

# Тензорный поезд в марковском случайном поле<sup>1</sup>

А. В. Новиков<sup>\*</sup>, А. О. Родоманов<sup>\*</sup>, А. А. Осокин<sup>\*</sup>,  
Д. П. Ветров<sup>\*†</sup>

*(\* 119992 Москва, Ленинские горы, 1, МГУ ф-т ВМК;  
† 125319 Москва, Кочновский проезд, 4., НИУ ВШЭ ф-т ФКН)*  
*e-mail: novikov@bayesgroup.ru; anton.rodomanov@gmail.com; osokin@bayesgroup.ru;  
vetrovd@yandex.ru*

## Аннотация

В этой статье предлагается новый подход для работы с вероятностными графическими моделями, основанный на недавно предложенном разложении тензорного поезда (Tensor Train, ТТ). ТТ-разложение позволяет хранить тензор в компактном виде и эффективно применять к нему операции линейной алгебры. В данной работе предложен метод для перевода тензоров энергии и ненормированной вероятности марковского случайного поля в ТТ-формат. Свойства этого формата используются для подсчета нормировочной константы и поиска конфигурации наибольшей вероятности. Предоставлены теоретические гарантии точности оценки нормировочной константы. Предложенные методы сравниваются с аналогами на различных наборах данных.

## 1 Введение

Вероятностные графические модели являются удобным инструментом для решения задач в таких областях, как компьютерное зрение, машинное обучение, анализ социальных сетей и т. д. Одно из их основных достоинств — это возможность компактно задать ненормированное распределение на сотни переменных в виде произведения факторов низких порядков. Это позволяет, с одной стороны, учесть сложные зависимости в данных, а с другой стороны точно или приближенно вычислять статистики полученного распределения.

---

<sup>1</sup>Публикация базируется на работе, финансируемой Сколковским институтом науки и технологий (Сколтех): Соглашение о совместной лаборатории № 081-R от «01» октября 2013 года, приложение А2; а так же при финансовой поддержке РФФИ (коды проектов 12 01 00938, 12 01 33085).

Многие возникающие задачи становятся сложными, если фактор граф содержит циклы. В этой статье рассматриваются две актуальные для графических моделей задачи: оценка нормировочной константы распределения и поиск наиболее вероятной конфигурации (минимизация энергии). Решение этих задач особенно важно при настройке параметров графической модели по тренировочной выборке. Лучшие алгоритмы обучения требуют либо быстрого решения задачи минимизации энергии (как структурные варианты SVM [19, 20]), либо хороших оценок нормировочной константы (обучение по методу максимального правдоподобия [23]). В последние годы для этих задач предложено множество различных методов, но, тем не менее, их пока рано называть полностью решенными.

Совместное распределение дискретных переменных можно задать через многомерный массив (тензор) ненормированной вероятности. Тогда поиск конфигурации наибольшей вероятности и подсчет нормировочной константы сводятся к поиску наибольшего элемента тензора и суммированию всех элементов тензора соответственно. Для хранения и обработки тензора в явном виде требуется экспоненциальное количество памяти и вычислительных ресурсов. Тензорные разложения позволяют компактно хранить тензор и эффективно совершать над ним различные операции.

В этой статье рассматриваются недавно предложенное И. Оселедцом [17] разложение тензорного произведения (ТТ-разложение или ТТ-формат). Алгоритмы ТТ-разложения хранят тензор в специальном формате, который позволяет эффективно оперировать над тензорами. Эффективность ТТ-разложения определяется величиной ТТ-рангов тензора. В работе показано, что тензор энергии (минус логарифма вероятности) марковского случайного поля обладает низкими ТТ-рангами, а тензор вероятности — высокими ТТ-рангами. Представление тензора энергии в ТТ-формате позволяет решать задачу минимизации энергии с помощью методов линейной алгебры. Для подсчета нормировочной константы в условиях высоких ТТ-рангов тензора вероятности предложен метод основанный на факторизации графической модели. Факторы графической модели переводятся в ТТ-формат и комбинируются таким образом, чтобы оценить нормировочную константу, не строя сам тензор вероятности. Этот подход также позволяет выполнять подсчет маргинальных распределений.

