрицательного ТТ с вагонами  $\tilde{G}_1, \dots, \tilde{G}_d$  и ТТ-рангами  $\tilde{r}_i$ ,  $i = \overline{1, d-1}$
- 2 Инициализировать  $B$  - неотрицательный ТТ с ТТ-рангами  $r_i$ ,  $i = \overline{1, d-1}$ .
- 3  $G_{<1} = G_{>d} = 1$ ;
- 4 **while** *TRUE* **do**
  - 5 **for** (  $k = (d-1):1$  ) {
  - 6 | Вычислить  $Q_{>k} = G_{>k}^{(1)}(G_{>k}^{(1)})^T$ ;
  - 7 | Вычислить  $\tilde{Q}_{>k} = \tilde{G}_{>k}^{(1)}(\tilde{G}_{>k}^{(1)})^T$ ;
  - 8 | }
  - 9 **for**  $k = 1:(d-1)$ 
    - 10 | Вычислить  $Q_{<k} = G_{<k}^{(k)}(G_{<k}^{(k)})^T$ ;
    - 11 | Вычислить  $\tilde{Q}_{<k} = \tilde{G}_{<k}^{(k)}(\tilde{G}_{<k}^{(k)})^T$ ;
    - 12 | Вычислить  $W = G_k \times^1 G_{k+1}$  и переформировать его в матрицу  $W(k, k+1) \in \mathbb{R}_+^{n_k n_{k+1} \times r_{k-1} r_{k+1}}$ ;
    - 13 | Вычислить  $\tilde{W} = \tilde{G}_k \times^1 \tilde{G}_{k+1}$  и переформировать его в матрицу  $\tilde{W}(k, k+1) \in \mathbb{R}_+^{n_k n_{k+1} \times \tilde{r}_{k-1} \tilde{r}_{k+1}}$ ;
    - 14 |  $numerator = \tilde{Q}_{>k+1} \otimes \tilde{Q}_{<k}$ ;
    - 15 |  $numerator = [\tilde{W}(k, k+1) * numerator]_+$  ; //  $[\cdot]_+$  меняет отрицательные элементы на  $\rho$
    - 16 |  $denominator = Q_{>k+1} \otimes Q_{<k}$ ;
    - 17 |  $denominator = W(k, k+1) * denominator$ ;
    - 18 |  $updW = W(k, k+1) \circ \frac{[numerator]}{[denominator]}$  ; // Обновленный супервагон

```

19
20
21     Переформировать  $updW$  to  $n_k r_{k-1} \times n_{k+1} r_{k+2}$ ;
22     Найти новую факторизацию ранга  $r_k$ :
         $updW = WH + E, \frac{\|E\|}{\|updW\|} \leq \tilde{\varepsilon}, W \geq 0, H \geq 0.$ 
23     Переформировать  $W$  и  $H$  в  $r_{k-1} \times n_k \times r_k$  и  $r_k \times n_{k+1} \times r_{k+2}$ 
        соответственно.
24     Сохранить  $W, H$  как новые  $G_k, G_{k+1}$ .
25     end
26     if выполнен критерий останова then
27         | break;
28     end
29 end

```

### Алгоритм 6. Псевдокод NTT-MU с DMRG.

Единственные оценки неотрицательного ранга в задаче 5 в общем случае [21]:

$$\text{rank}(V) \leq \text{rank}_+(V) \leq \min(m, n).$$

И поиск неотрицательного ранга факторизации - NP-трудная задача. Поэтому обычно методы NMF требуют неотрицательный ранг в качестве входного параметра. В алгоритме 6 предлагается организовать бинарный поиск  $r_k$  между значениями ранга матрицы и минимума из ее размеров, чтобы построить NMF с нужной точностью.

Для получения вагонов из супервагона любой алгоритм NMF может быть использован. Если размер супервагона большой, рекомендуется применять lraNMF – NMF метод, основанный на малоранговой аппроксимации, был предложен в [36]. Подробно об этом методе рассказывалось в 2.2.

В алгоритме 5 была использована та же идея, что и в [36], а именно замена исходного тензора на его разложение в ТТ.

Использование крестового метода с алгоритмом lraNMF\_HALS поможет значительно уменьшить число операций для неотрицательной факторизации супервагонов.

## 2.6. Построение тензорного поезда с неотрицательными элементами с помощью коррекции элементов

Для неотрицательных тензорных разложений стандартом является требование неотрицательности элементов в матрицах/тензорах-факторах. Для этого есть две основные причины. Интуитивно ожидается, что части, представленные факторами, суммируются для получения неотрицательного результата. И во многих приложениях требование неотрицательности позволяет интерпретировать результаты разложения. Вторая причина гораздо более практична: добавление такого ограничения позволяет нам гарантированно получить неотрицательную аппроксимацию.

Однако бывают ситуации, когда неотрицательность ядер не является необходимой, так как мы заинтересованы только в элементах результирующей аппроксимации. Разложение тензоров - операция, позволяющая управлять большими объемами данных эффективно. Многие алгоритмы могут использовать такие представления для ускорения вычислений. Но при этом существуют примеры, когда именно неотрицательность тензорного разложения является достаточным условием для сходимости алгоритмов. Поэтому, когда главная цель заключается в том, чтобы получить все плюсы, которые дает тензорная декомпозиция, такие как меньший расход памяти и более быстрые расчеты, и одновременно сохранить свойства исходного тензора, то имеет смысл рассмотреть следующую задачу.

Для неотрицательного тензора требуется построить такое неотрицательное разложение в тензорный поезд (неотрицательное ТТ-разложение), что:

$$y(i_1, \dots, i_d) = \sum_{\alpha_1, \dots, \alpha_{d-1}} G_1(i_1, \alpha_1) G_2(\alpha_1, i_2, \alpha_2) \dots \\ \dots G_{d-1}(\alpha_{d-2}, i_{d-1}, \alpha_{d-1}) G_d(\alpha_{d-1}, i_d) \geq 0,$$

где вагоны (ядра)  $G_1, \dots, G_d \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$ . Так как снимается ограничение на неотрицательность факторов, то можно надеяться, что решение задачи будет обладать либо меньшей сложностью, либо более высокой точностью.

Предлагается новый подход к неотрицательной тензорной факторизации. Когда строится тензорное разложение с заданной точностью, ожида-

ется, что в итоге все элементы аппроксимации будут близки к исходным, и если исходные данные были неотрицательными, то естественно предположить, что большая часть элементов аппроксимации таковыми и останутся. Тогда, если число отрицательных элементов мало, то можно найти их и поправить.

Поэтому прежде всего необходим метод, способный найти отрицательные элементы в аппроксимации тензорным поездом. Для начала рассмотрим задачу, как определить максимальный по модулю элемент. Конечно, поэлементный перебор неприемлем, так как его сложность  $O(N^d)$ , где  $N = \max_i(n_i)$ . Отметим, что для диагональной матрицы  $A$  ее максимальный по модулю элемент одновременно является ее собственным значением с максимальной абсолютной величиной. Один из самых простых и известных алгоритмов для поиска таких значений - степенной метод. В степенном методе изначально выбирается произвольный вектор  $r_0$ . И далее по итеративной формуле вычисляется вектор  $r_{k+1} = \frac{Mr_k}{\|Mr_k\|}$ , который при некоторых условиях будет сходиться к собственному вектору, отвечающему доминантному собственному значению. Но наша цель - найти минимальное значение. И это можно сделать, если применить степенной метод к матрице  $A - \lambda_{max}I$ , где  $\lambda_{max}$  - максимальное по модулю собственное значение,  $I$  - единичная матрица.

Вернемся к тензорам. Существует несколько подходов к поиску собственных значений в тензорном поезде. Они включают итеративные методы с дополнительным округлением ранга [52], оптимизацию по Риману и метод переменных направлений. Разберем, как можно найти отрицательные элементы в ТТ-аппроксимации с помощью классического метода поиска собственных значений. Рассмотрим тензорный поезд  $\tilde{Y}$  как диагональ диагональной матрицы. Тогда вычисление  $r_{k+1} = \frac{Mr_k}{\|Mr_k\|}$  существенно ускоряется. Обсудим детально каждый шаг процесса и оценим итоговую сложность.

Прежде всего используем следующий факт:

$$Ba = b \otimes a, \quad B \in \mathbb{R}^{m \times m}, \quad a \in \mathbb{R}^m, \quad b \in \mathbb{R}^m,$$

где  $\otimes$  обозначает произведение Адамара,  $B$  - диагональная матрица с диагональю  $b$ . И если векторы  $b$  и  $a$  в ГТ формате, то можно выполнять поэлементное умножение достаточно быстро.

Произведение Адамара  $C$  двух тензорных поездов  $b$  и  $a$  может быть записано как

$$\begin{aligned} C(i_1, \dots, i_d) &= A_1(i_1) \dots A_d(i_d) B_1(i_1) \dots B_d(i_d) = \\ &= (A_1(i_1) \dots A_d(i_d)) \otimes (B_1(i_1) \dots B_d(i_d)) = (A_1(i_1) \otimes B_1(i_1)) \\ &\quad (A_2(i_2) \otimes B_2(i_2)) \dots (A_d(i_d) \otimes B_d(i_d)). \end{aligned}$$

Символ  $\otimes$  обозначает произведение Кронекера. Это означает, что ядра  $C$  имеют вид

$$C_k(i_k) = A_k(i_k) \otimes B_k(i_k), \quad k = \overline{1, d},$$

где  $B_k(i_k)$  и  $A_k(i_k)$  – матрицы размера  $r'_{k-1} \times r'_k$  и  $r''_{k-1} \times r''_k$  соответственно. Поэтому общая сложность оценивается как  $O(dNr^4)$ .

Но ранги ядер  $C$  равны  $r'_{k-1}r''_{k-1}$  и  $r'_kr''_k$ ,  $k = 0, \dots, d$ . Это означает, что с каждой итерацией вычислительная сложность будет расти. Чтобы избежать роста рангов, требуется их уменьшать, при этом сохраняя точность, и это возможно сделать с помощью алгоритма ГТ-округления [4]. Так как ранги равны  $r^2$ , то сложность составит  $O(dNr^6)$ . Далее необходимо вычислить норму тензорного поезда, сложность данной операции  $O(dNr^3)$  [4]. Оставшийся шаг – деление на число – тривиален; нужно лишь поделить на это число элементы одного из ядер. Ниже представлен псевдокод для

степенного метода.

**Дано:** Тензор  $\tilde{Y}$ , число итераций  $niters$

**Найти:** Вектор  $x_{niters}$  в ТТ формате

- 1 Инициализируем  $x_1$  как ТТ из одних единиц того же размера, что и  $Y$ .
- 2 **for** (  $i = 1:niters$  ) {
- 3      $x_{i+1} = \tilde{Y} \otimes x_i$
- 4     ТТ-округление( $x_{i+1}$ )
- 5     Нормировать  $x_{i+1}$
- 6 }

**Алгоритм 7.** Псевдокод для степенного метода в ТТ формате.

Если у тензора только одно ведущее собственное число, то в точной арифметике метод должен сходиться к тензору с рангами 1. Для получения отвечающего собственного значения можно использовать отношение Рэлея.

Следует отметить, что, помимо алгоритмов вычисления собственных значений, есть и другие методы поиска минимального элемента в тензорном поезде. Например, в [28] представлена процедура, вычисляющая статистику, такую как наибольшее и наименьшее по модулю элементы или элементы с наибольшей/наименьшей действительными частями в тензорном поезде с помощью алгоритма крестовых методов. При проведении экспериментов данный подход показал результаты, не уступающие степенному методу.

Одним из направлений будущих исследований является тестирование других методов поиска собственных значений (обратный степенной метод, итеративный метод Рэлея и т.д.) и сравнение их результатов. Конечно, каждый метод имеет свои плюсы и минусы. Например, метод обратной итерации демонстрирует быструю сходимость, но только если выбраны оптимальные сдвиги. И на каждой итерации требуется решить линейную систему.

Но обнаружение минимального элемента  $\tilde{y}(k_1, \dots, k_d)$  в ТТ - только часть решения. Наша главная цель - аппроксимация в виде тензорного поезда с только неотрицательными значениями. Конечно, если минимум больше нуля, тогда нам не нужно далее ничего вычислять, так как решение уже найдено. Но обычно это не выполняется. Когда нам известно, какой

элемент является отрицательным, можно специально приравнять его желаемому значению, прибавив тензор ранга 1. У этого тензора  $C$  все элементы нули, кроме одного:  $c(k_1, \dots, k_d) = y(k_1, \dots, k_d) - \tilde{y}(k_1, \dots, k_d)$ . После коррекции элементов процесс повторяется: находится минимум в исправленном тензорном поезде и корректируется, если он оказывается отрицательным. Таким образом в итоге будет построен ТТ с только неотрицательными значениями. Однако такой подход имеет серьезный недостаток: для суммы двух ТТ их ранги также складываются. Поэтому на каждой итерации ранги будут увеличиваться на единицу. И, в отличие от степенного метода, в данном случае нельзя применить ТТ-округление, потому что при этом не сохраняется неотрицательность элементов тензора. Чтобы избежать неконтролируемого роста рангов, предлагается сперва добавить некоторую маленькую константу ко всем элементам ТТ аппроксимации. Выбор ее значения должен зависеть от определенного минимума. Логично ожидать, что отрицательные элементы аппроксимации неотрицательного тензора будут близки к нулю. Тогда добавление тензора-константы не только позволит сделать большинство элементов положительными, но также не сильно повлияет на точность аппроксимации. После этого шага можно возобновить индивидуальную коррекцию элементов.

Суммируем предложенные шаги для построения приближения в формате тензорного поезда с неотрицательными элементами в алгоритм.

- Построить аппроксимацию тензорным поездом  $\tilde{Y}$  для исходного тензора  $Y$ . Она может быть найдена с помощью, например, методов ТТ-SVD или ТТ-CROSS ([34]). Приближение требуется вычислить лишь один раз.
- Найти минимальный элемент  $\tilde{Y}$ . Возможны различные подходы к этому шагу. Например, можно найти максимальное по модулю значение с помощью степенного метода, вычесть его из ТТ и повторить поиск поиск для этого нового тензорного поезда с только отрицательными значениями.
- К  $\tilde{Y}$  добавляется маленькая константа, чье значение зависит от найденного минимума. С помощью этого действия корректируются большин-

ство отрицательных элементов в аппроксимации. Ранги увеличатся на единицу.

- Продолжить поиск оставшихся отрицательных значений, определить их индексы и затем построить ТТ с равными 1 рангами для добавления к аппроксимации.

Конечно, остается большой простор для будущих исследований. Ведется работа по анализу распределения отрицательных элементов с целью для выведения более строгого правила для определения константы из третьего шага. Также интересным представляется сравнение различных методов для поиска минимального значения в ТТ. Результаты экспериментов выглядят многообещающе, как можно увидеть в секции 4.

## 2.7. Быстрый алгоритм построения неотрицательного канонического разложения на основе тензорного пезда

Каноническое полиадическое разложение для тензора  $Y \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$  состоит из матриц-факторов  $U^{(i)} = [u_1^{(i)} \dots u_R^{(i)}] \in \mathbb{R}^{n_i \times R}$  и определяется как

$$y(i_1, \dots, i_d) = \sum_{r=1}^R u_r^1(i_1) u_r^2(i_2) \dots u_r^d(i_d),$$

где  $R$  - ранг разложения.

Как было сказано во введении, каноническое полиадическое разложение (CPD) очень популярно и было использовано для многих приложений, например обработка сигналов и задачи машинного обучения, такие как восстановление изображений, извлечение признаков и т.д. Данная модель и ее свойства, включая единственность, хорошо изучены. Однако, несмотря на все преимущества, у нее есть существенный недостаток: вычислительные затраты большинства существующих алгоритмов для CPD растут экспоненциально с порядком тензора. Но существует способ существенно снизить сложность с  $O(dRN^d)$  до  $O(dNR^3)$ , когда  $n_1 = \dots = n_d = N$ .

В [51] авторы предложили перед использованием канонического полиадического разложения изначально сжать исходный тензор в ТТ формат. Для тензоров без возмущений было представлено точное отображение ядер ТТ разложения данного тензора с данными в матрицы-факторы отвечающего CPD, а для тензоров с возмущениями был разработан итеративный алгоритм для оценки матриц-факторов, сложность которого  $O(dNR^3)$ .

Теперь рассмотрим каноническое полиадическое разложение с ограничением на неотрицательность. Неотрицательное каноническое полиадическое разложение (NCPD) для тензора  $Y \in \mathbb{R}_+^{n_1 \times n_2 \times \dots \times n_d}$  составлено из неотрицательных матриц-факторов  $U^{(i)} = [u_1^{(i)} \dots u_R^{(i)}] \in \mathbb{R}_+^{n_i \times R}$ . И по аналогии с классическим CPD известные алгоритмы имеют тот же минус –  $O(dRN^d)$  сложность. В таком случае представляется подходящим использовать тот же подход, чтобы ускорить работу методов.

И в 2020 году группа исследователей попробовала реализовать эту идею и представили результаты как доклад на конференции [54]. Их алгоритм включал 3 шага: уменьшение размерности с помощью неотрицательного тензорного поезда, последующая факторизация ядер полученного неотрицательного ТТ разложения с применением малорангового неотрицательного CPD и построение итогового NCPD. Но данный метод был признан авторами неудачным из-за высокой ошибки аппроксимации даже на тестовых примерах без возмущений.

В данной работе предлагается отличный от [54] подход к построению NCPD представления со сложностью всего лишь  $O(dNR^3)$  и желаемой точностью. Большинство алгоритмов для неотрицательного и классического канонического полиадического разложений используют матрицы развертки исходного тензора, и операции с ними (обычно матричное умножение) дают нам итоговую сложность, оцениваемую как  $O(dRN^d)$ .

$[U]_j$	$j$ ый столбец $[U]$
$u_j^{(n)}$	$j$ ый столбец $U^n$
$U^{\odot -n}$	$U^{(d)} \odot \dots \odot U^{(n+1)} \odot U^{(n-1)} \odot \dots \odot U^{(1)}$
$U^{\otimes -n}$	$U^{(d)} \otimes \dots \otimes U^{(n+1)} \otimes U^{(n-1)} \otimes \dots \otimes U^{(1)}$
$Y_{(n)}$	матрица-развертка $Y$ по моде $n$

Таблица 1. Базовые тензорные операции и обозначения

Для примера рассмотрим FAST-HALS NTF (быстрый иерархический ALS метод для неотрицательной тензорной факторизации) из [55]. В псевдокоде для Алгоритма 8 поэлементное деление, произведение Кронекера, Хатри-Рао (столбцовое произведение Кронекера), Адамара и внешнее произведение обозначаются как  $\oslash$ ,  $\otimes$ ,  $\odot$ ,  $\otimes$ ,  $\circ$  соответственно. Операция  $[\cdot]_+$  меняет отрицательные элементы его аргумента на некоторую очень маленькую константу  $\rho$ .

Для более быстрой версии алгоритма нам требуется произведение тензора на матрицу, которое называют умножением по моде  $k$ . Имея тензор  $A = [A(i_1, i_2, \dots, i_d)]$  и матрицу  $U = [U(\alpha, i_k)]$ , определим результат умножения по моде  $k$  как тензор  $B = [B(i_1, \dots, \alpha, \dots, i_d)]$  ( $\alpha$  на  $k$ -м месте), коорый получается сжатием по  $k$ -му индексу:

$$B(i_1, \dots, \alpha, \dots, i_d) = \sum_{i_k=1}^{n_k} A(i_1, i_2, \dots, i_d) U(\alpha, i_k).$$

Обозначим эту операцию следующим образом:

$$B = A \times_k U.$$

Другие базовые операции и обозначения представлены в таблице 1.

**Дано:** Тензор  $Y(i_1, \dots, i_d)$ ,  $\rho \geq 0$ , неотрицательный CPD ранг  $R$ .

**Найти:** матрицы-факторы  $U^{(i)} = [u_1^i \dots u_R^i] \in \mathbb{R}_+^{n_i \times R}$ .

```

1 Случайная неотрицательная или неотрицательная ALS
   инициализация  $U^{(n)}$ .
2 Нормировать все  $u_j^{(i)}$  для  $i = 1, \dots, d - 1$  до единичной длины.
3  $T_1 = (U^{(1)T}U^{(1)}) \otimes \dots \otimes (U^{(d)T}U^{(d)})$ 
4 while TRUE do
5    $\gamma = \text{diag}(U^{(d)T}U^{(d)})$ 
6   for (  $i = 1:d$  ) {
7     if  $i = d$  then
8        $\gamma = 1$ 
9     end
10     $T_2 = Y_{(i)} \{U^{\odot -i}\}$ 
11     $T_3 = T_1 \oslash (U^{(i)T}U^{(i)})$ 
12    for (  $j = 1:R$  ) {
13       $u_j^{(i)} = [\gamma_j u_j^{(i)} + [T_2]_j - U^{(i)} [T_3]_j]_+$ 
14      if  $i \neq d$  then
15         $u_j^{(i)} = u_j^{(i)} / \|u_j^{(i)}\|_2$ 
16      end
17    }
18     $T_1 = T_3 \otimes U^{(i)T}U^{(i)}$ 
19  }
20  if критерий останова выполнен then
21    break;
22  end
23 end

```

**Алгоритм 8.** Псевдокод для FAST-HALS NTF из [55].

Несмотря на то, что Алгоритм 8 называется FAST HALS, в нем есть операция, которая существенно мешает быстрым вычислениям, а именно:  $Y_{(n)} \{U^{\odot -n}\}$ , так как размер  $Y_{(n)} = N \times N^{d-1}$ .