Предложенные методы значительно опережают аналоги на задачах подсчета нормировочной константы и маргинальных распределений и показывают сравнимые результаты на задаче минимизации энергии.

Основной вклад настоящей работы заключается в следующем:

- Предложен алгоритм для представления тензора энергии (минус логарифма вероятности) в ТТ-формате. Показано, что тензор энергии графической модели низкого порядка обладает низкими ТТ-рангами и доказана оценка сверху на его ТТ-ранги.
- Предложен алгоритм оценки нормировочной константы через ТТ-разложение факторов графической модели, который не строит ТТ-представление тензора ненормированного совместного распределения.

- Приведены теоретические гарантии на точность оценки нормировочной константы.

Настоящая статья организована следующим образом. В разделе 2 проведен обзор литературы, а в разделе 3 введены обозначения. В разделе 4 кратко освещено разложение тензорного произведения, а в разделе 5 обсуждена его связь с марковскими случайными полями. В разделе 6 предложен алгоритм для подсчета нормировочной константы и проведен его анализ. Раздел 7 содержит результаты экспериментов.

## 2 Обзор литературы

Обзор можно разделить на три основных направления: тензорные разложения, позволяющие компактно представить тензор, использование тензорных методов в графических моделях и различные алгоритмы вывода.

В отличие от классических тензорных форматов (канонический формат [1] и формат Таккера [21]), для ТТ-разложения существуют устойчивые алгоритмы построения, а само ТТ-разложение не подвержено проклятию размерности. Альтернативой ему является иерархический формат Таккера [6, 4], который также позволяет устойчиво искать компактное представление тензора. ТТ-формат эквивалентен иерархическому формату Таккера с линейным деревом размерностей, что является преимуществом с алгоритмической точки зрения: большинство алгоритмов ТТ-формата значительно проще аналогов для иерархического формата Таккера.

В статьях по графическим моделям тензорные подходы часто упоминаются применительно к восстановлению структуры модели. Джернит и др.; Мария Иштева и др. принимают локальные решения, основываясь на свойствах четырехмерных тензоров [9, 8]. Сонг и др. используют иерархические тензорные разложения для восстановления параметров графической модели [18].

Существует две основные постановки задачи вывода в графической модели: минимизация энергии и вероятностный вывод. На текущий момент, задача минимизации энергии хорошо изучена. В последних экспериментальных исследованиях Каппес и др. показывают [10], что для многих актуальных задач текущие методы дают достаточно хорошие результаты. Особенно много внимания уделено парно-сепарабельным марковским случайным полям (см. детальный обзор Каппеса и др. [10]). Под вероятностным выводом понимается множество различных задач. В этой работе уделено внимание двум из них: вычислению нормировочной константы по ненормированному распределению марковского случайного поля, и вычислению маргинальных распределений. Вторая проблема является более сложной, так как для её решения зачастую требуется вычисление нормировочной константы.

Можно выделить следующие группы методов подсчета нормировочной константы: методы, основанные на генерации выборки (например Annealed Importance Sampling [16, 5]), методы передачи сообщений (классический Loopy Belief Propagation [13] и его многочисленные модификации), методы,

основанные на минимизации KL-дивергенции (Mean Field [23] и Expectation Propagation [14]), методы декомпозиции графов (Tree-Reweighted Message-passing [22]) и методы, основанные на минимизации энергии (randomized MAP-predictors [7] и недавно предложенный метод WISH [3]). Для решения задачи вычисления маргинальных распределений подходят более простые методы, такие как Gibbs sampling [23].

### 3 Обозначения

В данной статье часто будет использоваться понятие многомерного массива вещественных чисел. Одномерные массивы будем называть *векторами*, двумерные — *матрицами*, а массивы большей размерности будем называть *тензорами*. Для обозначения векторов будут использоваться маленькие жирные буквы ( $\mathbf{a}$ ), для матриц большие буквы ( $A$ ), и для тензоров большие жирные буквы ( $\mathbf{A}$ ).

Будем рассматривать все массивы как функции их индексов:  $\mathbf{a}(i) = a_i$ ,  $A(x_1, x_2)$ ,  $\mathbf{A}(\mathbf{x}) = \mathbf{A}(x_1, \dots, x_n)$ , где  $n$  это размерность тензора  $\mathbf{A}$ .

Символом  $\|\cdot\|_F$  будем обозначать норму Фробениуса матриц и её обобщение на случай тензоров:

$$\|\mathbf{A}\|_F = \sqrt{\sum_{x_1, \dots, x_n} \mathbf{A}^2(x_1, \dots, x_n)}.$$

Под  $\|\cdot\|_2$  будем понимать евклидову норму векторов и спектральную норму матриц:  $\|A\|_2 = \max_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2}$ .

В работе будут использоваться несколько различных матричных произведений: произведение Адамара (символ “ $\odot$ ”), произведение Кронекера (символ “ $\otimes$ ”) и обычное матричное произведение (символ “ $\cdot$ ”).

### 4 ТТ-разложение

Будем говорить, что  $n$ -мерный тензор  $\mathbf{A}$  представлен в ТТ-формате, если для всех размерностей  $i = 1, \dots, n$  и всех значений индексов по этим размерностям  $x_i = 1, \dots, d_i$  ( $d = \max_{i=1, \dots, n} d_i$ ) существуют матрицы  $G_i^{\mathbf{A}}[x_i]$ , такие, что каждый элемент тензора  $\mathbf{A}$  представим в виде произведения матриц:

$$\mathbf{A}(x_1, \dots, x_n) = G_1^{\mathbf{A}}[x_1]G_2^{\mathbf{A}}[x_2] \dots G_n^{\mathbf{A}}[x_n]. \quad (1)$$

При этом все матрицы относящиеся к одной и той же размерности  $i$  должны иметь одинаковые размеры  $r_{i-1}(\mathbf{A}) \times r_i(\mathbf{A})$ . Чтобы результат матричного произведения (1) был равен числу, положим  $r_0(\mathbf{A}) = r_n(\mathbf{A}) = 1$ . Последовательность  $\{r_i(\mathbf{A})\}_{i=0}^n$  будем называть *ТТ-рангами* тензора  $\mathbf{A}$ , а максимальный элемент последовательности — *максимальным ТТ-рангом* тензора  $\mathbf{A}$ :  $r(\mathbf{A}) = \max_{i=0, \dots, n} r_i(\mathbf{A})$ . Набор матриц  $G_i^{\mathbf{A}}$ , соответствующих одному

Таблица 1: Операции, которые можно эффективно выполнять над тензорами в ТТ-формате. Для каждой операции указана её вычислительная сложность и ТТ-ранг результата для ситуаций, когда результат является тензором в ТТ-формате.

ОПЕРАЦИЯ	РАНГ РЕЗУЛЬТАТА	ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ
$\mathbf{C} = \mathbf{A} \cdot \text{const}$	$r(\mathbf{C}) = r(\mathbf{A})$	$O(d r(\mathbf{A}))$
$\mathbf{C} = \mathbf{A} + \text{const}$	$r(\mathbf{C}) = r(\mathbf{A}) + 1$	$O(n d r^2(\mathbf{A}))$
$\mathbf{C} = \mathbf{A} + \mathbf{B}$	$r(\mathbf{C}) \leq r(\mathbf{A}) + r(\mathbf{B})$	$O(n d (r(\mathbf{A}) + r(\mathbf{B}))^2)$
$\mathbf{C} = \mathbf{A} \odot \mathbf{B}$	$r(\mathbf{C}) \leq r(\mathbf{A}) r(\mathbf{B})$	$O(n d r^2(\mathbf{A}) r^2(\mathbf{B}))$
$\mathbf{c} = M \mathbf{b}$	$r(\mathbf{c}) \leq r(M) r(\mathbf{b})$	$O(n d^2 r^2(M) r^2(\mathbf{b}))$
$\text{sum } \mathbf{A}$	–	$O(n d r^2(\mathbf{A}))$
$\ \mathbf{A}\ _F$	–	$O(n d r^3(\mathbf{A}))$
$\mathbf{C} = \text{round}(\mathbf{A}, \varepsilon)$	$r(\mathbf{C}) \leq r(\mathbf{A})$	$O(n d r^3(\mathbf{A}))$

измерению, называется *ТТ-ядром* тензора  $\mathbf{A}$ . Представление тензора в ТТ-формате будем называть *ТТ-разложением* или *ТТ-представлением*.