Чтобы решить эту проблему, используем следующие свойства произве-

дений Кронекера и Хатри-Рао двух векторов:

$$\left[ U^{(1)} \odot U^{(2)} \right]_j = \left[ u_1^{(1)} \otimes u_1^{(2)} \dots u_R^{(1)} \otimes u_R^{(2)} \right]_j = u_j^{(1)} \odot u_j^{(2)}.$$

Предположим, что у нас имеется исходный тензор  $Y$  в формате тензорного поезда

$$y(i_1, \dots, i_d) = \sum_{\alpha_1, \dots, \alpha_{d-1}} G_1(i_1, \alpha_1) G_2(\alpha_1, i_2, \alpha_2) \dots \\ \dots G_{d-1}(\alpha_{d-2}, i_{d-1}, \alpha_{d-1}) G_d(\alpha_{d-1}, i_d).$$

Тогда элементы матрицы,  $T_2 = Y_{(k)} \{U^{\odot-k}\} \in \mathbb{R}^{n_k \times R}$  могут быть записаны как

$$T_2(s, j) = \sum_{i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_d} y(i_1, \dots, s, \dots, i_d) u_j^{(1)}(i_1) \dots \\ \dots u_j^{(k-1)}(i_{k-1}) u_j^{(k+1)}(i_{k+1}) \dots u_j^{(d)}(i_d) = \sum_{i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_d} G_1(i_1) \dots \\ \dots G_d(i_d) u_j^{(1)}(i_1) \dots u_j^{(k-1)}(i_{k-1}) u_j^{(k+1)}(i_{k+1}) \dots u_j^{(d)}(i_d),$$

где  $G_k(i_k)$  – матрица размера  $r_{k-1} \times r_k$ .

$$T_2(s, j) = \sum_{i_1} G_1(i_1) u_j^{(1)}(i_1) \dots G_k(s) \dots \sum_{i_d} G_d(i_d) u_j^{(d)}(i_d),$$

с  $s = \overline{1, n_k}$ ,  $j = \overline{1, R}$ .

Введем матрицы  $\Gamma_j^{(m)} = \sum_{i_m} G_m(i_m) u_j^{(m)}(i_m)$ . Расчет  $\Gamma_j^{(m)}$  означает вычисление произведения тензора на вектор по заданной моде [4]. Фактически для каждого  $j$  нам нужно вычислить  $d - 1$  произведений матрицы на вектор. Тогда вычисление  $\Gamma_j^{(1)} \dots G_k(s) \dots \Gamma_j^{(d)}$  для всех  $s$  и  $j$  потребует  $O(dNr^2R)$  операций. Поэтому итоговая сложность в случае, когда ранги тензорного поезда меньше или равны рангам NCPD, оценивается как  $O(dNR^3)$ . Тогда алгоритм 8 может быть переписан. Реализации необходи-

мых операций с ТТ доступны, например, в [28].

**Дано:** Тензорный поезд  $\tilde{Y}(i_1, \dots, i_d)$ ,  $\rho \geq 0$ , неотрицательный канонический CPD ранг  $R$ .

**Найти:** матрицы-факторы  $U^{(i)} = [u_1^i \dots u_R^i] \in \mathbb{R}_+^{n_i \times R}$ .

```

1 Неотрицательная случайная или неотрицательная ALS
  инициализация  $U^{(n)}$ .
2 Нормировать все  $u_j^{(i)}$  для  $i = 1, \dots, d - 1$  до единичной длины.
3  $T_1 = (U^{(1)T}U^{(1)}) \otimes \dots \otimes (U^{(d)T}U^{(d)})$ 
4 while TRUE do
5    $\gamma = \text{diag}(U^{(d)T}U^{(d)})$ 
6   for (  $j = 1:R$  ) {
7      $W_j = \tilde{Y}$ 
8     for (  $m = 1:d$  ) {
9        $W_j = W_j \times_m (u_j^{(m)})^\top$ 
10    }
11  }
12  for (  $i = 1:d$  ) {
13    if  $i = d$  then
14       $\gamma = 1$ 
15    end
16     $T_3 = T_1 \oslash (U^{(i)T}U^{(i)})$ 
17    for (  $j = 1:R$  ) {
18       $W = W_j$ 
19      Заменить  $i$ -ое ядро  $W$  на  $G_i$ .
20       $[T_2]_j$  - полная форма ТТ тензора  $W$ .
21       $u_j^{(i)} = [\gamma_j u_j^{(i)} + [T_2]_j - U^{(i)} [T_3]_j]_+$ 
22      if  $i \neq d$  then
23         $u_j^{(i)} = u_j^{(i)} / \|u_j^{(i)}\|_2$ 
24      end
25      Заменить  $i$ -ое ядро  $W_j$  на  $G_i$ .
26       $W_j = W_j \times_i (u_j^{(i)})^\top$ 
27    }
28     $T_1 = T_3 \otimes U^{(i)T}U^{(i)}$ 
29  }

```

```

30
31 |   if критерий останова выполнен then
32 |     |   break;
33 |   end
34 end

```

**Алгоритм 9.** Псевдокод FAST-HALS для неотрицательной факторизации тензора с использованием тензорного поезда.

Наивный подход к вычислению  $T_2$  с нуля на каждой итерации приведет к сложности  $O(d^2NR^3)$ . Лучше заранее вычислить общие части вычислений и затем лишь обновлять их. Тогда общая сложность итерации будет оцениваться как  $O(dNR^3)$  с дополнительной памятью  $O(dR^3)$ .

Таким образом, имея тензор в формате тензорного поезда, можно значительно ускорить алгоритм NCPD. Отметим также дополнительное полезное свойство алгоритма 9: даже если ТТ аппроксимация имеет отрицательные элементы, метод построит неотрицательное каноническое разложение. И в данном случае нет нужды в факторизации в неотрицательный тензорный поезд. В качестве методов для построения ТТ приближения  $\tilde{Y}$  можно применить, к примеру, ТТ-SVD или ТТ-CROSS [34].

Пакет Matlab для тензорных разложений "TensorBox" [56] включает различные методы для NCPD: ANLS с методом активных множеств и группировкой столбцов, метод мультипликативных обновлений и т.д. Все они имеют одну и ту же трудоемкую операцию  $Y_{(n)} \{U^{\odot -n}\}$ . Используя подход, описанный выше, можно значительно ускорить все эти алгоритмы декомпозиции в неотрицательное каноническое разложение. Но существуют также методы, для которых требуется прежде построить именно неотрицательный тензорный поезд. Как один из способов сделать это можно использовать технику, описанную в секции 2.6, так как в данном случае неотрицательными должны быть значения именно у аппроксимации, а не у всех факторов. Методы для построения NTT с неотрицательными ядрами могут быть найдены в [5], 2.4, 2.2.

## 2.8. Быстрый алгоритм построения неотрицательного разложения Таккера на основе тензорного поезда

Разложение Таккера для тензоров высокого порядка представляется как декомпозиция исходного тензора  $Y \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$  в тензор-ядро  $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_d}$ , умноженное на набор  $d$  матриц-факторов,  $A^{(i)} = \begin{bmatrix} a_1^{(i)} \\ \dots \\ a_{R_i}^{(i)} \end{bmatrix} \in \mathbb{R}^{n_i \times R_i} (i = 1, 2, \dots, d)$ , в результате:

$$y(i_1, \dots, i_d) = \sum_{j_1, j_2, \dots, j_d} g(j_1, j_2, \dots, j_d) a_{j_1}^1(i_1) a_{j_2}^2(i_2) \dots a_{j_d}^d(i_d).$$

Классическое неотрицательное разложение Таккера (NTD) должно иметь  $\mathcal{G} \in \mathbb{R}_+^{R_1 \times R_2 \times \dots \times R_d}$  и  $A^{(i)} = \begin{bmatrix} a_1^{(i)} \\ a_2^{(i)} \\ \dots \\ a_{R_i}^{(i)} \end{bmatrix} \in \mathbb{R}_+^{n_i \times R_i} (i = 1, 2, \dots, d)$ .

Чтобы построить неотрицательную аппроксимацию Таккера  $\tilde{Y}$ , требуется минимизировать следующую функцию потерь:

$$D_{\text{NTD}} = \frac{1}{2} \|Y - \tilde{Y}\|_F^2.$$

Для оптимизации  $A^n$  рассмотрим матрицу-развертку по моде  $n$  тензора  $Y$  и  $\tilde{Y}$  и перепишем функционал:

$$D_{\text{NTD}} = \frac{1}{2} \left\| Y_{(n)} - A^{(n)} \mathcal{G}_{(n)} (A^{\otimes -n})^\top \right\|_F^2, \quad A^{(n)} \geq 0.$$

Для оптимизации  $\mathcal{G}$ , учитывая векторизацию  $Y$  и  $\tilde{Y}$ , перезапишем функцию  $D_{\text{NTD}}$ :

$$D_{\text{NTD}} = \frac{1}{2} \left\| \text{vec}(Y) - (A^1 \otimes \dots \otimes A^d) \text{vec}(\mathcal{G}) \right\|_F^2, \quad \mathcal{G} \geq 0.$$

Чтобы распространить методы чередующихся наименьших квадратов на неотрицательную тензорную факторизацию в разложение Таккера, нам требуется вычислить частные градиенты  $D_{\text{NTD}}$  по  $A^{(n)}$  и  $\mathcal{G}$ :

$$\frac{\partial D_{\text{NTD}}}{\partial A^{(n)}} = A^{(n)} B^{(n)\top} B^{(n)} - Y_{(n)} B^{(n)},$$

где  $B^{(n)} = (A^{\otimes -n})\mathcal{G}_{(n)}^\top$ , и

$$\frac{\partial D_{\text{NTD}}}{\partial \mathcal{G}} = \tilde{Y} \times_1 (A^{(1)})^\top \cdots \times_d (A^{(d)})^\top - Y \times_1 (A^{(1)})^\top \cdots \times_d (A^{(d)})^\top.$$

Эти частные градиенты и существующие алгоритмы неотрицательной матричной факторизации являются базисом для ряда алгоритмов первого порядка для неотрицательной тензорной факторизации в разложение Таккера [59]. Но все операции, включающие вычисления с исходным тензором  $Y$ , являются очень затратными как по времени, так и по памяти, особенно при работе с большими данными. Аналогично тому, как это было сделано с методами для неотрицательной факторизации в каноническое разложение, можно внедрить разложение в тензорный поезд  $Y$  в алгоритм и таким образом эффективно получить разложение Таккера. И в данном случае это очень легко сделать: все, что требуется, это умножение тензора в ТТ формате на матрицу по заданной моде. Для  $\frac{\partial D_{\text{NTD}}}{\partial \mathcal{G}}$  это следует напрямую из формулы. И для  $Y_{(n)}B^{(n)}$  из  $\frac{\partial D_{\text{NTD}}}{\partial A^{(n)}}$  примем во внимание, что  $Y_{(n)}A^{\otimes -n}$  является в точности матрицей-разверткой по моде  $n$  тензора  $Y \times_1 (A^{(1)})^\top \cdots \times_{n-1} (A^{(n-1)})^\top \times_{n+1} (A^{(n+1)})^\top \cdots \times_d (A^{(d)})^\top$ . Таким образом можно снизить сложность операции до  $O(dNR^3)$ .

В [57] предлагается метод чередующихся ближайших градиентов (alternating proximal gradient method - APG) для построения неотрицательного разложения Таккера. В отличие от метода чередующихся неотрицательных наименьших квадратов (ANLS), в котором находится точное решение каждой подзадачи, APG обновляет каждую матрицу-фактор, решая ослабленную подзадачу с сепарабельной квадратичной целевой функцией. Каждая ослабленная подзадача имеет известное решение, в итоге сложность каждой итерации снижается. Благодаря экстраполяции APG достаточно быстро сходится. Но, как прокомментировали авторы, вычисление частных градиентов занимает очень много времени для больших тензоров. Введение разложения в тензорный поезд, как это обсуждалось ранее, помогает значительно ускорить данный алгоритм. Метод чередующихся ближайших градиентов для построения неотрицательного разложения Таккера на каж-

дой итерации правит значения  $A^n$  и  $\mathcal{G}$ , которые получаются отрицательными, и это означает, что не требуется использовать именно неотрицательное приближение в виде тензорного произведения, достаточно любой факторизации в ТТ, поскольку результат при этом не изменится.

## 2.9. Чередующиеся проекции: общая идея

Общая идея чередующихся проекций заключается в построении итеративной последовательности матриц или тензоров, сходящихся к некоторой точке от пересечения двух множеств: множества неотрицательных матриц (тензоров)  $M_{\geq 0}$  и множества матриц / тензоров низкого ранга  $M_{\leq r}$ . Первое множество  $M_{\geq 0}$  является выпуклым множеством, а  $M_{\leq r}$  является более сложным гладким многообразием (смотрите доказательство для случая матрицы в [60]).

Оптимальная операция проецирования для набора неотрицательных матриц / тензоров  $M_{\geq 0}$  в терминах нормы Фробениуса проста

$$\pi_1(M)(i_1, \dots, i_d) = \begin{cases} M(i_1, \dots, i_d), & M(i_1, \dots, i_d) \geq 0 \\ 0, & M(i_1, \dots, i_d) < 0 \end{cases},$$

Оператор квазиоптимальной проекции (см., например, [33]) на множество тензоров низкого ранга в формате ТТ - это просто операция TTSVD

$$\pi_2(M) = \text{TTSVD}(M, r_1, \dots, r_{d-1}).$$

Таким образом, итерационная процедура может быть организована следующим образом:

1. Принимает текущий тензор  $M_k$  из предыдущих итераций и набор целевых ТТ-рангов  $r_1, \dots, r_{d-1}$  в качестве входных данных;
2. Вычисляется  $M_{k+1}^* = \pi_1(M_k)$ , который удаляет отрицательные элементы и приводит к увеличению рангов по сравнению с предыдущей итерацией;
3. Вычисляется  $M_{k+1} = \pi_2(M_{k+1}^*, r_1, \dots, r_{d-1})$ , что дает приближение тен-

зора низкого ранга, но может вводить отрицательные элементы в построенное приближение.

4. Проверяются критерии остановки, например,

$$\|M_{k+1} - M_k\|_F < \varepsilon$$

и продолжаются итерации до сходимости.

Доказательство, приводящее к линейной сходимости такой процедуры, можно найти в [60] для матричного случая, а первое обобщение для тензоров низкого ранга в канонических полиадических форматах и форматах Таккера можно найти в [61]. В более поздних работах [58, 62] экспериментально было показано, что такая схема может быть ускорена с использованием менее точных, но более эффективных рандомизированных процедур SVD для построения либо матрицы низкого ранга, либо тензорных аппроксимаций (в форматах TT и Tucker). В случае формата Tucker нам следует просто установить  $\pi_2(M) = \text{HOSVD}(M, r_1, \dots, r_d)$  или его более эффективную версию с последовательным усечением (STHOSVD). Конечная сложность каждой итерации таким методом составляет  $O(n^d)$  операций и требует обработки и хранения всех тензорных элементов в отличие от NTT, NTT-MU или APG, основанных на модели TT.

### 3. Применение неотрицательной тензорной факторизации для ускорения вычислений меры влияния узлов многомерной сети

#### 3.1. Теорема Фробениуса-Перрона для мультиоднородных отображений

Будем использовать  $V, V_+$  и  $V_{++}$  для обозначения пространства  $V = \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_d}$ , конуса  $V_+ = \mathbb{R}_+^{n_1} \times \cdots \times \mathbb{R}_+^{n_d}$  и  $V_{++} = \mathbb{R}_{++}^{n_1} \times \cdots \times \mathbb{R}_{++}^{n_d}$ , где  $\mathbb{R}_+$  – все неотрицательные действительные числа, а  $\mathbb{R}_{++}$  – все положительные действительные числа.

Рассмотрим функцию  $\mathcal{F} = (F_1, \dots, F_d) : V \rightarrow V$  со следующими свойствами:

1.  $\mathcal{F}$  непрерывна
2. Существует  $d \times d$  матрица  $A = (a_{ij}) \geq 0$  такая, что

$$F_i(\mathbf{x}_1, \dots, \lambda \mathbf{x}_j, \dots, \mathbf{x}_d) = \lambda^{a_{ij}} F_i(\mathbf{x}_1, \dots, \mathbf{x}_j, \dots, \mathbf{x}_d), \quad \forall i, j = 1, \dots, d$$

3.  $\mathcal{F}(\underline{\mathbf{x}}) \in V_+$  для любого вектора  $\underline{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_d) \in V_+$
4.  $\mathcal{F}(\underline{\mathbf{x}}) \in V_{++}$  для любого вектора  $\underline{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_d) \in V_{++}$
5. Если  $\mathbf{x}_i \geq \mathbf{y}_i$  (поэлементно) для любого  $i = 1, \dots, d$ , то  $F_i(\underline{\mathbf{x}}) \geq F_i(\underline{\mathbf{y}})$  для любого  $i = 1, \dots, d$ , где  $\underline{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_d)$  и  $\underline{\mathbf{y}} = (\mathbf{y}_1, \dots, \mathbf{y}_d)$ .

Скажем, что  $\mathcal{F} \in \text{MH}_d$ , если она удовлетворяет всем вышперечисленным свойствам. Каждой  $\mathcal{F} \in \text{MH}_d$  соответствует матрица  $A$ , определенная как в пункте 2, которую назовем матрицей однородности  $\mathcal{F}$ . Отметим, что свойство пункта 2 можно сократить:

$$\mathcal{F}(\lambda \odot \underline{\mathbf{x}}) = \lambda^A \odot \mathcal{F}(\underline{\mathbf{x}}),$$

где  $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_d) \in \mathbb{R}^d$ ,  $\underline{\mathbf{x}} \in V$ ,  $\boldsymbol{\lambda} \odot \underline{\mathbf{x}} = (\lambda_1 \mathbf{x}_1, \dots, \lambda_d \mathbf{x}_d)$ , а “матричное возведение в степень”  $\boldsymbol{\lambda}^A$  — это вектор  $\mathbb{R}^d$ , определяемый следующим образом

$$\boldsymbol{\lambda}^A = (\lambda_1^{a_{11}} \cdots \lambda_d^{a_{1d}}, \dots, \lambda_1^{a_{d1}} \cdots \lambda_d^{a_{dd}}) = \left( \prod_{j=1}^d \lambda_j^{a_{1j}}, \dots, \prod_{j=1}^d \lambda_j^{a_{dj}} \right).$$

Представляет интерес существование, единственность и максимальность собственных векторов и значений такого рода функций, которые определяются следующим образом:

**Определение 2.** По определению  $\underline{\mathbf{x}} \in V$  и  $\boldsymbol{\lambda} \in \mathbb{R}^d$  являются собственным вектором и собственным значением (соответственно)  $\mathcal{F} \in \text{MH}_d$ , если

$$\mathcal{F}(\underline{\mathbf{x}}) = \boldsymbol{\lambda} \odot \underline{\mathbf{x}}$$

В соответствии с этим определением собственное значение отображения в  $\text{MH}_d$  является вектором, а не числом. Более того, если  $\underline{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_d)$  — собственный вектор, то также  $\boldsymbol{\alpha} \odot \underline{\mathbf{x}} = (\alpha_1 \mathbf{x}_1, \dots, \alpha_d \mathbf{x}_d)$  — собственный вектор с, возможно, другим собственным значением. Итак, в общем случае будем говорить, что собственный вектор определен “с точностью до масштабирования” в том смысле, что  $\underline{\mathbf{x}}$  и  $\boldsymbol{\alpha} \odot \underline{\mathbf{x}}$  будут для нас одним и тем же собственным вектором.

Справедлив следующий результат наподобие теоремы Перрона-Фробениуса.

**Теорема 9.** [68] Пусть  $\mathcal{F} \in \text{MH}_d$  и  $A > 0$  — ее матрица однородности со спектральным радиусом  $\rho(A)$ . Если  $\rho(A) < 1$ , то

1. Существует единственный (с точностью до масштабирования) собственный вектор  $\underline{\mathbf{u}} \in V_{++}$  оператора  $\mathcal{F}$ .
2. Соответствующее собственное значение  $\boldsymbol{\lambda} \in \mathbb{R}^d$  является поэлементно положительным и максимальным в том смысле, что если  $\mathcal{F}(\underline{\mathbf{x}}) = \boldsymbol{\mu} \odot \underline{\mathbf{x}}$  для некоторых  $\boldsymbol{\mu}$  и  $\underline{\mathbf{x}}$ , то

$$\lambda_1^{b_1} \cdots \lambda_d^{b_d} \geq |\mu_1^{b_1} \cdots \mu_d^{b_d}|$$

где  $\mathbf{b} = (b_1, \dots, b_d) > 0$  таков, что  $A^T \mathbf{b} = \rho(A) \mathbf{b}$ , т.е.  $\mathbf{b}$  — любой собственный вектор Перрона матрицы однородности  $A^T$  из  $\mathcal{F}$ .