Для любого  $n$ -мерного тензора  $\mathbf{A}$  существует ТТ-представление с максимальным ТТ-рангом  $r(\mathbf{A}) \leq d^{\frac{n}{2}}$  (см. теорему 2.1 Оселедца [17]). Отметим, что ТТ-представление тензора не единственно.

Для обозначения  $(\alpha_{i-1}, \alpha_i)$ -ого элемента матрицы  $G_i^{\mathbf{A}}[x_i]$  будет использоваться символ  $G_i^{\mathbf{A}}[x_i](\alpha_{i-1}, \alpha_i)$ . Пользуясь определением произведения матриц, можно переписать формулу (1) через элементы ТТ-ядер:

$$\mathbf{A}(\mathbf{x}) = \sum_{\alpha_0, \dots, \alpha_n} G_1^{\mathbf{A}}[x_1](\alpha_0, \alpha_1) \dots G_n^{\mathbf{A}}[x_n](\alpha_{n-1}, \alpha_n). \quad (2)$$

Для хранения всех элементов тензора  $\mathbf{A}$  требуется  $\prod_{i=1}^n d_i$  ячеек памяти, тогда как хранение  $\mathbf{A}$  в ТТ-формате требует  $\sum_{i=1}^n d_i r_{i-1}(\mathbf{A}) r_i(\mathbf{A})$ . Таким образом, ТТ-представление тензора с низкими ТТ-рангами существенно компактнее перечисления его элементов.

Существуют две различные постановки задачи перевода тензора в ТТ-формат: точное ТТ-представление (алгоритм ТТ-SVD [17], применимый для небольших тензоров), и построение приближенного ТТ-представления по небольшому подмножеству элементов тензора. Наилучшим из алгоритмов второго класса в настоящее время является метод AMEn-cross [2].

Одним из основных достоинств ТТ-формата является возможность эффективно применять различные операции к тензорам в ТТ-формате: умножение тензора на константу, добавление константы к тензору, поточечное сложение и умножение тензоров (результат этих операций — это тензор в ТТ-формате с возросшими ТТ-рангами); подсчет глобальных характеристик тензора, таких как сумма всех элементов или норма Фробениуса. Обзор операций, которые используются в этой статье, приведен в таблице 1 (детальный обзор проведен Оселедцом [17]).

Применение операций к ТТ-тензорам увеличивает ТТ-ранги даже в том случае, когда существует низкоранговое ТТ-представление результата. Чтобы контролировать рост ТТ-рангов существует операция ТТ-округления. По ТТ-представлению тензора  $\mathbf{A}$  и относительной точности  $\varepsilon \geq 0$  операция ТТ-округления  $\text{round}(\mathbf{A}, \varepsilon)$  найдет тензор  $\tilde{\mathbf{A}}$  в ТТ-формате который, во-первых, достаточно близок к тензору  $\mathbf{A}$ :  $\|\mathbf{A} - \tilde{\mathbf{A}}\|_F \leq \varepsilon \|\mathbf{A}\|_F$  и, во-вторых, обладает минимальными ТТ-рангами среди всех тензоров  $\mathbf{B}$ :  $\|\mathbf{A} - \mathbf{B}\|_F \leq \frac{\varepsilon}{\sqrt{n-1}} \|\mathbf{A}\|_F$ . Наличие операции ТТ-округления позволяет применять последовательность операций к тензорам (например, округляя результат после применения каждой операции), контролируя рост ТТ-рангов.

Для повышения эффективности работы с векторами и матрицами специальным образом вводятся понятия ТТ-формата вектора и ТТ-формата матрицы. Пусть существует отображение между индексами вектора  $\mathbf{b} \in \mathbb{R}^q$  и  $n$ -мерными векторами  $\mathbf{y} = (y_1, \dots, y_n)^1$ . ТТ-представлением вектора  $\mathbf{b}$  называется ТТ-представление тензора  $\mathbf{B}(y_1, \dots, y_n)$ , содержащего все элементы  $\mathbf{b}$ .