Заметим, что благодаря этой теореме мы можем называть  $\underline{\mathbf{u}} \in V_{++}$  собственным вектором Перрона  $\mathcal{F}$ . В качестве примечания отметим, что почти тот же результат справедлив и для случая  $\rho(A) = 1$ , но с дополнительными условиями на "неприводимость" отображения  $\mathcal{F}$ .

Поскольку единственность собственного вектора Перрона обеспечивается только с точностью до масштабирования, в дальнейшем зафиксируем некоторые нормы — для простоты  $\ell^p$ -нормы — для любого из пространств  $\mathbb{R}^{n_i}$ . Итак, пусть  $\mathbf{p} = (p_1, \dots, p_d) \geq 1$  ( $\mathbf{p}$  для нас — вектор, идентифицирующий выбранные нормы) и рассмотрим сферу  $V$ , определенную формулой

$$S_{\mathbf{p}} = \{ \underline{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_d) \in V : \|\mathbf{x}_1\|_{p_1} = \dots = \|\mathbf{x}_d\|_{p_d} = 1 \}.$$

Очевидным следствием приведенной выше теоремы является то, что при любом выборе  $\mathbf{p} \geq 1$  отображение  $\mathcal{F} \in \text{MH}_d$  с  $\rho(A) < 1$  имеет единственный положительный собственный вектор  $\underline{\mathbf{u}} \in S_{\mathbf{p}}$ . Выполняется также следующее

**Теорема 10.** [68] Пусть  $\mathcal{F} = (F_1, \dots, F_d)$  такой же, как в теореме 9. Тогда для любого выбора  $\underline{\mathbf{x}}^{(0)} \in V_{++}$  последовательность

$$\underline{\mathbf{x}}^{(k+1)} = \left( \frac{F_1(\underline{\mathbf{x}}^{(k)})}{\|F_1(\underline{\mathbf{x}}^{(k)})\|_{p_1}}, \dots, \frac{F_d(\underline{\mathbf{x}}^{(k)})}{\|F_d(\underline{\mathbf{x}}^{(k)})\|_{p_d}} \right)$$

принадлежит  $S_{\mathbf{p}}$  и сходится к единственному положительному собственному вектору  $\mathcal{F}$ :  $\underline{\mathbf{u}} \in S_{\mathbf{p}}$ . Более того,  $\mu_{\mathbf{b}}(\underline{\mathbf{u}}, \underline{\mathbf{x}}^{(k)}) \leq \frac{\rho(A)^k}{1-\rho(A)} \mu_{\mathbf{b}}(\underline{\mathbf{x}}^{(1)}, \underline{\mathbf{x}}^{(0)})$   $\forall k \in \mathbb{N}$ , где

$$\mu_{\mathbf{b}}(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^d b_i \ln \left[ \left( \max_{j_i \in [n_i]} \frac{x_{i,j_i}}{y_{i,j_i}} \right) \left( \max_{l_i \in [n_i]} \frac{y_{i,l_i}}{x_{i,l_i}} \right) \right] \quad \forall \mathbf{x}, \mathbf{y} \in V_{++}$$

### 3.2. Пример $\mathcal{F} \in \text{MH}_d$ , определенного через неотрицательный тензор

Одной из главных задач специалистов по обработке данных является выявление релевантных компонентов в сети или графе. Сети часто пред-

ставляют в виде матриц; следовательно, инструменты матричного анализа оказываются полезными при решении различных проблем, таких как выявление сообществ и анτισообществ, разделение сети на кластеры или идентификация центральных узлов, ребер. За последние годы несколько авторов, сосредоточив внимание на этой последней проблеме и, в частности, на проблеме определения наиболее центральных узлов, работали над определением различных показателей центральности. Мера центральности - это вещественнозначная функция набора узлов, которая является инвариантной при перемаркировке узлов и, таким образом, может быть использована для их ранжирования в соответствии с их важностью. Меры, основанные на собственных векторах или сингулярных векторах подходящих матриц, в дальнейшем называемые (линейными) центральностями собственных векторов, являются одними из наиболее популярных мер центральности. Соответствующие примеры включают, например, индекс Боначича и рейтинг PageRank; последний широко известен во всем мире благодаря использованию поисковой системой Google.

В [6] авторы используют собственный вектор Перрона простого мультиоднородного отображения для ранжирования узлов важности многоуровневой сети, а именно мультиплекса - набора графов. Все слои мультиплекса содержат один и тот же набор узлов, при этом нет никаких связей между узлами, которые принадлежат двум разным слоям. Мультиплекс можно описать как набор графов:

$$\mathcal{G} = \left\{ G^{(\ell)} = \left( V_n, E^{(\ell)} \right) \right\}_{\ell \in V_L}$$

где  $V_n = \{1, 2, \dots, n\}$  - множество узлов,  $V_L = \{1, \dots, L\}$  - множество слоев,  $E^{(\ell)} \subset V_n \times V_n$  - множество ребер на слое  $\ell$ . Для каждого  $\ell \in V_L$  граф  $G^{(\ell)}$  описывается неотрицательной матрицей смежности  $A^{(\ell)} = \left( A_{ij}^{(\ell)} \right) \in \mathbb{R}^{n \times n}$ . Поэтому мультиплекс можно описать тензором третьего порядка  $\mathcal{A} = (\mathcal{A}_{ij\ell})$ , называемым тензором смежности:

$$\mathcal{A}_{ij\ell} = A_{ij}^{(\ell)} = \begin{cases} w_{\ell}(i, j) & \text{если } (i, j) \in E^{(\ell)} \\ 0 & \text{иначе} \end{cases}$$

где  $w_\ell(i, j)$  - положительное число, отражающее силу связи между узлами  $i$  и  $j$  в слое  $\ell \in V_L$ . Мультиплексы являются частным случаем многомерных графов, однако на практике мы сталкиваемся и с более сложными объектами.

Пусть  $\mathbf{T} = (t_{i_1, \dots, i_d}) \geq 0$  — тензор с неотрицательными элементами. Для  $\underline{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_d) \in V$  и  $j = 1, \dots, d$  определяется  $T_j : V \rightarrow \mathbb{R}^{n_j}$  как

$$T_j(\underline{\mathbf{x}})_{i_j} = \sum_{i_1, \dots, i_{j-1}, i_{j+1}, \dots, i_d} t_{i_1, \dots, i_d} (\mathbf{x}_1)_{i_1} \cdots (\mathbf{x}_{j-1})_{i_{j-1}} (\mathbf{x}_{j+1})_{i_{j+1}} \cdots (\mathbf{x}_d)_{i_d} \quad (18)$$

для  $i_j = 1, \dots, n_j$ . Конечно, приведенная выше сумма справедлива для  $i_1 = 1, \dots, n_1, i_2 = 1, \dots, n_2$  и так далее.

Теперь, учитывая  $\alpha_1, \dots, \alpha_d > 0$ , определяется  $\mathcal{F} : V \rightarrow V$  как

$$\mathcal{F}(\underline{\mathbf{x}}) = \left( T_1(\underline{\mathbf{x}})^{1/\alpha_1}, \dots, T_d(\underline{\mathbf{x}})^{1/\alpha_d} \right) \quad (19)$$

где степень берется поэлементно, т.е.  $\underline{\mathbf{x}}^\beta = (x_1^\beta, x_2^\beta, \dots)$ . Тогда  $\mathcal{F} \in \text{MH}_d$  со следующей матрицей однородности

$$A = \begin{pmatrix} 0 & 1/\alpha_1 & 1/\alpha_1 & 1/\alpha_1 & 1/\alpha_1 & \cdots \\ 1/\alpha_2 & 0 & 1/\alpha_2 & 1/\alpha_2 & 1/\alpha_2 & \cdots \\ 1/\alpha_3 & 1/\alpha_3 & 0 & 1/\alpha_3 & 1/\alpha_3 & \cdots \\ 1/\alpha_4 & 1/\alpha_4 & 1/\alpha_4 & 0 & 1/\alpha_4 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \quad (20)$$

Предположим, что параметры  $\alpha_i$  выбраны так, что  $\rho(A) < 1$ . Тогда для любого  $\underline{\mathbf{x}}^{(0)} \in V_{++}$  можно построить сходящуюся последовательность

$$\left( \frac{T_1(\underline{\mathbf{x}}^{(k)})^{1/\alpha_1}}{\|\cdots\|_{p_1}}, \dots, \frac{T_d(\underline{\mathbf{x}}^{(k)})^{1/\alpha_d}}{\|\cdots\|_{p_d}} \right) \xrightarrow{k \rightarrow \infty} \underline{\mathbf{u}} \in V_{++} \quad (21)$$

где  $\underline{\mathbf{u}}$  — единственный положительный вектор такой, что  $\mathcal{F}(\underline{\mathbf{u}}) = \lambda \odot \underline{\mathbf{u}}$ .

Когда  $t_{i_1, \dots, i_d}$  описывает свойства сети (например, это может быть тензор смежности), тогда вектор  $\underline{\mathbf{u}}$  можно использовать для присвоения определенной важности узлам и слоям (и временным интервалам) в многоуров-

невой сети. Это приложение для сетей особенно интересно для нас, поскольку в этом случае нам не нужна высокая точность. В том смысле, что нам необязательно знать значение всех элементов  $\mathbf{u} = (\mathbf{u}_1, \dots, \mathbf{u}_d)$  с высокой точностью. Что важно, так это относительный порядок самых больших элементов каждого  $\mathbf{u}_i$ , которые обычно являются элементами, которые сходятся быстрее в последовательности (21).

Поясним это на примере. Предположим, у нас есть последовательность  $\mathcal{S}_t$  многослойных графов с  $t = 1, \dots, \bar{t}$ , где для каждого  $t$   $\mathcal{S}_t$  - это множество графов  $\mathcal{S}_t = (S_1, \dots, S_{\bar{\ell}})$  на одном и том же наборе узлов  $\{1, \dots, n\}$ . Параметр  $t$  является меткой времени, т.е. предполагается, что многоуровневый граф/сеть меняется во времени. Весь этот объект может быть однозначно описан следующим тензором смежности

$$\mathbf{A} = (a_{ijlt}) = \begin{cases} \omega > 0 & , \text{ если существует ребро от узла } i \text{ до узла } j \text{ в слое } \ell \\ & \text{ в момент времени } t \\ 0 & , \text{ иначе} \end{cases}$$

Определим разные виды векторов “важности” (влиятельности) для последовательности  $\mathcal{S}_1, \dots, \mathcal{S}_{\bar{t}}$ :  $\mathbf{x} = (x_1, \dots, x_n)$  — вектор влиятельности узлов, которые можно назвать *трансляторами*,  $\mathbf{y} = (y_1, \dots, y_n)$  — вектор влиятельности узлов известных как *получатели*,  $\mathbf{w} = (w_1, \dots, w_{\bar{\ell}})$  — вектор влиятельности слоев, а  $\mathbf{z} = (z_1, \dots, z_{\bar{t}})$  — вектор влиятельности меток времени. Тогда возможная модель влиятельности в точности определяется выражением  $\mathcal{F}(\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{z}) = (\lambda, \mu, \nu, \eta) \odot (\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{z})$ , где  $\mathcal{F}$  определяется как в (19), с  $\mathbf{A}$  вместо  $\mathbf{T}$ .

Нам интересно узнать, какие  $k$  (скажем, 10) наиболее важных узлов, слоев и меток времени. Итак, фиксируются  $\alpha_1, \dots, \alpha_d$  так, чтобы  $\rho(A) < 1$ ; вычисляется собственный вектор Перрона  $(\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{z})$   $\mathcal{F}$ , а затем отмечаются 10 крупнейших элементов (и их порядок)  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{w}$  и  $\mathbf{z}$ . Однако на практике мы хотим выбрать  $\rho(A) \approx 1$ , и это может замедлить сходимость. Более того, если у нас много слоев и большое временное окно, умножение тензора смежности, как в (18), является трудоемкой операцией.

### 3.3. Применение неотрицательной тензорной факторизации

Если аппроксимировать  $\mathbf{T} = t_{i_1, \dots, i_d}$  с помощью НТТ, т.е.  $\mathbf{T} \approx \tilde{\mathbf{T}} = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_{d-1}=1}^{r_{d-1}} G_1(i_1, \alpha_1) G_2(\alpha_1, i_2, \alpha_2) \cdots G_d(\alpha_{d-1}, i_d)$ , можно определить отображения  $\mathcal{S}_j : V \rightarrow R^{n_j}$  как в (18), но заменяя  $\mathbf{T}$  на  $\tilde{\mathbf{T}}$ . Конечно, отображение  $\tilde{\mathcal{F}}(\underline{\mathbf{x}}) = \left( \tilde{T}_1(\underline{\mathbf{x}})^{1/\alpha_1}, \dots, \tilde{T}_d(\underline{\mathbf{x}})^{1/\alpha_d} \right)$  также принадлежит  $\text{MH}_d$  с той же матрицей однородности  $A$ , что и  $\mathcal{F}$ . Так

$$\left( \frac{\tilde{T}_1(\underline{\mathbf{x}}^{(k)})^{1/\alpha_1}}{\|\cdots\|_{p_1}}, \dots, \frac{\tilde{T}_d(\underline{\mathbf{x}}^{(k)})^{1/\alpha_d}}{\|\cdots\|_{p_d}} \right) \xrightarrow{k \rightarrow \infty} \tilde{\mathbf{u}} \in V_{++}$$

### 3.4. Ошибка приближенного решения

Для простоты напишем все для случая  $d = 3$ , т.е. тензоров третьего порядка. Версия порядка  $d$  с  $d > 3$  выводится по аналогии. Для вектора  $\mathbf{b} = (b_1, \dots, b_d)$  рассмотрим следующую гильбертову метрику "высшего порядка":

$$\mu_{\mathbf{b}}(\underline{\mathbf{x}}, \underline{\mathbf{y}}) = \sum_{k=1}^d b_k \mu(\mathbf{x}_k, \mathbf{y}_k), \quad \mu(\mathbf{x}, \mathbf{y}) = \log \left( \frac{\max_i x_i / y_i}{\min_i x_i / y_i} \right)$$

Пусть  $\Omega \subseteq \mathbb{R}^n$  — множество всех стохастических векторов

$$\Omega := \left\{ \mathbf{s} \in \mathbb{R}^n : \sum_{i=1}^n s_i = 1, s_i \geq 0, i = 1, \dots, n \right\},$$

и пусть  $\Omega_+ \subseteq \Omega$  — множество поэлементно положительных стохастических векторов.

Одним из наиболее известных алгоритмов определения важности узлов в ориентированном графе, представляющем веб-страницы, является модель PageRank, разработанная Google [64]. Учитывая матрицу перехода  $P = (p_{ij})$ ,  $\sum_i p_{ij} = 1$  случайного блуждания по графу с  $n$  узлами  $V = \{1, \dots, n\}$  и учитывая  $\mathbf{v} = (v_1, \dots, v_n) : \sum_{i=1}^n v_i = 1, v_i > 0, i = 1, \dots, n$ , необходимо вычислить положительный собственный вектор матрицы пере-

хода PageRank

$$P_{\text{PR}} = \alpha P + (1 - \alpha)\mathbf{v}\mathbf{e}^T,$$

где  $\mathbf{e}$  — вектор из всех единиц. Такая матрица перехода PageRank  $P_{\text{PR}}$  моделирует новое случайное блуждание, при котором с вероятностью  $\alpha$  делается шаг по исходной цепи Маркова, а с вероятностью  $1 - \alpha$  случайным образом происходит перемещение на узел  $i$  согласно фиксированной вероятности телепортации  $v_i > 0$ . Обратим внимание, что, так как  $\mathbf{v} \in \Omega_+$ , для любого  $0 \leq \alpha < 1$  матрица PageRank  $P_{\text{PR}}$  неприводима и, следовательно, по теореме Фробениуса-Перрона существует единственный положительный собственный вектор  $\mathbf{s} \in \Omega_+$  такой, что  $P_{\text{PR}}\mathbf{s} = \mathbf{s}$ .

Для мультилинейного PageRank [65, 66] задача формулируется следующим образом. Учитывая стохастический тензор  $\mathcal{P}$ , описывающий вероятности перехода случайного блуждания по графу более высокого порядка с  $n$  узлами, рассматривается следующий тензор перехода PageRank

$$\mathcal{P}_{\text{PR}} := \alpha\mathcal{P} + (1 - \alpha)\mathcal{V},$$

где, учитывая вектор телепортации  $\mathbf{v} \in \Omega_+$  и вектор, состоящий из одних единиц  $\mathbf{e}$ ,  $\mathcal{V}$  — это положительный тензор первого ранга  $\mathcal{V} = \mathbf{v} \otimes \mathbf{e} \otimes \dots \otimes \mathbf{e}$  с элементами  $\mathcal{V}_{i_1, \dots, i_m} = v_{i_1}$ . Теперь для любого  $\mathbf{s} \in \Omega$  имеем  $\mathcal{V}\mathbf{s}^{m-1} = (\mathbf{v} \otimes \mathbf{e} \otimes \dots \otimes \mathbf{e})\mathbf{s}^{m-1} = (\mathbf{e}^T \mathbf{s})^{m-1} \mathbf{v} = \mathbf{v}$  и, таким образом, справедливо

$$\mathcal{P}_{\text{PR}}\mathbf{s}^{m-1} = \alpha\mathcal{P}\mathbf{s}^{m-1} + (1 - \alpha)\mathcal{V}\mathbf{s}^{m-1} = \alpha\mathcal{P}\mathbf{s}^{m-1} + (1 - \alpha)\mathbf{v},$$

показывая, что  $\mathbf{s} \in \Omega$  является фиксированной точкой  $f(\mathbf{s}) = \alpha\mathcal{P}\mathbf{s}^{m-1} + (1 - \alpha)\mathbf{v}$  тогда и только тогда, когда  $\mathcal{P}_{\text{PR}}\mathbf{s}^{m-1} = \mathbf{s}$ .

Следуя мультилинейной модели PageRank, также можно рассмотреть тензор перехода  $T$  вместо исходного тензора  $P$ .

Теперь предположим, что нам даны два неотрицательных тензора  $T$  и  $\tilde{T}$  такие, что  $(1 - \rho)T_{ijk} \leq \tilde{T}_{ijk} \leq (1 + \rho)T_{ijk}$ . Хотя это неравенство означает, что  $T_{ijk} = 0$  тогда и только тогда, когда  $\tilde{T}_{ijk} = 0$ , поскольку рассматривается тензор перехода, все его значения гарантированно будут положительными.

Тогда можно определить гильбертово расстояние между  $T$  и  $\tilde{T}$  как

$$\mu(T, \tilde{T}) = \log \left( \max_{\substack{i,j,k \\ a,b,c}} \frac{T_{ijk} \tilde{T}_{abc}}{\tilde{T}_{ijk} T_{abc}} \right)$$

Если  $(1 - \rho)T_{ijk} \leq \tilde{T}_{ijk} \leq (1 + \rho)T_{ijk}$  для ненулевых элементов  $T$  и  $\tilde{T}$ , то

$$\mu(T, \tilde{T}) \leq \log \left( \frac{1 + \rho}{1 - \rho} \right).$$

Для тензора  $T$  пусть  $(T_1 \mathbf{y} \mathbf{z})_i = \sum_{jk} T_{ijk} y_j z_k$ ,  $(T_2 \mathbf{x} \mathbf{z})_j = \sum_{ik} T_{ijk} x_i z_k$  и так далее. Пусть также  $\mathcal{F}_T$  и  $\underline{\mathbf{u}} = (\mathbf{x}, \mathbf{y}, \mathbf{z})$  — мультиоднородное отображение и соответствующий собственный вектор, определяемые соответственно как

$$\mathcal{F}_T(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \begin{bmatrix} (T_1 \mathbf{y} \mathbf{z})^{1/\alpha_1} / \|(T_1 \mathbf{y} \mathbf{z})^{1/\alpha_1}\| \\ (T_2 \mathbf{x} \mathbf{z})^{1/\alpha_2} / \|(T_2 \mathbf{x} \mathbf{z})^{1/\alpha_2}\| \\ (T_3 \mathbf{x} \mathbf{y})^{1/\alpha_3} / \|(T_3 \mathbf{x} \mathbf{y})^{1/\alpha_3}\| \end{bmatrix} = \begin{bmatrix} \lambda_1 \mathbf{x} \\ \lambda_2 \mathbf{y} \\ \lambda_3 \mathbf{z} \end{bmatrix} = \boldsymbol{\lambda} \odot (\mathbf{x}, \mathbf{y}, \mathbf{z})$$

Напомним, что

$$A = \begin{bmatrix} 0 & 1/\alpha_1 & 1/\alpha_1 \\ 1/\alpha_2 & 0 & 1/\alpha_2 \\ 1/\alpha_3 & 1/\alpha_3 & 0 \end{bmatrix}$$

— матрица однородности  $\mathcal{F}_T$ . Справедлива следующая лемма

**Лемма 5.** [6] Пусть  $\mathbf{b} > 0$  — положительный собственный вектор Перрона  $A^T$  (т.е.  $A^T \mathbf{b} = \rho(A) \mathbf{b}$ ) и  $\|\mathbf{b}\|_1 = 1$ . Тогда

$$\mu_{\mathbf{b}} \left( \mathcal{F}_T(\mathbf{x}, \mathbf{y}, \mathbf{z}), \mathcal{F}_T(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}}) \right) \leq \rho(A) \mu_{\mathbf{b}} \left( (\mathbf{x}, \mathbf{y}, \mathbf{z}), (\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}}) \right)$$

для каждого положительного  $\underline{\mathbf{u}} = (\mathbf{x}, \mathbf{y}, \mathbf{z}), \tilde{\underline{\mathbf{u}}} = (\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ .

**Лемма 6.** Для любых неотрицательных нормированных векторов  $\mathbf{x}, \mathbf{y}$  справедливо, что  $\mu(\mathbf{x}, \mathbf{y}) \geq \|\mathbf{x} - \mathbf{y}\|_{\infty}$ .

**Доказательство.** Пусть  $\log(\mathbf{x})$  — вектор с элементами  $\log(\mathbf{x})_i = \log(x_i)$ .

Из неравенства  $|a - b| \leq \max\{a, b\} |\log a - \log b|$  получаем

$$\begin{aligned} \|\mathbf{x} - \mathbf{y}\|_\infty &\leq \left( \max_i \max\{x_i, y_i\} \right)^{-1} \|\mathbf{x} - \mathbf{y}\|_\infty \leq \|\log(\mathbf{x}) - \log(\mathbf{y})\|_\infty \\ &= \log(\max_i e^{|\log(x_i) - \log(y_i)|}) = \log(\max\{\max_i \frac{x_i}{y_i}, \max_i \frac{y_i}{x_i}\}) \leq \mu(\mathbf{x}, \mathbf{y}). \end{aligned}$$

**Теорема 11.** Пусть  $A$  — общая матрица однородности  $\mathcal{F}_T$  и  $\mathcal{F}_{\tilde{T}}$ . Если  $\rho(A) < 1$ , тогда существуют единственные  $\underline{\mathbf{u}} = (\mathbf{x}, \mathbf{y}, \mathbf{z}) > 0$  и  $\underline{\tilde{\mathbf{u}}} = (\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}}) > 0$  такие, что

$$\mathcal{F}_T(\underline{\mathbf{u}}) = \boldsymbol{\lambda} \odot \underline{\mathbf{u}} \quad \text{и} \quad \mathcal{F}_{\tilde{T}}(\underline{\tilde{\mathbf{u}}}) = \tilde{\boldsymbol{\lambda}} \odot \underline{\tilde{\mathbf{u}}},$$

а также имеет место следующая оценка:

$$\|\underline{\mathbf{u}} - \underline{\tilde{\mathbf{u}}}\|_{b,\infty} \leq \left( \frac{1}{d-1} \right) \left( \frac{\rho(A)}{1-\rho(A)} \right) \mu(T, \tilde{T}),$$

$$\|\underline{\mathbf{x}}\|_{b,\infty} = \sum_{i=1}^d b_i \|x_i\|_\infty.$$

**Доказательство.** Поскольку метрика Гильберта  $\mu_b$  масштабно инвариантна и выполнено неравенство треугольника, то

$$\begin{aligned} \mu_b(\underline{\mathbf{u}}, \underline{\tilde{\mathbf{u}}}) &= \mu_b(\boldsymbol{\lambda} \odot \underline{\mathbf{u}}, \tilde{\boldsymbol{\lambda}} \odot \underline{\tilde{\mathbf{u}}}) = \mu_b(\mathcal{F}_T(\underline{\mathbf{u}}), \mathcal{F}_{\tilde{T}}(\underline{\tilde{\mathbf{u}}})) \\ &\leq \mu_b(\mathcal{F}_{\tilde{T}}(\underline{\mathbf{u}}), \mathcal{F}_{\tilde{T}}(\underline{\tilde{\mathbf{u}}})) + \mu_b(\mathcal{F}_{\tilde{T}}(\underline{\mathbf{u}}), \mathcal{F}_T(\underline{\mathbf{u}})) \\ &\leq \rho(A) \mu_b(\underline{\mathbf{u}}, \underline{\tilde{\mathbf{u}}}) + \mu_b(\mathcal{F}_{\tilde{T}}(\underline{\mathbf{u}}), \mathcal{F}_T(\underline{\mathbf{u}})), \end{aligned}$$

где последнее неравенство следует из леммы 5. Далее сосредоточимся на втором слагаемом в неравенстве. Вновь, так как метрика Гильберта масштабно инвариантна, получаем

$$\mu_b(\mathcal{F}_{\tilde{T}}(\underline{\mathbf{u}}), \mathcal{F}_T(\underline{\mathbf{u}})) = \sum_{s=1}^3 b_s \mu((T_s \mathbf{y} \mathbf{z})^{1/\alpha_s}, (\tilde{T}_s \mathbf{y} \mathbf{z})^{1/\alpha_s}) \quad (22)$$

и, используя неравенство  $(a + b)/(c + d) \leq \max\{a/c, b/d\}$  (справедливое

для неотрицательных чисел), получаем

$$\begin{aligned} \mu((T_s \mathbf{y}z)^{1/\alpha_s}, (\tilde{T}_s \mathbf{y}z)^{1/\alpha_s}) &= \log \left( \max_i \left[ \frac{(T_s \mathbf{y}z)_i}{(\tilde{T}_s \mathbf{y}z)_i} \right]^{1/\alpha_s} \max_i \left[ \frac{(\tilde{T}_s \mathbf{y}z)_i}{(T_s \mathbf{y}z)_i} \right]^{1/\alpha_s} \right) \\ &\leq \frac{1}{\alpha_s} \log \left( \max_{ijk} \frac{T_{ijk}}{\tilde{T}_{ijk}} \max_{ijk} \frac{\tilde{T}_{ijk}}{T_{ijk}} \right) = \frac{1}{\alpha_s} \mu(T, \tilde{T}) \end{aligned}$$

Таким образом, из (22)  $\mu_{\mathbf{b}}(\mathcal{F}_{\tilde{T}}(\mathbf{u}), \mathcal{F}_T(\mathbf{u})) \leq \mu(T, \tilde{T}) \sum_{i=1}^3 b_i/\alpha_i$ . Поскольку  $\mathbf{b}$  — собственный вектор Перрона  $A^T$ , имеем  $A^T \mathbf{b} = \rho(A) \mathbf{b}$ . Это тождество по записи читается как  $\sum_{i \neq k} b_i/\alpha_i = \rho(A) b_k$ , таким образом, добавляя  $b_k/\alpha_k$  с обеих сторон,  $\sum_{i=1}^d b_i/\alpha_i = (\rho(A) + \alpha_k^{-1}) b_k$ . Суммируя последнее выражение по  $k$  и используя тождество  $\sum_{k=1}^d b_k = 1$ , получаем

$$d \sum_{i=1}^d \frac{b_i}{\alpha_i} = \sum_{k=1}^d \sum_{i=1}^d \frac{b_i}{\alpha_i} = \sum_{k=1}^d (\rho(A) + \frac{1}{\alpha_k}) b_k = \rho(A) + \sum_{k=1}^d \frac{b_k}{\alpha_k}.$$

Это доказывает, что  $\sum_{i=1}^d b_i/\alpha_i = \rho(A)(d-1)^{-1}$ . Завершая доказательство, используем лемму 6.

**Замечание 12.** Пусть  $\mathbf{1} = (1, \dots, 1)$  — вектор всех единиц. Если выбрать  $\alpha_i = \alpha$  для всех  $i = 1, \dots, d$ , то  $A^T \mathbf{1} = A \mathbf{1} = (d-1)/\alpha \mathbf{1}$ . Это показывает, что  $\rho(A) = (d-1)/\alpha$  и  $\mathbf{b} = \frac{1}{d} \mathbf{1}$ . Поэтому  $\sum_i b_i/\alpha_i = 1/\alpha$  и  $1/(1 - \rho(A)) = \alpha/(\alpha - d + 1)$ . Таким образом, неравенство записывается как

$$\|\underline{\mathbf{u}} - \tilde{\underline{\mathbf{u}}}\|_{\mathbf{b}, \infty} \leq \frac{\mu(T, \tilde{T})}{\alpha - d + 1}$$

В частности, при  $\alpha_i = d$  получаем  $\|\underline{\mathbf{u}} - \tilde{\underline{\mathbf{u}}}\|_{\mathbf{b}, \infty} \leq \mu(T, \tilde{T})$ .

**Замечание 13.** Заметим, что если в качестве норм, определяющих мультиоднородное отображение  $\mathcal{F}_T$ , выбраны нормы бесконечности, то последовательности, порождаемые степенным методом для  $\mathcal{F}_T$  и  $\mathcal{F}_{\tilde{T}}$ , нормализуются так, что  $\|\underline{\mathbf{u}}_i\|_{\infty} = \|\tilde{\underline{\mathbf{u}}}_i\|_{\infty} = 1$ . При таком выборе мы получаем  $\|\underline{\mathbf{u}}\|_{\mathbf{b}, \infty} = \|\tilde{\underline{\mathbf{u}}}\|_{\mathbf{b}, \infty} = 1$  и, таким образом, получаем относительную

*оценку ошибки*

$$\frac{\|\underline{\mathbf{u}} - \tilde{\underline{\mathbf{u}}}\|_{b,\infty}}{\|\underline{\mathbf{u}}\|_{b,\infty}} \leq \left(\frac{1}{d-1}\right) \left(\frac{\rho(A)}{1-\rho(A)}\right) \mu(T, \tilde{T})$$

## 4. Результаты численных экспериментов

### 4.1. Комплекс программ

В процессе подготовки и проведения численных экспериментов был подготовлен комплекс программ на языке Matlab, позволяющий быстро и эффективно строить неотрицательные матричные и тензорные разложения и проводить широкие наборы численных экспериментов. Для решения каждой из рассматриваемых задач был написан отдельный программный модуль, связанный с программными комплексами для построения тензорных разложений [28], [37], [56]. Программный комплекс размещён в репозитории с использованием системы контроля версий Git по адресу [https://github.com/eshcherbakova/NMF\\_NTT/](https://github.com/eshcherbakova/NMF_NTT/).

Программный комплекс содержит две директории - “nmf” и “ntf”, включающие программные реализации методов для неотрицательной матричной и тензорной факторизации соответственно. В Таблице 2 приведено краткое описание модулей комплекса программ.

nmf/FastSepNMF.m	Реализация Алгоритма 1
nmf/IREVA.m	Реализация Алгоритма IREVA [20]
nmf/cmpMethodTimeRank2.m	Реализация Алгоритма 3
nmf/lowrankSepNMF.m	Реализация редуцированного Алгоритма 2
nmf/lraNMF_cross.m	Реализация Алгоритма HALS на основе крестовой аппроксимации 2.2
ntt/nttf.m	Реализация Алгоритма 4
ntt/nttmu.m	Реализация Алгоритма 5
ntt/nttmudmrg.m	Реализация Алгоритма 6
ntt/ncp_from_TT.m	Реализация Алгоритма 9
ntt/ntd_fromTT.m	Реализация Алгоритма из секции 2.8
ntt/tt_powerm.m	Реализация Алгоритма 7
ntt/nntlra	Реализация алгоритмов чередующихся проекций 2.9, [58]

Таблица 2. Краткое описание модулей комплекса программ

## 4.2. Результаты численных экспериментов для неотрицательной матричной факторизации

### 4.2.1. Проверка сложности редуцированного алгоритма для сепарабельных матриц

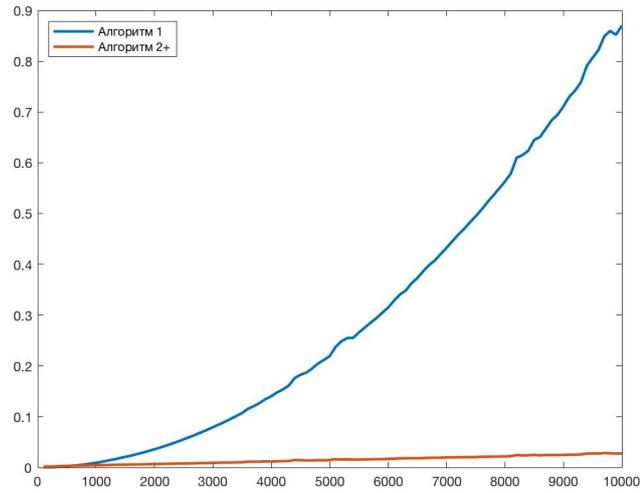
Прежде всего проведем сравнение алгоритма Гиллиса-Вавасиса (1) и нового предложенного метода, когда сначала строится крестовое разложение матрицы, которое затем подается на вход редуцированному алгоритму 2. Такой двухэтапный метод будем называть алгоритмом 2+. Весь код для проведения численных экспериментов был написан в Matlab. При построении крестового разложения использовались функции из библиотеки "ГТ-Toolbox" [28].

Для численных экспериментов сепарабельные матрицы  $M \in R_+^{n \times n}$  ранга  $r$  генерировались следующим образом. Сначала матрицы  $W \in R^{n \times r}$  и  $H' \in R^{r \times (n-r)}$  заполняются модулями нормально распределенных чисел,  $H = [I_r \ H']$ , при этом столбцы  $H$  нормируются так, чтобы сумма элементов в каждом была равна 1 и случайным образом перемешиваются. Итоговая матрица  $M$  равна произведению  $W$  и  $H$ .

Рассматривались  $n$  от 100 до 10000 с шагом 100, для каждого  $n$  строились 100 матриц  $M$ , которые подавались на вход алгоритмам 1 и 2+. На рисунке 2 приведены графики зависимости времени работы от  $n$  при фиксированных  $r$ . Программная реализация алгоритма 1 была взята из [29].

Во всех случаях оба алгоритма верно определили столбцы  $W$ . График времени работы нового метода демонстрирует линейную зависимость от размера, алгоритм 1 - квадратичную.

(а)



(б)

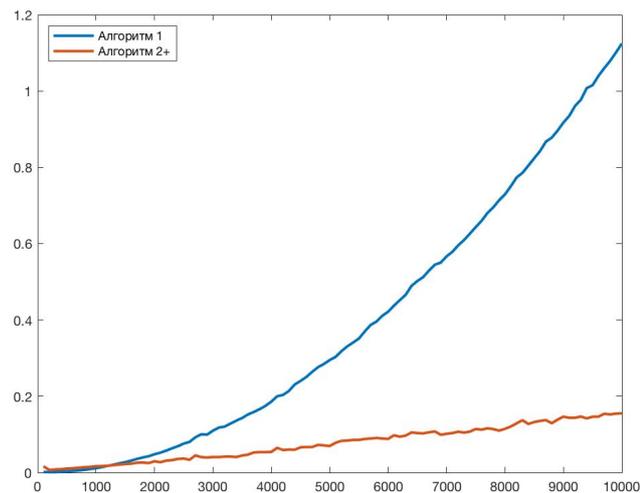


Рис. 2. Зависимость времени работы алгоритмов 1 и 2+ от размера матрицы. (а)  $r = 3$ ; (б)  $r = 10$ .

#### 4.2.2. Проверка сложности алгоритма для факторизации матриц ранга 2

Для случая неотрицательной матричной факторизации ранга 2 было проведено сравнение работы трех алгоритмов: алгоритма 3; алгоритма 1 при  $r = 2$ , матрица  $H$  затем находилась так же, как в алгоритме 3 (назовем такой двухэтапный метод алгоритмом 1+); метода, описанного в [18]

(исходный код взят из [19]).

Матрицы для экспериментов строились так же, как было описано выше,  $r = 2$ . На графике 3 демонстрируется время работы каждого из заявленных трех алгоритмов при изменении  $n$  от 100 до 2000 с шагом, равным 100. При этом для каждого  $n$  было проведено 100 экспериментов, результаты затем усреднялись.

Было получено, что алгоритмы 1+ и 3 верно построили разложение исходной матрицы, в то время как метод из [18] дает ошибку, увеличивающуюся с ростом  $n$ . Таким образом, приходим к выводу, что метод из [18] проигрывает остальным двум и в точности, и в быстроте.

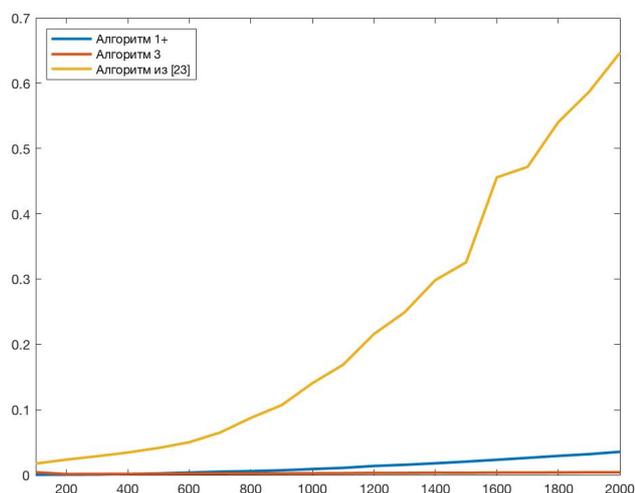


Рис. 3. Сравнение времени работы алгоритмов 1+, 3 и метода из [18] в зависимости от размера матрицы ранга 2.

Сравним, как изменяется время работы алгоритмов 1+ и 3 с изменением  $n$  от 100 до 10000 с шагом, равным 100. Результаты представлены на рисунке 4. Как видно, у алгоритма 1+ наблюдается квадратичная зависимость времени работы от размера матрицы, а у алгоритма 3 – линейная, что полностью подтверждает теоретические выкладки.

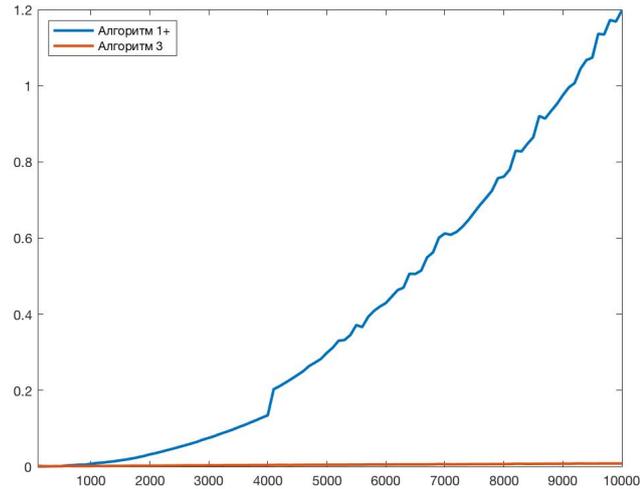


Рис. 4. Сравнение времени работы алгоритмов 1+ и 3 в зависимости от размера матрицы ранга 2.

#### 4.2.3. Проверка сложности редуцированного алгоритма для симметричной неотрицательной факторизации

Как обсуждалось в секции 1.3, алгоритм 2+ можно применять для построения симметричной неотрицательной факторизации симметричных неотрицательных матриц ранга  $r$ , содержащих диагональную главную подматрицу того же ранга. Сравним время его работы и метода IREVA [20].

Для тестирования генерировались неотрицательные симметричные матрицы  $M \in R_{\geq 0}^{n \times n} = WW^T$  ранга  $r$  с диагональной главной подматрицей того же ранга. Матрицы  $W^T \in R^{r \times n} = [I_r \ W']$ , где  $W' \in R^{r \times (n-r)}$  заполняются модулями нормально распределенных чисел, при этом столбцы  $W'$  нормируются так, чтобы сумма элементов в каждом была равна 1, и случайным образом перемешиваются.

Алгоритм IREVA можно условно поделить на 2 части. В первой вычисляется матрица  $V \in \mathbb{R}^{n \times r} : M = VV^T$ , у которой первый столбец состоит только из положительных чисел. При реализации метода было использовано предложение из [20], и для построения  $V$  использовалась комбинация формулы понижения ранга и разложения Холецкого. Во второй части алгоритма вычисляется ортогональная матрица  $R \in \mathbb{R}^{r \times r}$ , чтобы перевести столбцы  $V$  в  $\mathbb{R}_+^r$ .

Рассматривались  $n$  от 100 до 2500 с шагом 100, для каждого  $n$  строились 100 матриц  $M$ , которые подавались на вход доработанному алгоритму 2+ и методу IREVA (см. [20]). Ниже приведен график 5 зависимости времени работы от значения  $n$  при фиксированном  $r = 3$ . Как видно, у алгоритма IREVA наблюдается квадратичная зависимость времени работы от размера матрицы, а у алгоритма 2+ – линейная, что согласуется с нашими рассуждениями.