Определим понятие ТТ-формата для матриц. Пусть существует отображение между индексами строк и столбцов матрицы  $M$  в  $n$ -мерные вектора  $\mathbf{x}$  и  $\mathbf{y}$  соответственно. Переупорядочим размерности и представим получившийся тензор в ТТ-формате:

$$\mathbf{M}((x_1, y_1), \dots, (x_n, y_n)) = G_1^{\mathbf{M}}[x_1, y_1] \dots G_n^{\mathbf{M}}[x_n, y_n],$$

где  $G_i$ ,  $i = 1, \dots, n$  — это ТТ-ядра, а  $G_i^{\mathbf{M}}[x_i, y_i]$  — матрицы. ТТ-представлением матрицы  $M$  будем называть ТТ-представление тензора  $\mathbf{M}$ . Отметим, что матрица в ТТ-формате не обязана быть квадратной, т. к.  $x_i$  и  $y_i$  могут принимать разное число возможных значений

Для матрицы  $M$  и вектора  $\mathbf{b}$  представленных в ТТ-формате можно эффективно вычислять произведение  $\mathbf{c} = M\mathbf{b}$  (если соответствующие размерности совпадают). Результатом этой операции является вектор  $\mathbf{c}$  в ТТ-формате с рангами равными произведению рангов  $M$  и  $\mathbf{b}$ :  $r_i(\mathbf{c}) = r_i(M) r_i(\mathbf{b})$ .

Наличие специального определения ТТ-формата для векторов и матриц позволяет применять операции линейной алгебры к задачам большого размера. Например, для поиска минимального элемента тензора можно вытянуть его в диагональную матрицу и применить приближенный метод поиска минимальных собственных значений основанный на алгоритме DMRG [11].

## 5 Тензоры марковского случайного поля

Рассмотрим гиперграф  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  с конечным множеством вершин  $\mathcal{V}$  и гиперребер  $\mathcal{E}$ . Пусть все вершины пронумерованы от 1 до  $n$ , а все гиперребра от 1 до  $m$ .

Сопоставим каждой вершине  $i = 1, \dots, n$  переменную  $x_i$  принимающую значения из множества  $\mathcal{X}_i = \{1, \dots, d_i\}$ . Обозначим множество вер-

<sup>1</sup> Количество элементов в векторе  $\mathbf{b}$  равно  $\prod_{i=1}^n d_i$ .

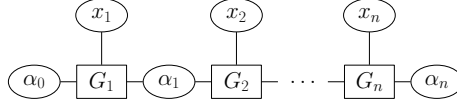


Рис. 1: Графическая модель, полученная в результате ТТ-разложения тензора вероятности  $\mathbf{P}(x_1, \dots, x_n)$ .

шин, входящих в гиперребро  $\ell = 1, \dots, m$ , через  $\mathbf{x}^\ell$ . Каждому гиперребру  $\ell = 1, \dots, m$  сопоставим вещественнозначную функцию  $\Theta_\ell$ , определенную на совместной области определения переменных из  $\mathbf{x}^\ell$ . Функцию  $\Theta_\ell$  будем называть *потенциалом*.

*Функцией энергии* марковского случайного поля (Markov random field, MRF) заданного на гиперграфе  $\mathcal{G}$ , назовем сумму всех потенциалов:  $\mathbf{E}(\mathbf{x}) = \sum_{\ell=1}^m \Theta_\ell(\mathbf{x}^\ell)$ .

Экспонента от минус энергии задает *ненормированное распределение Гиббса*:  $\hat{\mathbf{P}}(\mathbf{x}) = \exp(-\mathbf{E}(\mathbf{x}))$ . Для обозначения нормировочной константы будет использоваться символ  $Z$ ,  $Z = \sum_{\mathbf{x}} \hat{\mathbf{P}}(\mathbf{x})$ . Функции  $\Psi_\ell(\mathbf{x}^\ell) = \exp(-\Theta_\ell(\mathbf{x}^\ell))$  будем называть *факторами* марковского случайного поля.