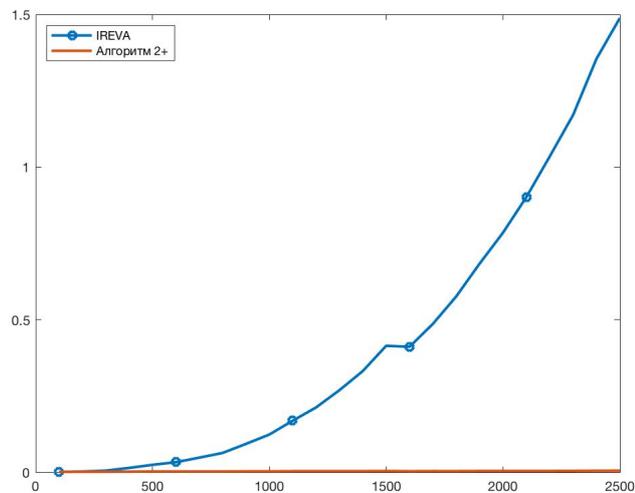


Рис. 5. Сравнение времени работы алгоритмов 2+ и IREVA (см. [20]) в зависимости от размера матрицы,  $r = 3$ .

#### 4.2.4. Редуцированный алгоритм для возмущенных сепарабельных матриц

Проверим работу алгоритма 2+ для возмущенных сепарабельных матриц. Будем строить матрицы для экспериментов так же, как в [15].

1. Матрица  $W$  будет генерироваться двумя способами:

- (а) *Равномерное распределение.* Элементы  $W \in R^{200 \times 20}$  – равномерно распределенные на отрезке  $[0; 1]$  случайные числа (функция `rand()` в MATLAB).

(б) *Плохо обусловленная.* Сперва генерируем  $W' \in R^{200 \times 20}$ , как описано в предыдущем пункте. Затем находится SVD представление:  $W' = U \Sigma V^T$ ,  $U \in R^{200 \times 20}$ ,  $\Sigma \in R^{20 \times 20}$ ,  $V \in R^{20 \times 20}$ . Тогда  $W = U S V^T$ , где  $S$  – диагональная матрица, у которой элементы на диагонали равны  $\alpha^{i-1}$ ,  $i = 1, 2, \dots, 20$ , а  $\alpha^{19} = 10^{-3}$ .

2. Матрицы  $H$ ,  $N$  тоже будут генерироваться двумя способами:

(в) *Средние точки.* Пусть  $H = [I_{20}, H'] \in R^{20 \times 210}$ , где столбцы  $H'$  содержат все возможные комбинации 2 ненулевых элементов, равных 0.5, в разных позициях. Поэтому у  $H'$   $C_{20}^2 = 190$  столбцов. Таким образом, первые 20 столбцов  $W$  совпадают со столбцами  $M$ , в то время как оставшиеся 190 являются средним каких-то двух столбцов  $M$ . Первые 20 столбцов  $M$  не изменяются, а к остальным добавляется шум –  $n_i = \delta(m_i - \bar{w})$ ,  $21 \leq i \leq 210$ ,  $\delta \geq 0$ , где  $\bar{w}$  – среднее столбцов  $W$ .

(г) *Распределения Дирихле и Гаусса.* Пусть  $H = [I_{20}, I_{20}, H'] \in R^{20 \times 240}$ , где столбцы  $H'$  получены из распределения Дирихле с  $r$  параметрами, выбранными равномерно на  $[0, 1]$ , то есть сумма элементов в каждом столбце  $H'$  равна 1. Значения матрицы шума  $N$  – нормально распределенные случайные числа:  $n_{ki} \sim \delta N(0, 1)$ ,  $1 \leq k \leq 200, 1 \leq i \leq 240$ .

В итоге возмущенная сепарабельная матрица  $M' = WH + N$  может быть получена четырьмя различными способами путем комбинации  $W$ ,  $H$  и  $N$ , построенных как описано выше.

Ниже приведены две таблицы. В табл. 3 указаны измеренные значения параметров исходных матриц, усредненные для всех экспериментов. Они практически полностью совпали с теми, что были получены в [15]: максимальное абсолютное отклонение составляет 0.01. Далее проанализировано, при каком наибольшем  $\delta$  алгоритмы 1 ([15]) и 2+ способны правильно идентифицировать столбцы, входящие в  $W$ . Результаты представлены в табл. 4. Заметим, что в экспериментах 1 – 3 максимальные значения  $\delta$  для алгоритма 2+ больше. Единственный случай, когда алгоритм 2+ несколько

уступил алгоритму 1 в случае возмущенной сепарабельной матрицы – эксперимент 4. Однако отметим, что при этом разница между максимальными  $\delta$  очень мала, то есть отличие представляется несущественным.

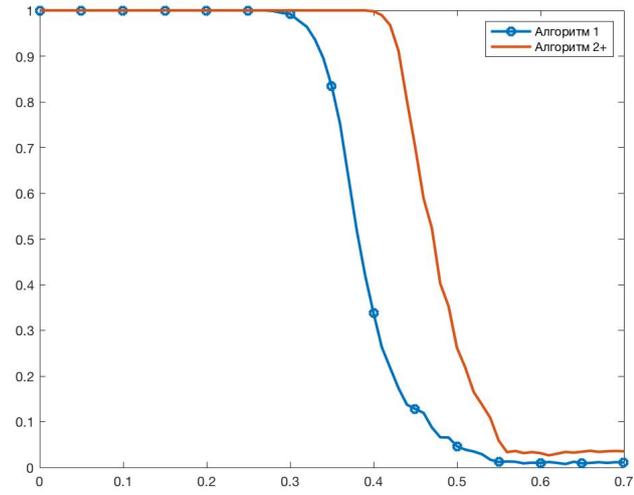
Таблица 3. Средние значения параметров для различных экспериментов при генерации зашумленных сепарабельных матриц.

Параметр	Эксперимент			
	1	2	3	4
$W$	(а)	(а)	(б)	(б)
$N$ и $H$	(в)	(г)	(в)	(г)
$\frac{\sigma_1(W)}{\sigma_r(W)}$	10.84	10.84	1000	1000
$\mu(W)$	8.64	8.64	0.41	0.41
$\sigma_r(W)$	2.95	2.95	$10^{-3}$	$10^{-3}$
Среднее $\max_i \ n_i\  \delta^{-1}$	3.06	16.15	0.30	16.15

Таблица 4. Максимальные значения  $\delta$  для верного определения столбцов  $W$  алгоритмом.

Алгоритм	Эксперимент			
	1	2	3	4
1	0.24	0.23	0.01	$20 \times 10^{-4}$
2+	0.39	0.25	0.38	$1 \times 10^{-4}$

(а)



(б)

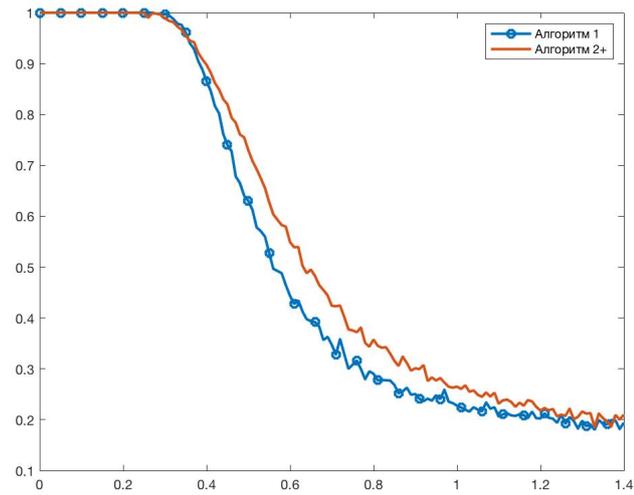
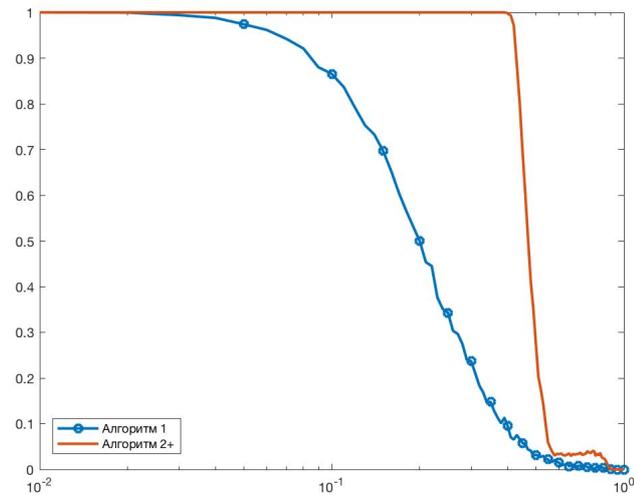


Рис. 6. Сравнение алгоритмов 1 и 2+ для возмущенных сепарабельных матриц. Зависимость доли верно найденных столбцов определяющей системы от  $\delta$ . (а) эксперимент 1; (б) эксперимент 2.

(а)



(б)

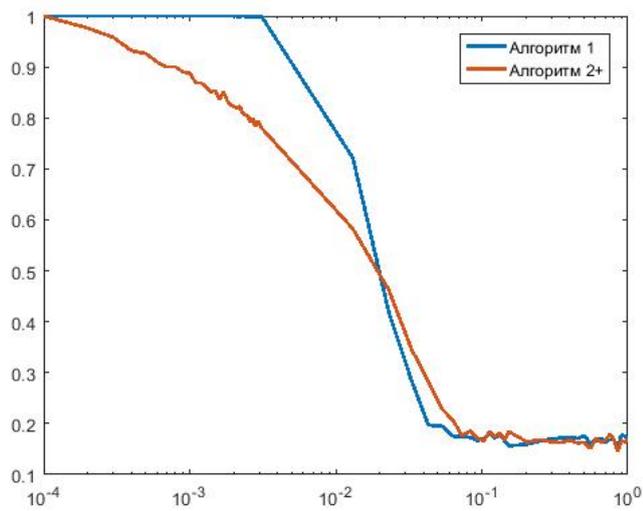


Рис. 7. Сравнение алгоритмов 1 и 2+ для возмущенных сепарабельных матриц. Зависимость доли верно найденных столбцов определяющей системы от  $\delta$ . (а) эксперимент 3; (б) эксперимент 4.

Итак, в случае возмущенных сепарабельных матриц алгоритм 2+ в большинстве случаев по качеству не уступает исходному алгоритму 1, имея при этом явное преимущество в скорости работы.

## 4.3. Результаты численных экспериментов для неотрицательной тензорной факторизации

### 4.3.1. Результаты NTTF на искусственных данных

Метод NTTF был реализован программно в Matlab. В качестве алгоритма NMF был выбран метод lraNMF, где крестовый метод из TT-Toolbox [28] комбинируется с lraNMF\_HALS из TDALAB [37]. Для всех экспериментов, представленных здесь, относительная погрешность считается как  $\frac{\|A-B\|_F}{\|A\|_F} \times 100\%$ , где  $B$  - построенный неотрицательный ТТ для аппроксимации исходного неотрицательного тензора  $A$ .

Для этого эксперимента генерировались случайные тензорные поездки с положительными элементами с помощью функции из TT-toolbox [28]. Алгоритм 4 применялся к их полным формам для сравнения рангов исходного и результирующего неотрицательных тензорных поездов. Были рассмотрены  $10 \times 10 \times 10 \times 10$ ,  $100 \times 100 \times 10 \times 10$  и  $100 \times 100 \times 100 \times 100$  тензоры. Для каждого эксперимента были сгенерированы 100 тензоров, результаты представлены в таблице 5. Для максимального и минимального рангов самое большое и самое маленькое значения соответственно среди каждого  $r_k$ ,  $k = \overline{1, 3}$  были выбраны. Найденные неотрицательные ТТ-ранги не всегда совпадают с исходными, но разница между максимальным и минимальным рангами приемлема, поскольку в любом случае достигается серьезный выигрыш в памяти.

Размер	Относит. ошибка(%)	Время(с., $\pm std$ )	Средние ранги	Мин. ранги	Макс. ранги
$10 \times 10 \times 10 \times 10$	1.14	$0.12 \pm 0.01$	5, 5, 6	4, 5, 5	5, 8, 8
$100 \times 100 \times 10 \times 10$	1.2	$0.6 \pm 0.05$	5, 5, 8	4, 5, 4	5, 5, 10
$100 \times 100 \times 100 \times 100$	1.66	$75.6 \pm 3$	5, 5, 8	5, 5, 5	5, 5, 8

Таблица 5. Применение NTTF (алгоритм 4) к положительным тензорам с неотрицательными ТТ-рангами  $r_1 = r_2 = r_3 = 5$ .

### 4.3.2. Результаты NTTF и метода с коррекцией отрицательных элементов при аппроксимации ядра для уравнения Смолуховского

Методы неотрицательной факторизации могут применяться в задачах физики. Уравнение Смолуховского - математическая модель процессов коагуляции при неупругих соударениях огромного числа частиц. Такие столкновения играют важную роль в различных физических процессах, например аэродинамике, полимеризации, свертывании крови и даже в формировании звезд. Когда частицы равномерно расположены в пространстве и допускаются только бинарные столкновения, мы получаем уравнение коагуляции Смолуховского:

$$\frac{\partial n(v, t)}{\partial t} = \frac{1}{2} \int_0^v K(v-u, u) n(v-u, t) n(u, t) du - n(v, t) \int_0^\infty K(v, u) n(u, t) du.$$

Это уравнение описывает эволюцию функции концентрации  $n(v, t)$  частиц размера  $v$  на единицу объема окружающей среды в момент  $t$ . Когда частицы состоят из  $l$  разных компонент, их размеры формируют вектор  $\bar{v} = (v_1, \dots, v_l)$ . Тогда коагуляционный процесс описывается мультикомпонентной версией уравнения Смолуховского. Обычно ядро коагуляции  $K(\bar{v}, \bar{u})$  зависит от рассматриваемой физической системы. В [45] неотрицательные функции были использованы в качестве ядер, для них строились ТТ-аппроксимации. Но не было гарантии выполнения условия неотрицательности. NTTF может помочь решить эту проблему. Было рассмотрено ядро коагуляции  $K(\bar{u}, \bar{v}) = (u_1 + u_2)^\mu (v_1 + v_2)^\nu + (u_1 + u_2)^\nu (v_1 + v_2)^\mu$ ,  $u_i \geq 0$ ,  $i = \overline{1, 2}$ ,  $v_i \geq 0$ ,  $i = \overline{1, 2}$ , где выбирались различные параметры  $\mu, \nu$  так, что  $\mu + \nu \leq 1$ ,  $|\mu - \nu| \leq 2$ . Переменные  $u_i, v_i, i = \overline{1, 2}$  принимают значения от 0.1 до 10 с шагом 0.1. Так мы получаем тензор четвертого порядка, у которого  $10^8$  элементов. Для каждой пары  $\mu, \nu$  эксперименты были проведены 10 раз. Результаты тестов представлены в таблице 7.

Как видно, даже в худшем случае для неотрицательного тензорного ядра нужно хранить менее одного процента исходного числа значений. В то же время точность хорошая и время работы программы приемлемое

для тензоров такого размера.

$\mu$	$\nu$	Относит. ошибка(%)	Время(с., $\pm std$ )	Средние ранги	Мин. ранги	Макс. ранги
0.1	-0.05	0.14	$55.8 \pm 4.5$	2, 21, 22	2, 2, 2	2, 96, 100
0.2	0.1	0.12	$63.2 \pm 2.8$	3, 25, 41	3, 1, 3	3, 85, 100

Таблица 6. Применение NTTF (алгоритм 4) к  $K(\bar{u}, \bar{v})$ , размер:  $100 \times 100 \times 100 \times 100$

В следующем эксперименте параметры  $\mu = 0.2$  и  $\nu = 0.1$ . Степенной метод для тензора сходится за 300 итераций к верному значению, что занимает 0.2 секунды. В данном случае проблема в том, что два максимальных значения очень близки друг к другу, но, несмотря на это, ТТ-ранги получаются небольшими. Однако минимизация, основанная на ТТ-cross из [28] занимает меньше времени, поэтому при проведении экспериментов предпочтение было отдано именно ей, хотя в общем случае применение метода может потребовать некоторой настройки параметров для поиска экстремума.

Для эксперимента три различных ТТ разложения из [28] были протестированы. Для сравнения рассматривается NTTF метод из 2.2.

Метод	Относит. ошибка(%)	Время(с., $\pm std$ )	Средние ранги	Мин.ранги	Макс. ранги
NTTF	0.12	$63.2 \pm 2.8$	3, 25, 41	3, 1, 3	3, 85, 100
AMEN-CROSS + коррекция элементов	0.0027	$1.3 \pm 0.5$	7, 3, 7	7, 3, 7	7, 3, 7
DMRG-CROSS + коррекция элементов	0.0471	$9.2 \pm 5.2$	7, 5, 7	7, 5, 7	7, 5, 7
ТТ-SVD + коррекция элементов	0.0338	$4.3 \pm 0.1$	5, 3, 5	5, 3, 5	5, 3, 5

Таблица 7. Применение методов к  $K(\bar{u}, \bar{v})$ , размер:  $100 \times 100 \times 100 \times 100$

Этот пример имеет хорошие ТТ-аппроксимации, поскольку появляющиеся отрицательные значения очень малы, и при добавлении константы все элементы становятся положительными. При этом относительная ошибка меняется незначительно и ранги увеличиваются всего на единицу. Результаты таблицы 7 подтверждают, что предложенный подход точечной коррекции отрицательных элементов для факторизации неотрицательных тензорных поездов дает лучшие результаты в рассматриваемом приложении.

### 4.3.3. Результаты методов неотрицательной тензорной факторизации для ERP (event related potential) тензора

Потенциал, связанный с событием (ERP – event related potential) – это измеренный отклик мозга, который формируется в ответ на сенсорное, когнитивное либо двигательное явление. В [5] тензор ERP, полученный из данных потенциалов, вызванных событием, был рассмотрен как приложение и пример для численных экспериментов. Целью было построить гибридное разложение NTT-Tucker тензора ERP. Метод, предложенный в [5], состоит из двух шагов. Сначала неотрицательный тензорный поезд (NTT) строится с помощью алгоритма NTT-HALS. Затем каждый вагон, являющийся тензором третьего порядка, факторизуется в тензор третьего порядка и матрицу-фактор, чтобы получить разложение Таккера.

Данные представлены в форме тензора четвертого порядка, чей размер  $71 \times 60 \times 42 \times 9$  [38, 39]. Размерности отвечают спектральной (частотный интервал), временной, объектной (2 группы из 21 ребенка) и пространственной (электрод) модам.

В [5] авторы представляют результаты для построения NTT-Tucker разложения тензора ERP. Они применяют двухэтапный метод, используя алгоритм NTT-HALS. К сожалению, авторы не приводят отдельных результатов для каждого этапа. Планируется провести больше сравнительных тестов с NTT-HALS в будущем.

Можно получить оценки рангов неотрицательного тензорного поезда из [5]. Они равны 4, 10, 6. Вычисления для алгоритма 4 с фиксированными рангами повторили 100 раз. Средняя относительная ошибка для результирующей ТТ-аппроксимации – 13.6%, что близко к 13.8% для NTT-Tucker из [5]. Время для построения неотрицательного ТТ с такой точностью в среднем 2.5 секунды, тогда как алгоритм для построения NTT-Tucker требует 4.4 секунды [5].

Интересным представляется, какие ранги определит алгоритм 6 для этих данных. Ответ – 4, 12, 9. Алгоритм 6 был применен 100 раз к тензору ERP, результаты усреднены.

Можно проверить качество неотрицательной ТТ-аппроксимации с ран-

гами 4, 10, 6, которую построит алгоритм 5 (300 проходов) для этих данных. Значения, приведенные в таблице 8, - усредненные результаты для 100 повторений эксперимента.

Метод	Относительная ошибка(%)	Время(.)
<i>NTTF</i>	13.6	2.5
Алгоритм 5	13.6	4.1

Таблица 8. Применение NTTF (4) и алгоритма 5 к тензору ERP, неотрицательные ТТ-ранги равны 4, 10, 6.

В этом примере с точки зрения времени и относительной ошибки были достигнуты лучшие результаты при построении неотрицательного ТТ с помощью метода, предложенного в разделе 2.2.

#### 4.3.4. Анализ алгоритма NTТ-MU (сложность, сходимость метода)

Методы 5 и 6 были реализованы в Matlab. В качестве NMF алгоритма *lraNMF* был выбран, где крестовый метод из ТТ-Toolbox [28] комбинируется с *lraNMF\_HALS* из TDALAB [37]. Для всех представленных экспериментов относительная ошибка считается как  $\frac{\|A-B\|_F}{\|A\|_F} \times 100\%$ , где  $B$  - построенная неотрицательная ТТ-аппроксимация исходного неотрицательного тензора  $A$ .

Цель эксперимента – подтвердить линейную зависимость от максимального размера моды тензора времени прохода в алгоритме 5. Метод 5 применялся к тензорам размера  $n \times n \times n \times n$ , где значение  $n$  изменялось от 10 до 100 с шагом 10, и время на 1 проход алгоритма вычислялось. Для каждого  $n$  рассматривались 100 тензоров, результаты усреднялись. ТТ с положительными элементами генерировались функцией из ТТ-toolbox [28], ранги – 10, 10, 10. Алгоритм 5 строил аппроксимацию с неотрицательными рангами 5, 5, 5. Результаты на рис. 8 теоретические предположения с несколькими понятными отклонениями.

Также численные эксперименты демонстрируют, что относительная ошибка уменьшается после каждого прохода, доказывая правильность алгоритма 5. ТТ с положительными элементами размера  $100 \times 100 \times 100 \times 100$  и

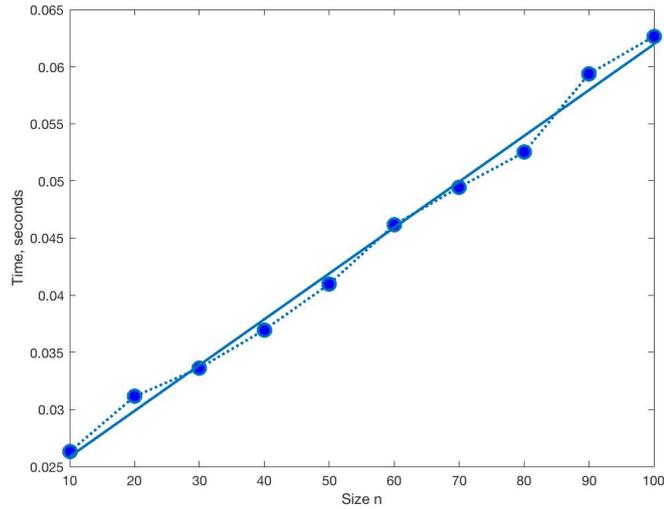


Рис. 8. Зависимость времени 1 прохода алгоритма 5 от размера моды тензора.

рангами 5, 5, 5 был сгенерирован с помощью функции из TT-toolbox [28]. Затем NTT-MU (алгоритм 5) был применен к этому разложению для поиска аппроксимации с рангами 4, 4, 4, и после каждого прохода относительная ошибка измерялась. Для инициализации 10 проходов TT-ALS из [28] были сделаны. Результат 5000 проходов представлен на рис. 9.

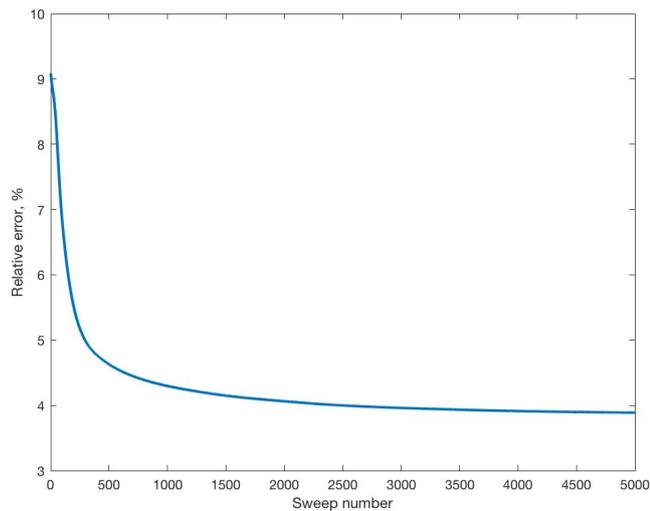


Рис. 9. Алгоритм 5, где для инициализации 10 проходов TT-ALS из [28] были сделаны: зависимость относительной ошибки (%) от числа проходов.

По Рис. 9 видно, что наступает момент, когда снижение относительной ошибки значительно замедляется. В такой ситуации есть два пути. Если точность удовлетворительная, имеет смысл остановить процесс. Другой вариант - использовать алгоритм 6 в попытке найти более подходящие неотрицательные ТТ-ранги.

#### 4.3.5. Сравнение NTTF и NTT-MU с DMRG на искусственных данных

Для этого эксперимента случайные тензорные поездки с положительными элементами были сгенерированы с помощью функции из TT-toolbox [28]. Алгоритм 6 и NTTF применялись к их полным формам для сравнения рангов исходного и результирующего неотрицательных ТТ. Были рассмотрены  $10 \times 10 \times 10 \times 10$ ,  $100 \times 100 \times 10 \times 10$  и  $100 \times 100 \times 100 \times 100$  тензоры. Для каждого эксперимента 100 тензоров были сгенерированы, усредненные результаты представлены в таблице 9. Для максимального и минимального рангов самое большое и самое маленькое значения соответственно среди каждого  $r_k$ ,  $k = \overline{1, 3}$  были выбраны.

Для алгоритма 6 и NTTF 4 в качестве NMF алгоритма метод неотрицательной матричной факторизации, основанный на малоранговых приближениях, из [36] был выбран. Малоранговые приближения находились с помощью крестового метода из TT-toolbox [28].

Исходные неотрицательные ТТ-ранги  $r_k$   $k = \overline{1, 3}$  были приравнены 2. Метод ТТ-SVD из [28] применялся для построения аппроксимации исходного тензора и 10 проходов ТТ-ALS из [28] были сделаны для инициализации неотрицательного тензорного поезда.

Метод	Размер	Относит. ошибка(%)	Время(., $\pm std$ )	Средние ранги	Мин.ранги	Макс. ранги
<i>NTTF</i>	$10 \times 10 \times 10 \times 10$	1.6	$0.12 \pm 0.01$	5, 5, 5	4, 3, 4	5, 5, 5
Algorithm 6	$10 \times 10 \times 10 \times 10$	3.5	$0.14 \pm 0.01$	4, 4, 4	3, 2, 3	5, 5, 5
<i>NTTF</i>	$100 \times 100 \times 10 \times 10$	1.6	$0.5 \pm 0.04$	5, 5, 5	5, 5, 5	5, 5, 7
Algorithm 6	$100 \times 100 \times 10 \times 10$	3.5	$0.3 \pm 0.03$	6, 6, 5	6, 5, 4	6, 6, 5
<i>NTTF</i>	$100 \times 100 \times 100 \times 100$	1.7	$73.6 \pm 2.16$	5, 5, 7	5, 5, 5	5, 5, 12
Algorithm 6	$100 \times 100 \times 100 \times 100$	3.7	$4.4 \pm 0.16$	5, 5, 5	5, 4, 5	5, 5, 5

Таблица 9. Применение NTTF и алгоритма 6 к положительным тензорам с неотрицательными ТТ-рангами  $r_1 = r_2 = r_3 = 5$ .

Исходя из таблицы 9, можно заключить следующее. Оба алгоритма находят неотрицательные ТТ-ранги, которые отличаются от настоящих не более чем на 2. С увеличением размера тензора алгоритм 6 продемонстрировал гораздо более быструю работу, однако он также систематически строил аппроксимацию с большей относительной погрешностью.

Если для алгоритма 5 относительная ошибка не возрастает после каждого прохода, это может быть не так для алгоритма 6. Такое может произойти из-за NMF аппроксимации. Поэтому предлагается использовать алгоритм 6 для оценки неотрицательных ТТ-рангов, а затем дальше улучшать построенную аппроксимацию с помощью алгоритма 5.

Например, для одного из тензоров размера  $100 \times 100 \times 100 \times 100$  из предыдущего эксперимента 2 проходы алгоритма 6 дали неотрицательную ТТ-аппроксимацию с  $r_1 = 5, r_2 = 5, r_3 = 5$  и относительной ошибкой, равной 3.15%. Тогда алгоритм 5 был применен, и после 642 итераций он построил неотрицательный ТТ с относительной погрешностью 1.7%, изменение относительной ошибки после каждого прохода представлено на графике 10. В итоге неотрицательный тензорный поезд с той же относительной погрешностью, что и у NTTF, построен. Но весь процесс занял примерно 29 секунд. NTTF потребовалось в среднем в 2.5 раза дольше, чтобы достичь схожего результата (Таблица 9).

#### **4.3.6. Анализ алгоритма для построения неотрицательного разложения Таккера на основе тензорного поезда**

Прежде всего было проверено, насколько ускоряет использование ТТ разложения алгоритм из [57]. Для чистоты эксперимента одна итерация каждого алгоритма была рассмотрена для тензоров размеров  $50 \times 50 \times 50 \times 50$  и  $100 \times 100 \times 100 \times 100$ . Каждый эксперимент повторялся 10 раз, все результаты были усреднены. Сперва был сгенерирован тензорный поезд с положительными элементами, у которого все ранги были равны 5. Он был использован как исходный для версии алгоритма, основанной на ТТ, в то время как изначальный метод из [57] применялся к его полной форме. Результаты представлены в таблице ниже.

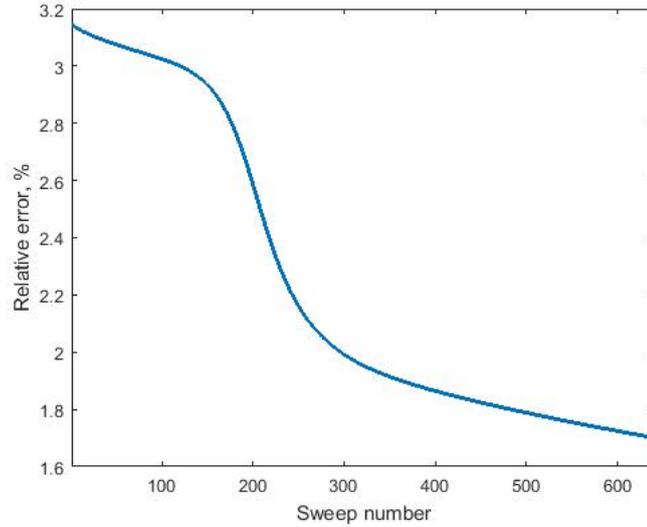


Рис. 10. Алгоритм 5, инициализированный результатом работы двух проходов алгоритма 6: зависимость относительной ошибки (%) от числа проходов.

Метод	Размер	Время(с.)
NTD с APG	$50 \times 50 \times 50 \times 50$	0.14
NTD с APG, основанный на ТТ	$50 \times 50 \times 50 \times 50$	0.02
NTD с APG	$100 \times 100 \times 100 \times 100$	2.96
NTD с APG, основанный на ТТ	$100 \times 100 \times 100 \times 100$	0.03

Таблица 10. Одна итерация алгоритмов, примененная к положительным тензорам с неотрицательными NTD рангами, равными 5.

Как и ожидалось, видим, что алгоритм, основанный на ТТ, работает гораздо быстрее.

Теперь рассмотрим ядро коагуляции для уравнения Смолуховского из секции 4.3.2. Для первого примера переменные  $u_i, v_i, i = \overline{1, 2}$ , находятся в диапазоне от 0 до 5 с шагом 0.1, для второго они варьируются от 0 до 10. Ранги для неотрицательного разложения Таккера были выбраны равными 5. Для предварительного разложения в ТТ использовался метод ТТ-SVD. Каждый эксперимент был проведен 10 раз, усредненные результаты представлены в таблице 11.

Метод	Размер	Относительная погрешность(%)	Время (с.)
NTD with APG	$50 \times 50 \times 50 \times 50$	0.8	80.2
NTD with APG, основанный на ТТ	$50 \times 50 \times 50 \times 50$	0.8	6.5
NTD with APG	$100 \times 100 \times 100 \times 100$	2.4	638.5
NTD with APG , основанный на ТТ	$100 \times 100 \times 100 \times 100$	2.3	8.2

Таблица 11. Алгоритмы факторизации в неотрицательное разложение Таккера применительно к ядру коагуляции Смолуховского.

Несмотря на необходимость в дополнительном времени для расчета ТТ аппроксимации с высокой точностью, для того чтобы не ухудшить относительную погрешность, вычисления становятся быстрее в несколько раз.

#### 4.3.7. Анализ алгоритма для построения неотрицательного канонического разложения на основе тензорного поезда

Для экспериментов был выбран метод NCP-HALS по двум причинам. Он хорошо известен как быстро работающий метод и дополнительным плюсом является то, что необязательно использовать именно неотрицательный тензорный поезд, так как алгоритм уже включает поправку возможных отрицательных элементов, возникающих при построении. Прежде всего, как это было сделано с методом для неотрицательного разложения Таккера, было проверено, насколько быстрее алгоритм, использующий ТТ, по сравнению с изначальным. Вновь в целях объективности эксперимента был сгенерирован положительный тензорный поезд с рангами 5. Он был использован в качестве входных данных для версии алгоритма, основанной на ТТ, в то время как для первоначального метода из [55] была применена полная форма тензора. Усредненные результаты для 10 повторений каждого эксперимента представлены в таблице ниже.

Метод	Размер	Относительная погрешность(%)	Время (с.)
FAST-HALS NTF	$50 \times 50 \times 50 \times 50$	6.8	6.6
FAST-HALS NTF с ТТ	$50 \times 50 \times 50 \times 50$	2.1	6.8
FAST-HALS NTF	$100 \times 100 \times 100 \times 100$	135.2	6.5
FAST-HALS NTF с ТТ	$100 \times 100 \times 100 \times 100$	2.6	6.5

Таблица 12. Сравнение алгоритмов, примененных к положительным тензорам с неотрицательными каноническими рангами 5.

Так как было использована одинаковая случайная инициализация для

обоих алгоритмов, подтверждается, что метод 9 предоставляет те же результаты значительно быстрее.

Далее используются алгоритмы, чтобы построить неотрицательное каноническое разложение для ядра уравнения Смолуховского из секции 4.3.2. Для первого примера переменные  $u_i, v_i, i = \overline{1, 2}$  варьируются от 0 до 5 с шагом 0.1, а для второго примера они изменяются от 0 до 10 с шагом 0.1. Ранги для неотрицательного канонического разложения были выбраны равными 5. В качестве метода для построения предварительного ТТ разложения был использован ТТ-SVD. Значения в таблице 13 являются усредненными результатами для 10 повторений каждого теста.

Метод	Размер	Относительная погрешность(%)	Время (с.)
FAST-HALS NTF	$50 \times 50 \times 50 \times 50$	0.5	8
FAST-HALS NTF с ТТ	$50 \times 50 \times 50 \times 50$	0.5	2.4
FAST-HALS NTF	$100 \times 100 \times 100 \times 100$	0.6	146.1
FAST-HALS NTF с ТТ	$100 \times 100 \times 100 \times 100$	0.6	8

Таблица 13. Сравнение алгоритмов, использованных для разложения ядра уравнения Смолуховского.

Из таблицы 13 следует, что высокая точность ТТ аппроксимации позволяет сохранить ту же относительную ошибку, что и для изначального тензора. И при этом все же наблюдается значительное сокращение времени, требуемого на вычисления, причем применение метода ТТ-CROSS может позволить ускорить алгоритм еще больше.

#### 4.4. Численные эксперименты по сравнению двух подходов к построению неотрицательных разложения Таккера и и ТТ

В этом разделе оценивается и сравнивается производительность детерминированных и рандомизированных реализаций алгоритмов NTTSVD (и NSTHOSVD) из [58], NTTF и NTT-MU из 2.2 и 2.4 соответственно и APG на основе ТТ из 2.8. К рандомизированным алгоритмам относятся НМТ и Тгорр, описание методов приведено в [58]. Ради справедливого сравнения были рассмотрены те же примеры, что и в [58]:

- тензор Гильберта,
- смесь многомерных гауссианов,
- гиперспектральное изображение.

#### 4.4.1. Синтетические данные

Нашим первым примером неотрицательного тензора приблизительно низкого ранга является тензор Гильберта

$$\mathbf{X}(i_1, \dots, i_d) = \frac{1}{i_1 + \dots + i_d - d + 1},$$

, который является многомерным обобщением хорошо известной матрицы Гильберта. Мы применяем эти методы к трехмерному тензору Гильберта размером  $128 \times 128 \times 128$  с рангами Таккера  $\mathbf{r} = (3, 2, 4)$  и ТТ-рангами  $\mathbf{r} = (3, 2)$ .

Мы определяем два алгоритма для методов, основанных на двухэтапном подходе, где первый используется для низкоранговой аппроксимации исходного тензора, тогда как второй для построения тензорной декомпозиции (ТТ или Tucker) с неотрицательными множителями. Например, для построения неотрицательного тензорного ряда с помощью двухэтапного подхода для тензора Гильберта была использована комбинация метода DMRG cross [67] с NTT-MU.

Таблица 14. Сравнение различных подходов к низкоранговой неотрицательной тензорной аппроксимации тензора Гильберта  $128 \times 128 \times 128$  с рангами Таккера  $(3, 2, 4)$  и ТТ-рангами  $(3, 2)$ : время выполнения, количество итераций, относительные ошибки и отрицательные элементы.

Метод	Время (с.)	Число итера- ций	Относи- тельная ошибка (Фробе- ниус)	Относи- тельная ошибка (Чебы- шев)	Отрица- тельный эл. (Фробе- ниус)	Отрица- тельные эл. (%)
TTSVD	0.1	–	$7.72 \cdot 10^{-2}$	$3.67 \cdot 10^{-1}$	$9.7 \cdot 10^{-2}$	$6.3 \cdot 10^{-3}$
NTTSVD, SVD <sub>r</sub> [58]	18.2	250	$7.88 \cdot 10^{-2}$	$3.94 \cdot 10^{-1}$	0	0
NTTSVD, HMT(1, 12) [58]	5.8	250	$7.88 \cdot 10^{-2}$	$3.94 \cdot 10^{-1}$	$1.7 \cdot 10^{-16}$	$4.8 \cdot 10^{-5}$
NTTSVD, HMT(0, 15) [58]	4.9	250	$7.88 \cdot 10^{-2}$	$3.94 \cdot 10^{-1}$	$2.1 \cdot 10^{-16}$	$1.4 \cdot 10^{-4}$
NTTSVD, Tropp(4, 30) [58]	3.7	250	$8.2 \cdot 10^{-2}$	$3.68 \cdot 10^{-1}$	0	0
DMRG_CROSS + NTT-MU	2.0	450	$7.88 \cdot 10^{-2}$	$4.1 \cdot 10^{-1}$	0	0
STHOSVD [58]	0.1	-	$7.72 \cdot 10^{-2}$	$3.67 \cdot 10^{-1}$	$9.7 \cdot 10^{-2}$	$6.3 \cdot 10^{-3}$
NSTHOSVD, SVD <sub>r</sub> [58]	22	250	$7.89 \cdot 10^{-2}$	$3.95 \cdot 10^{-1}$	0	0
NSTHOSVD, HMT(1, 11) [58]	13.5	250	$7.89 \cdot 10^{-2}$	$3.95 \cdot 10^{-1}$	0	0
NSTHOSVD, HMT(0, 15) [58]	11.2	250	$7.89 \cdot 10^{-2}$	$3.95 \cdot 10^{-1}$	0	0
NSTHOSVD, Tropp(6, 35) [58]	10	250	$7.9 \cdot 10^{-2}$	$3.95 \cdot 10^{-1}$	$3 \cdot 10^{-17}$	$4.8 \cdot 10^{-5}$
TTSVD APG, основанный на ТТ	+ 4.4	450	$7.87 \cdot 10^{-2}$	$4.1 \cdot 10^{-1}$	0	0

Во втором синтетическом эксперименте мы приводим результаты тестов неотрицательных приближений для представления низкого ранга много-

мерной смеси гауссианов

$$f(x) = \sum_{j=1}^m \alpha_j \exp \left( - (x - \mu_j)^\top A_j^{-1} (x - \mu_j) \right), \quad x \in \mathbb{R}^d,$$

с весами  $\alpha_j \in \mathbb{R}$  означает  $\mu_j \in \mathbb{R}^d$  и ковариационными матрицами  $A_j \in \mathbb{R}^{d \times d}$ . Каждый отдельный гауссиан имеет приблизительно низкий ранг (это тензор ранга 1, если  $A_j$  диагонален), поэтому смесь также может быть аппроксимирована. Мы рассматриваем смесь  $f(x)$  в гиперкубе  $[-a, a]^d$  и дискретизируем область на эквидистантной сетке тензорного произведения с шагом  $2a/(n-1)$ , что дает  $d$ -мерный  $n \times \dots \times n$  тензор  $\mathbf{X}$ .

Мы выбираем 4-мерный сценарий со сбалансированной смесью 2 гауссиан ( $\alpha_1 = \alpha_2$ ) следующими мат. ожиданиями,

$$\mu_1 = [0 \ 0 \ 0 \ 0]^\top, \quad \mu_2 = [0.5 \ -0.5 \ 0.5 \ -0.5]^\top,$$

и ковариационными матрицами,

$$A_1 = \begin{bmatrix} 0.403 & 0.236 & 0.159 & 0.188 \\ 0.236 & 0.422 & 0.193 & 0.313 \\ 0.159 & 0.193 & 0.124 & 0.164 \\ 0.188 & 0.313 & 0.164 & 0.288 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0.173 & 0.229 & 0.200 & 0.191 \\ 0.229 & 0.347 & 0.254 & 0.201 \\ 0.200 & 0.254 & 0.348 & 0.252 \\ 0.191 & 0.201 & 0.252 & 0.360 \end{bmatrix}.$$

Мы принимаем  $a = 1$  в качестве размера области и  $n = 64$  в качестве размера тензора.

В таблицах 14, 15 мы представляем результаты, полученные с помощью NTTSVD, NTT-MU для тензорной последовательности, а также NSTHOSVD и APG на основе ТТ для декомпозиции Таккера. Мы получаем, что применение NTT-MU и APG на основе ТТ приводит к полной очистке сжатого тензора и требует меньше времени по сравнению с подходами, основанными на чередующихся проекциях. Однако процесс внутренней очистки не требует от нас обработки всех элементов тензора, но может быть выполнен в низкоуровневом ТТ-формате.