Как энергия, так и ненормированная вероятность являются  $n$ -мерными тензорами. Потенциалы и факторы можно интерпретировать как  $n$ -мерные тензоры, если добавить в них переменные, от которых они зависят несущественно:  $\Theta_\ell(\mathbf{x}^\ell) = \Theta_\ell(\mathbf{x})$ . Тензоры энергии и вероятности можно определить следующим образом:  $\mathbf{E} = \sum_{\ell=1}^m \Theta_\ell$  и  $\hat{\mathbf{P}} = \odot_{\ell=1}^m \Psi_\ell$ .

Традиционно для компактного представления энергии используется набор потенциалов MRF. ТТ-разложение является альтернативным способом компактного представления тензора энергии. Эффективность применения ТТ-формата для тензора энергии и тензора вероятности обсуждается в разделах 5.1 и 5.2.

ТТ-представление тензора совместного распределения  $\mathbf{P} = \hat{\mathbf{P}}/Z$  обладает особой интерпретацией. Переменные  $\alpha_i$ ,  $i = 0, \dots, n$ , (см. (2)) можно интерпретировать как скрытые переменные в графической модели типа цепочка (рис. 1). Маргинализация по ним дает исходную вероятность:

$$\mathbf{P}(\mathbf{x}) = \sum_{\boldsymbol{\alpha}} \mathbf{P}(\mathbf{x}, \boldsymbol{\alpha}), \quad (3)$$

где  $\mathbf{P}(\mathbf{x}, \boldsymbol{\alpha})$  — совместное распределение вероятностей на векторы  $\mathbf{x}$  и  $\boldsymbol{\alpha}$ . Каждая добавленная переменная  $\alpha_i$  может принимать значения от 1 до соответствующего ТТ-ранга  $r_i(\mathbf{P})$ .

## 5.1 ТТ-формат для тензора энергии

В этом разделе предложен алгоритм для представления тензора энергии  $\mathbf{E}$  в ТТ-формате. Предлагаемый алгоритм существенно превосходит AMEn-cross по скорости и точности.

Алгоритм состоит из трех основных шагов.

Шаг 1. Найдем ТТ-представление для каждого потенциала  $\Theta_\ell(\mathbf{x}^\ell)$ , который будем интерпретировать как тензор зависящий только от переменных  $\mathbf{x}^\ell$ . Обычно потенциалы существенно зависят лишь от небольшого числа переменных и можно искать их ТТ-представление с помощью алгоритма ТТ-SVD.

Шаг 2. Добавим в каждый тензор  $\Theta_\ell$  несущественные переменные  $x \setminus x^\ell$ , сделав его  $n$ -мерным тензором.

Пусть тензор  $\Theta_\ell$  существенно зависит от  $p$  переменных:  $\mathbf{x}^\ell = (x_{i_1}, \dots, x_{i_p})$ ,  $i_1 < i_2 < \dots < i_p$ . После первого шага алгоритма получим следующее ТТ-представление  $\Theta_\ell(\mathbf{x}^\ell)$ :

$$\Theta_\ell(x_{i_1}, \dots, x_{i_p}) = \bar{G}_1^{\Theta_\ell}[x_{i_1}] \dots \bar{G}_p^{\Theta_\ell}[x_{i_p}], \quad (4)$$

где  $\bar{G}_k^{\Theta_\ell}[x_{i_k}]$  — матрица размера  $\bar{r}_{k-1} \times \bar{r}_k$ . Чтобы добавить несущественные размерности, можно, например, положить недостающие ядра равными набору единичных матриц подходящего размера<sup>2</sup>:

$$\begin{aligned} G_1^{\Theta_\ell}[x_1] &\equiv \dots \equiv G_{i_1-1}^{\Theta_\ell}[x_{i_1-1}] \equiv I_{\bar{r}_0} = I_1, \\ G_{i_1+1}^{\Theta_\ell}[x_{i_1+1}] &\equiv \dots \equiv G_{i_2-1}^{\Theta_\ell}[x_{i_2-1}] \equiv I_{\bar{r}_1}, \\ &\dots \\ G_{i_p+1}^{\Theta_\ell}[x_{i_p+1}] &\equiv \dots \equiv G_n^{\Theta_\ell}[x_n] \equiv I_{\bar{r}_p} = I_1, \end{aligned}$$

где  $G_{i_k}^{\Theta_\ell}[x_{i_k}] = \bar{G}_k^{\Theta_\ell}[x_{i_k}]$ . Полученные таким образом ТТ-ядра задают ТТ-формат тензора  $\Theta_\ell(\mathbf{x})$ . Отметим, что в процессе данного преобразования максимальный ТТ-ранг  $\Theta_\ell$  не изменяется.