Таблица 15. Сравнение различных подходов к малоранговой неотрицательной тензорной аппроксимации гауссовой смеси  $64 \times 64 \times 64 \times 64$  с рангами Таккера (14, 14, 14, 14) и ТТ-рангами (10, 20, 10): время выполнения, количество итераций, относительные ошибки и отрицательные элементы.

Метод	Время вы- пол- не- ния (сек.)	Число итера- ций	Отно- ситель- ная ошибка (Фробе- ниус)	Отно- ситель- ная ошиб- ка (Че- бы- шев)	Отрица- тельный эл. (Фро- бус)	Отрица- тельные эл. (%)
TTSVD	1.1	-	$7.4 \cdot 10^{-2}$	$1.5 \cdot 10^{-1}$	5.3	41.2
NTTSVD, SVD <sub>r</sub> [58]	259	200	$8.7 \cdot 10^{-2}$	$1.8 \cdot 10^{-1}$	$1.4 \cdot 10^{-2}$	1.1
NTTSVD, HMT(1, 40) [58]	254	200	$8.7 \cdot 10^{-2}$	$1.8 \cdot 10^{-1}$	$1.4 \cdot 10^{-2}$	1.1
NTTSVD, HMT(0, 45) [58]	217	200	$8.7 \cdot 10^{-2}$	$1.8 \cdot 10^{-1}$	$1.4 \cdot 10^{-2}$	0.9
NTTSVD, Tropp(38, 100) [58]	214	200	$9.1 \cdot 10^{-2}$	$1.9 \cdot 10^{-1}$	$1.3 \cdot 10^{-2}$	0.52
NTTSVD, Tropp(35, 100) [58]	202	200	$9.4 \cdot 10^{-2}$	$2 \cdot 10^{-1}$	$1.4 \cdot 10^{-2}$	0.62
NTTF + NTT-MU	146	4000 + 325	$10.8 \cdot 10^{-2}$	$2.0 \cdot 10^{-1}$	0	0
STHOSVD	2	-	$2.2 \cdot 10^{-2}$	$7.7 \cdot 10^{-2}$	1.8	38.4
NSTHOSVD, SVD <sub>r</sub> [58]	463	200	$2.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$1.6 \cdot 10^{-3}$	1.8
NSTHOSVD, HMT(1, 24) [58]	230	200	$2.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$1.6 \cdot 10^{-3}$	1.8
NSTHOSVD, HMT(0, 24) [58]	191	200	$2.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$1.7 \cdot 10^{-3}$	1.5
NSTHOSVD, Tropp(22, 80) [58]	192	200	$2.7 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$1.8 \cdot 10^{-3}$	1.4
NSTHOSVD, Tropp(18, 80) [58]	176	200	$2.8 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$1.7 \cdot 10^{-3}$	0.86
TTSVD + APG, ос- нованный на ТТ	94	3750	$4.3 \cdot 10^{-2}$	$1.5 \cdot 10^{-1}$	0	0

Чтобы продемонстрировать бесспорное превосходство двухэтапного подхода для крупномасштабных задач, мы рассмотрели тензор Гильберта размером  $1000 \times 1000 \times 1000$  факторизацию с рангами Таккера  $\mathbf{r} = (3, 2, 4)$  и ТТ-ранги  $\mathbf{r} = (3, 2)$ . Мы выбираем рандомизированную версию NTTSD с Тропп(4, 30) в соответствии с результатами из таблицы 14 как самый быстрый метод для построения неотрицательной аппроксимации ТТ низкого ранга с помощью первого подхода. Таким образом, одна итерация занимает 362,7 секунды и позволяет вычислить ТТ с относительной погрешностью 9,6%, содержащей 0,01% отрицательных значений. Норма Фробениуса отрицательной части в этом случае равна 0,41. В то же время комбинация метода AMEN cross [63] с 200 итерациями NTT-MU позволяет вычислить неотрицательный ТТ с относительной погрешностью 8,8% всего за 3,3 секунды. Используя те же рассуждения, что и ранее, мы выбираем NSTHOSVD с Тропп(6, 35) в качестве метода первого подхода для модели Таккера. В этом примере одна итерация занимает 278 секунд и позволяет вычислить разложение Таккера с относительной погрешностью 8,4%, содержащей 0,01% отрицательных значений. Норма Фробениуса отрицательной части в этом случае равна 0,31. Напротив, комбинация метода AMEN cross с 200 итерациями APG на основе ТТ вычисляет неотрицательную декомпозицию Таккера с относительной погрешностью 8,6% всего за 4,9 секунды. Этот пример наглядно показывает вычислительную эффективность методов двухэтапного подхода для работы с большими данными.

#### 4.4.2. Гиперспектральное изображение

Реальным примером является гиперспектральное изображение Национального торгового центра Вашингтона, округ Колумбия<sup>1</sup> размером  $307 \times 307 \times 191$ , последним измерением являются спектральные полосы (мы также линейно масштабируем элементы до  $[0, 1]$ ). Мы устанавливаем те же значения рангов Таккера (40, 40, 33) и ТТ-рангов (33, 33), что и в [58], и используем их в качестве более сложного базового примера для наших тестов. Чтобы измерить качество приближений низкого ранга, мы снова

<sup>1</sup>Данные доступны по адресу [https://github.com/JakobSig/HSI2RGB/blob/master/washington\\_hsi.mat](https://github.com/JakobSig/HSI2RGB/blob/master/washington_hsi.mat)

используем относительную погрешность в норме Фробениуса, среднее значение по полосе измерения индекса структурного сходства (SSIM [69, 70]) и статистический коэффициент детерминации  $R^2$ ,

$$R^2 = 1 - \frac{\|\mathbf{X} - \mathbf{Y}\|_F^2}{\|\mathbf{X} - \alpha\|_F^2}, \quad \alpha = \frac{1}{\prod_{k=1}^d n_k} \sum_{(i_1, \dots, i_d)} \mathbf{X}(i_1, \dots, i_d) \in \mathbb{R},$$

где  $\mathbf{Y}$  - аппроксимант низкого ранга.

Для этих сложных данных (см. таблицу 16) мы видим, что NTTF и APG на основе ГТ работают еще лучше по сравнению с чередующимися проекциями без существенной дополнительной ошибки и сопоставимых оценок SSIM и  $R^2$ . Методы как для форматов ГТ, так и для форматов Таккера работают в несколько раз быстрее.

Таблица 16. Сравнение неотрицательных тензорных факторизаций с моделью тензорного поезда для низкоранговой неотрицательной тензорной аппроксимации гиперспектрального изображения  $307 \times 307 \times 191$  Национального торгового центра Вашингтона с рангами Таккера (40, 40, 33) и ТТ-рангами (33, 33): время выполнения, количество итераций, относительные ошибки, отрицательные элементы, SSIM и  $R^2$ .

Метод	Время выпол- нения (сек.)	Коли- чество итера- ций	Относи- тельная ошибка (Фробе- ниус)	SSIM	$R^2$	Отрица- тельные эл. (Фро- бени- ус)
TTSVD	2.9	-	$1.8 \cdot 10^{-1}$	0.67	0.94	2.1
NTTSVD, $SVD_r$ [58]	304.4	100	$1.8 \cdot 10^{-1}$	0.64	0.94	$6.5 \cdot 10^{-3}$
NTTSVD, HMT(1, 75) [58]	61	100	$1.8 \cdot 10^{-1}$	0.63	0.94	$6.6 \cdot 10^{-3}$
NTTSVD, HMT(0, 75) [58]	39	100	$2.0 \cdot 10^{-1}$	0.58	0.93	$0.7 \cdot 10^{-2}$
NTTSVD, Tropp(60, 150) [58]	37.2	100	$2.6 \cdot 10^{-1}$	0.47	0.88	$1.3 \cdot 10^{-2}$
NTTSVD, Tropp(50, 150) [58]	33.6	100	$2.6 \cdot 10^{-1}$	0.47	0.88	$1.2 \cdot 10^{-2}$
NTTF	17.4	2000	$2.1 \cdot 10^{-1}$	0.54	0.92	0
STHOSVD	3.9	100	$1.8 \cdot 10^{-1}$	0.64	0.94	2.1
NSTHOSVD, $SVD_r$ [58]	407.6	100	$1.8 \cdot 10^{-1}$	0.60	0.94	$2.0 \cdot 10^{-2}$
NSTHOSVD, HMT(1, 75) [58]	108.7	100	$1.9 \cdot 10^{-1}$	0.60	0.94	$2.1 \cdot 10^{-2}$
NSTHOSVD, HMT(0, 75) [58]	88.6	100	$2.1 \cdot 10^{-1}$	0.53	0.92	$2.6 \cdot 10^{-2}$
NSTHOSVD, Tropp(60, 150) [58]	80.7	100	$2.3 \cdot 10^{-1}$	0.50	0.9	$2.7 \cdot 10^{-2}$
NSTHOSVD, Tropp(50, 150) [58]	77.2	100	$2.5 \cdot 10^{-1}$	0.47	0.89	$2.7 \cdot 10^{-2}$
TTSVD + APG, ос- нованный на ТТ	18.2	700	$2.1 \cdot 10^{-1}$	0.52	0.92	0

Из наших экспериментов мы видим, что последний подход более выгоден по нескольким причинам. Методология чередующихся проекций стро-

ит аппроксимацию с неотрицательными элементами, но нет требования, чтобы коэффициенты факторизации были неотрицательными. Несмотря на эти смягченные ограничения, второй подход позволяет получить полностью неотрицательную декомпозицию низкого ранга намного быстрее с сопоставимой точностью. Более того, сложность каждой итерации по чередующимся проекциям равна  $O(n^d)$  и линейно масштабируется в зависимости от общего числа элементов тензора. Следовательно, алгоритм обычно приходится останавливать после нескольких итераций, если сходимость занимает слишком много времени. Это означает, что у нас все еще могут быть отрицательные элементы в аппроксимации, тогда как во втором подходе такой проблемы нет. В случае большого тензора Гильберта с  $10^9$  элементов мы получаем ускорение вычислений более чем в 50 раз для формата Таккера и более чем в 100 раз для тензорного поезда. В целом, наиболее важным преимуществом нового подхода является возможность выполнять все операции в малоранговом формате, позволяющем работать с действительно большими данными. Это свойство значительно расширяет спектр возможных применений.

#### 4.5. Вычисление меры влияния узлов с помощью НТТ, искусственные данные

Были рассмотрены несколько примеров, подтверждающих ускорение вычисления собственных векторов влияния при использовании неотрицательной тензорной факторизации вместо исходного тензора. Прежде всего были построены тензоры 4-го порядка, затем использовался алгоритм для построения неотрицательного ТТ в качестве аппроксимации. После этого для каждого измерения (моды) вычислялся собственный вектор влияния. Чтобы проверить разницу между результатами, также вычислялся собственный вектор влияния с исходным тензором, затем оценивалось сходство пересечений между векторами и точечные графики.

Сходство пересечений — это мера, используемая для сравнения первых элементов  $K$  двух ранжированных списков, которые могут содержать разные элементы. Он определяется следующим образом: пусть  $\mathcal{L}^1$  и  $\mathcal{L}^2$  —

Являются ли данные зашумленными	Размер	Относительная ошибка	Время (с)	Средние ранги
Нет	$100 \times 100 \times 100 \times 100$	$3.36 * 10^{-14}$	0.99	3, 3, 3
Да	$100 \times 100 \times 100 \times 100$	0.22	1.63	3, 3, 3

Таблица 17. Неотрицательная ТТ-аппроксимация положительных тензоров рангов 3, 3, 3.

два ранжированных списка, и назовем  $\mathcal{L}_k^j$  списком вершин  $k$  элементов, перечисленных в  $\mathcal{L}^j$ , для  $j = 1, 2$ . Тогда сходство пересечения первых  $K$  элементов между  $\mathcal{L}^1$  и  $\mathcal{L}^2$  определяется как

$$\text{isim}_K(\mathcal{L}^1, \mathcal{L}^2) = \frac{1}{K} \sum_{k=1}^K \frac{|\mathcal{L}_k^1 \Delta \mathcal{L}_k^2|}{2k}$$

где  $|S|$  обозначает мощность множества  $S$ , а  $\mathcal{L}_k^1 \Delta \mathcal{L}_k^2$  — симметрическую разность между  $\mathcal{L}_k^1$  и  $\mathcal{L}_k^2$ . Когда упорядоченные последовательности, содержащиеся в  $\mathcal{L}^1$  и  $\mathcal{L}^2$ , совершенно различны, то сходство пересечений между ними максимально и равно 1. С другой стороны, сходство пересечений между двумя векторами равно 0 тогда и только тогда, когда две упорядоченные последовательности совпадают.

Для всех представленных далее экспериментов относительная точность рассчитывается как  $\frac{\|A-B\|_F}{\|A\|_F} \times 100\%$ , где  $B$  является построенной неотрицательной ТТ-аппроксимацией исходного неотрицательного тензора  $A$ .

Для этого эксперимента случайные последовательности тензоров с положительными элементами были сгенерированы с помощью функции из набора инструментов ТТ [28]. Были рассмотрены  $100 \times 100 \times 100 \times 100$  тензоры с рангами 3, 3, 3. Алгоритм AMEN-CROSS был применен к полным формам тензора перехода для расчета разложения тензорного произведения низкого ранга. Затем проверялось наличие отрицательных элементов, если таковые были, они исправлялись, как это обсуждалось в 2.6. Дополнительно проводились эксперименты с зашумленными данными, когда каждое значение тензора возмущалось некоторым равномерно распределенным случайным числом в интервале  $(0, 1)$ . Усредненные результаты экспериментов за 10 запусков представлены в таблице 17.

После этого вычислялась собственные векторы влияния для каж-

дого измерения (моды) с исходным тензором перехода и его неотрицательным ТТ. Для исходного тензора перехода вычисления потребовали 1,76 секунды, тогда как вычисления с неотрицательным ТТ заняли всего 0,01 секунды. Таким образом, для данных с шумом и без него получено сокращение времени, необходимого для оценки собственных векторов влиятельности.

На рисунках 11 и 12 показаны примеры результатов. Точечный график показывает, что собственные векторы влиятельности, вычисленные с помощью исходного тензора перехода и его неотрицательного ТТ, практически совпадают, поскольку точки образуют линию. И мера сходства пересечений подтверждает это, поскольку на рисунке 12 видно, что более чем 30 первых индексов для каждого измерения имеют одинаковый рейтинг при вычислении с исходным тензором перехода и его неотрицательным ТТ, а для остальных мера сходства не превышает  $1,2 \cdot 10^{-3}$ , при этом ноль означает полное совпадение.

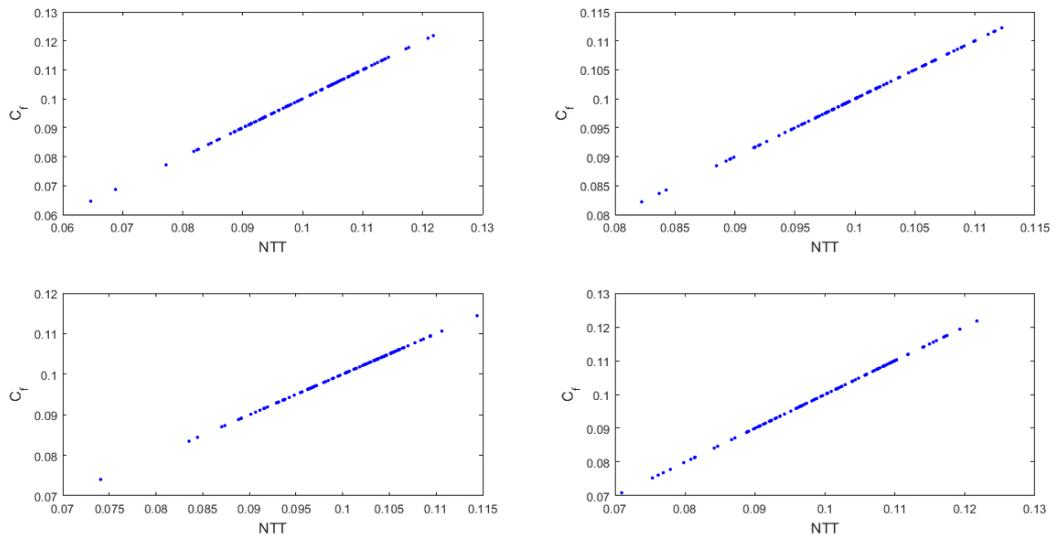


Рис. 11. Сгенерированные зашумленные данные: точечные графики между различными векторами влиятельности. Для собственного вектора влиятельности был использован параметр  $\alpha = 3.2$ . Четыре измерения слева направо, сверху вниз.

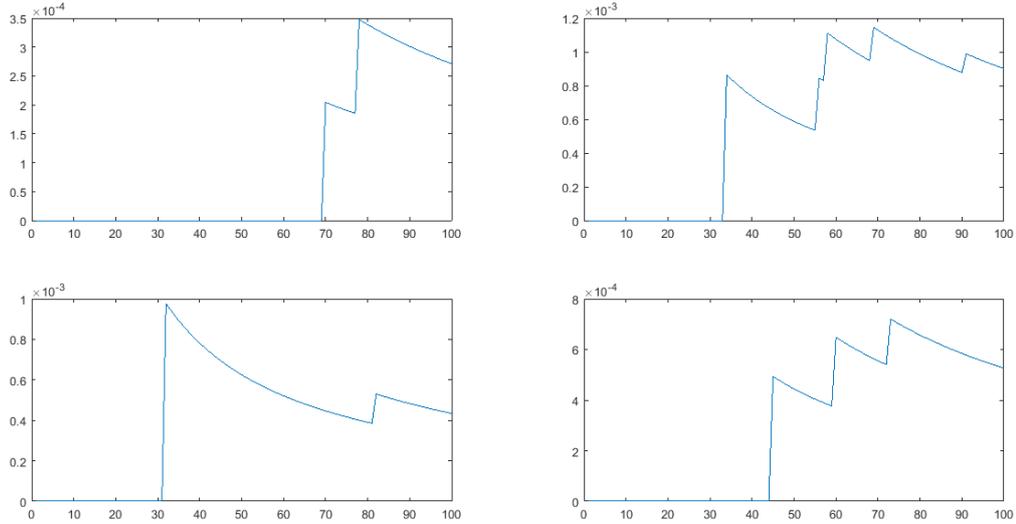


Рис. 12. Сгенерированные зашумленные данные: график пересечения меры схожести между ранжированием из собственного вектора влияния для исходного тензора и его NTT для  $K = 1, 2, \dots, n$ . При вычислении был использован параметр  $\alpha = 3.2$ . Четыре измерения слева направо, сверху вниз.

#### 4.6. Вычисление меры влияния узлов с помощью NTT на примере транспортной модели

Был построен набор данных с идеей транспортировки продуктов между городами. Величина потребления всегда зависит от множества факторов, например, логично иметь линейную зависимость от цены. Часто временные ряды также могут иметь сезонную или даже еженедельную неравномерность. Принимая все это во внимание, функция для элемента набора данных определяется как:

$$f(i, j, k, l) = \frac{1}{i + j - 1} kc(l),$$

где  $i$  и  $j$  соответствуют городу/региону,  $k$  и  $l$  соответствуют цене и дню недели соответственно. Имеется семь значений  $c(l), l = 1, \dots, 7$ , определяющих неравномерность дня недели. Мы можем считать, что города отсортированы по численности населения в порядке убывания, поэтому самый маленький город ( $i = n$ ) может потреблять/производить наименьшее ко-

личество. Таким образом, мы получаем тензор четвертого порядка, для эксперимента рассматривался размер  $100 \times 100 \times 1000 \times 7$ .

Алгоритм AMEN-CROSS был применен к полной форме тензора перехода для расчета разложения тензорного произведения низкого ранга. Затем проверялось наличие отрицательных элементов, если таковые были, то они корректировались, как это обсуждалось в 2.6. Эксперимент был повторен 10 раз, в среднем расчет неотрицательной ТТ-аппроксимации занимал 1,5 секунды с относительной ошибкой 1,02% и рангами 6, 1, 1. После этого были вычислены собственные векторы влияния для каждого измерения с исходным тензором перехода и его неотрицательным ТТ. Для исходного тензора перехода процесс занял 1,6 секунды, тогда как расчеты с неотрицательным ТТ заняли всего 0,01 секунды. Таким образом, вновь было получено сокращение времени, необходимого для оценки собственных векторов влияния.

Ранги собственных векторов влияния для исходного тензора перехода и его НТТ в этом случае совпадают, поэтому для всех  $K = 1, \dots, n_i$ ,  $i = 1, \dots, 4$  мера сходства пересечений равна нулю. Значения обоих собственных векторов влияния также совпадают, как показано на рисунке 13.