Шаг 3. Просуммируем полученные на шаге 2 ТТ-тензоры  $\Theta_\ell$ , чтобы получить тензор энергии:

$$\mathbf{E} = \sum_{\ell=1}^m \Theta_\ell. \quad (5)$$

Теорема 1 содержит верхнюю оценку на максимальный ТТ-ранг тензора, построенного описанным алгоритмом.

**Теорема 1.** *Если порядок каждого потенциала не превосходит  $p$ , то предложенный алгоритм строит ТТ-представление тензора энергии  $\mathbf{E}$  с максимальным ТТ-рангом, ограниченным сверху следующим образом:*

$$\text{r}(\mathbf{E}) \leq d^{\frac{p}{2}} \cdot m. \quad (6)$$

Теорема 1 показывает, что можно построить точное низкоранговое ТТ-представление тензора энергии. Отметим, что после шага 3 к тензору энергии можно применить процедуру ТТ-округления, чтобы уменьшить ТТ-ранги, если это возможно.

<sup>2</sup> $I_k$  используется для обозначения единичной матрицы размера  $k \times k$ .



*Доказательство теоремы 1.* Оценим, какими ТТ-рангами обладают факторы после шага 1 алгоритма. Из теоремы 2.1 [17] следует, что ТТ-ранги тензора не превосходят  $d^{\frac{n}{2}}$ , где  $n$  — это размерность тензора, а  $d$  — максимальное число возможных значений, которые может принимать каждая переменная  $x_i$ . По условию теоремы, порядок каждого потенциала не превосходит  $p$ , а значит после шага 1 верна следующая оценка:

$$r(\Theta_\ell) \leq d^{\frac{p}{2}}. \quad (7)$$

На шаге 2 максимальный ТТ-ранг потенциалов не возрастает, а значит после шага 2 неравенство (7) всё ещё выполнено.

При суммировании тензоров на шаге 3, ТТ-ранги растут аддитивно. Чтобы завершить доказательство, осталось вспомнить, что всего есть  $m$  потенциалов.  $\square$

## 5.2 ТТ-формат для тензора вероятности

Алгоритм из раздела 5.1 легко адаптировать для построения ТТ-представления тензора ненормированной вероятности  $\hat{\mathbf{P}}$ .

На шагах 1 и 2 алгоритма вместо потенциалов  $\Theta_\ell$  следует работать с факторами  $\Psi_\ell$ . На шаге 3 вместо суммы нужно вычислить поэлементное произведение:

$$\hat{\mathbf{P}} = \bigodot_{\ell=1}^m \Psi_\ell. \quad (8)$$

Данный алгоритм построит точное ТТ-представление тензора  $\hat{\mathbf{P}}$ . Тем не менее, ТТ-ранги  $\hat{\mathbf{P}}$  экспоненциально зависят от количества вершин (рис. 2), что делает ТТ-представление тензора вероятности неприменимым на больших задачах. На рис. 2 изображены ранги точного ТТ-представления тензора энергии и вероятности для MRF разного размера. ТТ-ранги тензора вероятности остаются экспоненциально большими и после округления  $\hat{\mathbf{P}}$  с точностью  $\varepsilon = 10^{-8}$ . Таким образом, точного низкорангового ТТ-представления  $\hat{\mathbf{P}}$  не существует.