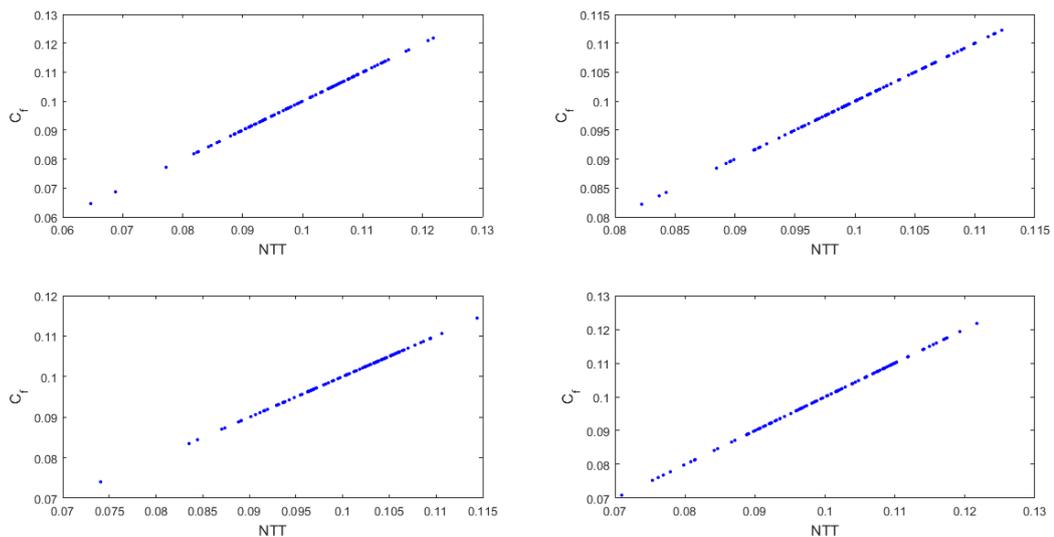


Рис. 13. Набор транспортных данных: точечные графики между различными векторами влияния. Четыре измерения слева направо, сверху вниз.

Если попытаться вычислить значения собственных векторов влияния для тензора перехода размером  $400 \times 400 \times 1000 \times 7$ , нам не удастся сделать это легко, поскольку размер массива превышает предел по умолчанию для MATLAB. Но для метода AMEN-CROSS нам не нужно хранить все элементы в памяти, только часть из них используется для вычисления разложения. И проверка наименьшего элемента дает нам положительное число, а это означает, что нам не нужно вносить какие-либо поправки в разложение. Таким образом, этот процесс требует всего 2,1 секунды. Вычисление собственных векторов влияния в форме тензорного произведения занимает всего 0,2 секунды, поэтому за 2,3 секунды был получен результат, которого невозможно достичь с полной формой исходного тензора перехода.

## Заключение

Данная диссертация посвящена неотрицательной факторизации матриц и тензоров с условием неотрицательности.

Разработаны эффективные методы для неотрицательной факторизации матриц и тензоров на основе малоранговых аппроксимаций и быстрых алгоритмов линейной алгебры, проведено теоретическое исследование предложенных методов, их реализация в виде программ и применение для решения прикладных задач. Предложенный двухэтапный подход для факторизации неотрицательных матриц и тензоров позволяет строить быстрые алгоритмы для получения неотрицательных разложений, сложность итераций которых линейно зависит от размерности исходного объекта, что делает их предпочтительными при работе с большими данными. В качестве первого шага с целью дополнительного сокращения числа операций предлагается использовать крестовые методы. Для всех методов выведены оценки сложности вычислений.

Кроме того, получены следующие частные результаты:

- исследована применимость редуцированного двухэтапного алгоритма к возмущенным сепарабельным матрицам, выведена оценка зависимости

точности приближения от величины возмущения, численные эксперименты показали, что на практике отличие в точности является очень незначительным;

- рассмотрена комбинация одного из алгоритмов построения неотрицательного тензорного произведения с техникой DMRG, что позволяет подбирать также ранги аппроксимации, которые в большинстве алгоритмов считаются входными данными;
- предложен подход к построению неотрицательного ТТ-приближения (без требования неотрицательности элементов ядер), основанный на точечной коррекции элементов, данный метод успешно применяется для математического моделирования процессов коагуляции при неупругих соударениях огромного числа частиц при решении мультикомпонентного уравнения Смолуховского с сохранением неотрицательности;
- проведено сравнение двухэтапного подхода к неотрицательным тензорным факторизациям с подходом чередующихся проекций, численные эксперименты продемонстрировали, что первый, в отличие от второго, позволяет получать при сравнимой точности строго неотрицательные разложения за гораздо меньшее время;
- описано применение модели неотрицательного ТТ при ранжировании узлов многомерного графа, при этом выводится оценка погрешности при использовании для расчетов аппроксимации вместо исходного тензора.

В качестве приложений были рассмотрены гиперспектральные изображения, тензор ERP (event related potential – электрофизиологический отклик на стимул), ядра уравнения Смолуховского и ранжирование вершин многомерного графа. На данный момент алгоритмы, полученные в диссертации, успешно применяются для вычисления неотрицательного решения уравнения Смолуховского [73], [71], сжатия видео и изображений с запуском на нескольких процессорах [72]. В случае решения мультикомпонентного уравнения Смолуховского неотрицательные сдвиги применяются

к численной аппроксимации решения в формате тензорного ядра. Уже известно, что предложенная методика может быть полезна как в режиме регулярных поправок при интегрировании по времени, так и при постобработке стационарных решений. Предлагаемые методы позволяют решать задачи с решением, содержащим  $400^4$  элементов в случае баллистического ядра коагуляции и  $3200^4$  в случае постоянного ядра коагуляции.

Полученные теоретические результаты и разработанный комплекс программ позволяют качественно расширить круг задач, доступных для детального изучения методами математического моделирования.

## Список литературы

- [1] Paatero P. , Tapper U. Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values // *Environmetrics*. – 1994. – V. 5. – P. 111–126.
- [2] Lee D., Seung H. Learning the Parts of Objects by Nonnegative Matrix Factorization // *Nature*. – 1999. – V. 401. – P. 788-791.
- [3] Vavasis Stephen A. On the complexity of nonnegative matrix factorization // *SIAM J. Optim.* – 2008. – V. 20, № 3. – P. 1364–1377.
- [4] Oseledets Ivan Tensor-Train Decomposition // *SIAM J. Scientific Computing*. – 2011. – V. 33. – P. 2295-2317.
- [5] Lee N., Phan A., Cong F., Cichocki A. Nonnegative Tensor Train Decompositions for Multi-domain Feature Extraction and Clustering // *LNCS*. – 2016. – V. 9949. – P. 87-95.
- [6] Tudisco F., Arrigo F., Gautier A. Node and Layer Eigenvector Centralities for Multiplex Networks // *SIAM Journal on Applied Mathematics*. – 2018. – V. 78, № 2. – 853-876.
- [7] Goreinov S.A., Tyrtysnikov E.E., Zamarashkin N.L. A theory of pseudo-skeleton approximations // *Linear Algebra Appl.* – 1997. – V. 261. – P. 1–21.
- [8] Tyrtysnikov E.E. Incomplete cross approximation in the mosaic-skeleton method // *Computing*. – 2000. – V.64, №4. – P. 367-380.
- [9] Goreinov S.A., Tyrtysnikov E.E. The maximal-volume concept in approximation by low-rank matrices // *Contemporary Mathematics*. – 2001. – V. 208. – P. 47-51.
- [10] Goreinov S., Oseledets I., Savostyanov D., Tyrtysnikov E., Zamarashkin N. How to find a good submatrix // *Matrix Methods: Theory, Algorithms and Applications*. – 2010. – P. 247-256.

- [11] Donoho D., Stodden V. When does non-negative matrix factorization give a correct decomposition into parts? // INIPS. – 2003. – P. 1141-1148.
- [12] Arora S., Ge R., Kannan R., Moitra A. Computing a Nonnegative Matrix Factorization – Provably // SIAM Journal on Computing. – 2016. – V. 45, № 4. – P. 1582-1611.
- [13] Esser E., Moller M., Osher S., Sapiro G., Xin J. A convex model for nonnegative matrix factorization and dimensionality reduction on physical space // IEEE Transactions on Image Processing. – 2012. – V. 21, № 7. – P. 3239–3252.
- [14] Bittorf V., Recht B., Ré C., Tropp J. Factoring nonnegative matrices with linear programs // NIPS 2012. – 2012. – P. 1223–1231.
- [15] Gillis N., Vavasis S.A. Fast and Robust Recursive Algorithms for Separable Nonnegative Matrix Factorization // IEEE Trans. on Pattern Analysis and Machine Intelligence. – 2014. – V. 36. – P. 698-714.
- [16] Ding C., He X., Simon H.D. On the equivalence of nonnegative matrix factorization and spectral clustering // Proceedings of the SIAM International Conference on Data Mining (SDM'05). – 2005. – P. 606–610.
- [17] Vandendorpe A., Ho H.N., Vanduffel S., Dooren P.V. On the Parameterization of the Creditrisk-Plus Model for Estimating Credit Portfolio Risk // Insurance: Mathematics and Economics. – 2008. – V. 42, № 2. – P. 736-745.
- [18] Kuang D., Park H. Fast rank-2 nonnegative matrix factorization for hierarchical document clustering // The 19th ACM SIGKDD International Conference on Knowledge, Discovery, and Data Mining (KDD '13). – 2013. – P. 739-747.
- [19] Kuang D. Hierarchical rank-2 NMF for document clustering and topic modeling with visualization [Source code], <https://github.com/dakuang/hiernmf2>.

- [20] Kalofolias V., Gallopoulos E. Computing symmetric nonnegative rank factorizations // *Linear Algebra Appl.* – 2012. – V. 436, № 2. – P. 421–435.
- [21] Cohen J., Rothblum U. Nonnegative ranks, decompositions and factorizations of nonnegative matrices // *Linear Algebra and its Applications.* – 1993. – V. 190. – P. 149–168.
- [22] Gillis N. Sparse and unique nonnegative matrix factorization through data preprocessing // *Journal of Machine Learning Research.* – 2012. – V. 13. – P. 3349-3386.
- [23] Bakula M.K. Jensen–Steffensen inequality for strongly convex functions // *M. J Inequal Appl* 2018. – 2018. – V. 306. – P. 1-12.
- [24] Тыртышников Е.Е. Методы численного анализа. // М.: Издательский центр “Академия”. – 2007.
- [25] 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.
- [26] Zamarashkin N.L., Osinsky A.I. New Accuracy Estimates for Pseudoskeleton Approximations of Matrices // *Doklady Mathematics.* – 2016. – V. 94, № 3. – P. 643-645.
- [27] Osinsky A., Zamarashkin N. Pseudo-skeleton approximations with better accuracy estimates // *Linear Algebra and its Applications.* – 2018. – V. 537. – P. 221-249.
- [28] Oseledets I. et al. TT-Toolbox 2.2 // Institute of Numerical Mathematics, Moscow, Russia. – 2009-2013. <https://github.com/oseledets/TT-Toolbox>.
- [29] Gillis N. Fast and robust recursive algorithm for separable NMF [Source code], <https://sites.google.com/site/nicolasgillis/code>.
- [30] Oseledets I.V. Compact matrix form of the d-dimensional decomposition // Preprint 2009-01, INM RAS. – March 2009.

- [31] Oseledets I.V., Savostianov D.V., Tyrtysnikov E.E. Tucker dimensionality reduction of three-dimensional arrays in linear time // *SIAM J. Matrix Anal. Appl.* – 2008. – V. 30, № 3. – P. 939–956.
- [32] Goreinov S.A. On cross approximation of multi-index array // *Doklady Math.* – 2008. – V. 420, № 4. – P. 1–3.
- [33] Oseledets I. Tensor-Train Decomposition // *SIAM Journal on Scientific Computing.* – 2011. – V. 33, № 5. – P. 2295-2317.
- [34] Oseledets I., Tyrtysnikov E. TT-cross approximation for multidimensional arrays // *Linear Algebra and its Applications.* – 2010. – Volume 432, № 1. – P. 70-88.
- [35] Lee D., Seung H. Algorithms for Non-negative Matrix Factorization // *Adv. Neural Inform. Process. Syst.* – 2001. – V. 13.
- [36] Zhou G., Cichocki A., Xie S. Fast Nonnegative Matrix/Tensor Factorization Based on Low-Rank Approximation // *IEEE Transactions on Signal Processing.* – 2012. – V. 60, № 6. – P. 2928-2940.
- [37] TDALAB – MATLAB Toolbox for High-order Tensor Data Decompositions and Analysis, <https://github.com/andrewssobral/TDALAB>. Current version 1.1, released May 1st, 2013.
- [38] Cong F., Lin Q.-H., Kuang L.-D., Gong X.-F., Astikainen P., Ristaniemi T. Tensor decomposition of EEG signals: a brief review // *J. Neurosci. Methods.* – 2015. – V. 248. - P. 59-69.
- [39] Cong F., Phan A., Zhao Q., Huttunen-Scott T., Kaartinen J., Ristaniemi T., Lyytinen H., Cichocki A. Benefits of Multi-domain Feature of Mismatch Negativity Extracted by Nonnegative Tensor Factorization from Low-density Array EEG // *International Journal of Neural Systems.* – 2012. – V. 22, № 6.
- [40] Savostyanov D., Oseledets I. Fast adaptive interpolation of multi-dimensional arrays in tensor train format // 2011 7th International Workshop on Multidimensional (nD) Systems, nDS. – 2011. – P. 1-8.

- [41] Lin C.-J. Projected Gradient Methods for NonNegative Matrix Factorization // *Neural computation*. – 2007. – V. 19, № 10. – P. 2756-2779.
- [42] Zdunek R., Cichocki A. Non-negative Matrix Factorization with Quasi-Newton Optimization // *Lecture Notes in Computer Science*. – 2006. – V. 4029. – P. 870-879.
- [43] Cichocki A., Zdunek R., Amari S. Hierarchical ALS Algorithms for Nonnegative Matrix and 3D Tensor Factorization // *LNCS*. – 2007. – V. 4666. – P. 169-176.
- [44] Berry M., Browne M., Langville A., Pauca P., Plemmons R. Algorithms and Applications for Approximate Nonnegative Matrix Factorization // *Computational Statistics & Data Analysis*. – 2007. – V. 52. – P. 155-173.
- [45] Matveev S., Zheltkov D., Tyrtysnikov E., Smirnov A. Tensor train versus Monte Carlo for the multicomponent Smoluchowski coagulation equation // *Journal of Computational Physics*. – 2016. – V. 316. – P. 164-179.
- [46] Hitchcock F.L. Multiple invariants and generalized rank of a p-way matrix or tensor // *Journal of Mathematics and Physics*. – 1927. – V. 7. – P. 39–79.
- [47] Harshman R. A. Determination and proof of minimum uniqueness conditions for PARAFAC1 // *UCLA Working Papers in Phonetics*. – 1972. – V. 22.
- [48] Carroll J.D., Chang J.J. Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart–Young decomposition // *Psychometrika*. – 1970. – V. 35, № 3. – P. 283–319.
- [49] Kruskal J.B. Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics // *Linear Algebra Appl*. – 1977. – V. 18. – P. 95–138.

- [50] Carroll J.D., G. De Soete, Pruzansky S. Fitting of the latent class model via iteratively reweighted least squares CANDECOP with nonnegativity constraints. // *Multiway data analysis*. – 1989. – P. 463–472.
- [51] Phan A. -H., Cichocki A., Oseledets I., Calvi G. G., Ahmadi-Asl S., Mandic D. P. Tensor Networks for Latent Variable Analysis: Higher Order Canonical Polyadic Decomposition // *IEEE Transactions on Neural Networks and Learning Systems*. – 2020. – V. 31, № 6. – P. 2174-2188.
- [52] Lebedeva O. Tensor conjugate-gradient-type method for rayleigh quotient minimization in block qtt-format // *Russian journal of numerical analysis and mathematical modelling*. – 2011. – V. 26, № 5. – P. 465–489.
- [53] Рахуба М. В. Тензорные методы решения многомерных частичных задач на собственные значения: дис. кандидата физико-математических наук: 01.01.07 - МФТИ, Москва, 2017 - 167 с.
- [54] Kuksova S., Skau E. W., Alexandrov B. S. Error Analysis of Nonnegative Tensor Train Utilized for Nonnegative Canonical Polyadic Decomposition // *ICML*. – 2020.
- [55] Cichocki A., Phan A.-H. Fast Local Algorithms for Large Scale Nonnegative Matrix and Tensor Factorizations // *IEICE Transactions*. – 2009. – V. 92-A. – P. 708-721.
- [56] Phan A.-H., Tichavsky P., Cichocki A. TENSORBOX: a MATLAB package for tensor decomposition. // <https://github.com/phananh Huy/TensorBox> – 2019.
- [57] Xu Y. Alternating proximal gradient method for sparse nonnegative Tucker decomposition // *Mathematical Programming Computation*. – 2015. – V. 7, № 1. – P. 39–70.
- [58] Sulstonov A., Matveev S., Budzinskiy S. Low-rank nonnegative tensor approximation via alternating projections and sketching // *Computational and Applied Mathematics*. – 2023. – V. 42, № 68.

- [59] Zhou G., Cichocki A., Zhao Q., Xie S. Efficient Nonnegative Tucker Decompositions: Algorithms and Uniqueness // IEEE Transactions on Image Processing. – 2015. – V. 24, № 12. – P. 4990-5003.
- [60] Guangjing S., Ng M. Nonnegative Low Rank Matrix Approximation for Nonnegative Matrices. // Applied Mathematics Letters. – 2020. – V. 105. – P. 106300.
- [61] Jiang T.-X., Ng M., Pan J., Song G. 2023. Nonnegative low rank tensor approximations with multidimensional image applications // Numerische Mathematik. – 2023. – V. 153, № 1. – P. 141-170.
- [62] Matveev S., Budzinskiy S. Sketching for a low-rank nonnegative matrix approximation: Numerical study. // Russian Journal of Numerical Analysis and Mathematical Modelling. – 2023. – V. 38, № 2. – P. 99-114.
- [63] Dolgov S., Savostyanov D. Alternating minimal energy methods for linear systems in higher dimensions // SIAM Journal on Scientific Computing. – 2014. – V. 36, № 5 – P. A2248-A2271.
- [64] Page L., Brin S., Motwani R., Winograd T. The PageRank Citation Ranking: Bringing Order to the Web // Stanford InfoLab – 1999.
- [65] Gleich D., Lim, L.-H., Yu Y. Multilinear PageRank // SIAM Journal on Matrix Analysis and Applications. – 2015. – V. 36, № 4. – P. 1507-1541.
- [66] Cipolla S., Redivo-Zaglia M., Tudisco F. Extrapolation Methods for fixed-point Multilinear PageRank computations // arXiv. – 2019.
- [67] Savostyanov D., Oseledets I. Fast adaptive interpolation of multidimensional arrays in tensor train format // The 2011 International Workshop on Multidimensional (nD) Systems. – 2011. – P. 1-8.
- [68] Gautier A., Tudisco F., Hein M. The Perron-Frobenius Theorem for Multihomogeneous Maps // 10.48550/arXiv.1702.03230. – 2017.

- [69] Wang Z., Bovik A., Sheikh H., Simoncelli E. Image quality assessment: from error visibility to structural similarity // IEEE transactions on image processing. – 2004. – V. 13, № 4. – P. 600-612.
- [70] Yuan Q., Zhang L., Shen H. Hyperspectral image denoising employing a spectral-spatial adaptive total variation model // IEEE Transactions on Geoscience and Remote Sensing. – 2012. – V. 50, № 10. – P. 3660-3677.
- [71] Matveev S., Tretyak I. Nonnegative tensor train for the multicomponent Smoluchowski equation // <https://arxiv.org/abs/2404.10898> – 2024.
- [72] Bhattarai M., Chennupati G., Skau E., Vangara R., Djidjev H., Alexandrov, B. Distributed Non-Negative Tensor Train Decomposition // arXiv. – 2020. – P. 1-10.
- [73] Manzini G., Skau E., Truong D., Vangara R. Nonnegative Tensor-Train Low-Rank Approximations of the Smoluchowski Coagulation Equation // LNCS. – 2022. – V. 13127. – P. 342-350.

## Публикации автора по теме диссертации

Научные статьи, опубликованные в журналах Scopus, WoS, RSCI, а также в изданиях, рекомендованных для защиты в диссертационном совете МГУ по специальности.

- A.1 Shcherbakova E. Nonnegative Tensor Train Factorization with DMRG Technique // Lobachevskii Journal of Mathematics. – 2019. – V. 40, № 11. – P. 1863-1872.
- A.2 Тыртышников Е.Е., Щербакова Е.М. Методы неотрицательной матричной факторизации на основе крестовых малоранговых приближений // Журнал вычислительной математики и математической физики. – 2019. – Т. 59, № 8. – С. 1314-1330.
- A.3 Shcherbakova E., Tyrtysnikov E. Nonnegative Tensor Train Factorizations and Some Applications // Lecture Notes in Computer Science. – 2020. – V. 11958 – P. 156-164.

- A.4 Shcherbakova E., Tyrtysnikov E. Fast Nonnegative Tensor Factorizations with Tensor Train Model // Lobachevskii Journal of Mathematics. – 2022. – V. 43, № 4. – P. 882–894.
- A.5 Shcherbakova Elena M., Matveev Sergey A., Smirnov Alexander P., Tyrtysnikov Eugene E. Study of performance of low-rank nonnegative tensor factorization methods // Russian Journal of Numerical Analysis and Mathematical Modelling. – 2023. – V. 38, № 4. – P. 231-239.