## 6 Вычисление нормировочной константы $Z$

Естественный подход к подсчету нормировочной константы состоит в представлении всего тензора ненормированной вероятности  $\hat{\mathbf{P}}$  в ТТ-формате и в последующем суммировании всех его элементов. На практике тензор  $\hat{\mathbf{P}}$  обладает экспоненциально большими ТТ-рангами, и работа с его ТТ-представлением становится неэффективной. С другой стороны, каждый отдельный фактор графической модели можно точно представить в ТТ-формате. В этом разделе предложен алгоритм подсчета нормировочной константы, который работает с ТТ-представлением отдельных факторов  $\Psi_\ell$ , не строя ТТ-представление всего тензора  $\hat{\mathbf{P}}$ .

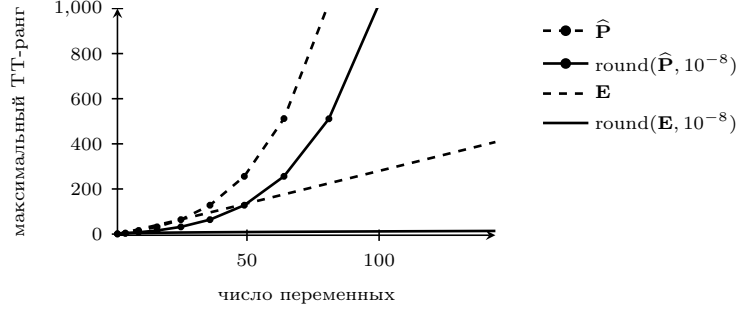


Рис. 2: Максимальный ТТ-ранг тензора энергии  $\mathbf{E}$  и тензора ненормированной вероятности  $\hat{\mathbf{P}}$  для гомогенной решетки Изинга с температурой равной 10 и весами парных потенциалов равными 1. Детали см. в разделе 7.2.

## 6.1 Алгоритм

Пусть все факторы графической модели уже представлены в ТТ-формате (см. раздел 5.2):

$$\Psi_\ell(\mathbf{x}^\ell) = \Psi_\ell(\mathbf{x}) = G_1^{\Psi_\ell}[x_1] \dots G_n^{\Psi_\ell}[x_n]. \quad (9)$$

Далее символ  $G_i^\ell[x_i](\alpha_{i-1}^\ell, \alpha_i^\ell)$  будет использоваться как сокращенное обозначение для  $G_i^{\Psi_\ell}[x_i](\alpha_{i-1}^{\Psi_\ell}, \alpha_i^{\Psi_\ell})$ .

По определению, нормировочная константа  $Z$  вычисляется как сумма значений ненормированного распределения на всех конфигурациях:

$$Z = \sum_{\mathbf{x}} \prod_{\ell=1}^m \underbrace{\Psi_\ell(\mathbf{x})}_{\in \mathbb{R}} = \sum_{x_1, \dots, x_n} \bigotimes_{\ell=1}^m (G_1^\ell[x_1] \dots G_n^\ell[x_n]).$$

Второе равенство выполнено, т. к. кронекерово произведение чисел (матриц размера  $1 \times 1$ ) эквивалентно обычному произведению.

Пользуясь свойством смешанного произведения  $AC \otimes BD = (A \otimes B)(C \otimes D)$ , преобразуем выражение для  $Z$ :

$$Z = \sum_{x_1, \dots, x_n} (G_1^1[x_1] \otimes \dots \otimes G_1^m[x_1]) \dots (G_n^1[x_n] \otimes \dots \otimes G_n^m[x_n]).$$

Обозначим кронекерово произведение матриц  $G_i^\ell[x_i]$  через  $A_i[x_i]$ :

$$A_i[x_i] = G_i^1[x_i] \otimes \dots \otimes G_i^m[x_i].$$

Для любого значения  $x_i$  матрица  $A_i[x_i]$  имеет размеры  $(r_{i-1}(\Psi_1) \dots r_{i-1}(\Psi_m)) \times (r_i(\Psi_1) \dots r_i(\Psi_m))$ . Значение её элементов выражается через элементы матриц  $G_i^\ell[x_i]$ :

$$A_i[x_i](\alpha_{i-1}^1, \dots, \alpha_{i-1}^m; \alpha_i^1, \dots, \alpha_i^m) = G_i^1[x_i](\alpha_{i-1}^1, \alpha_i^1) \dots G_i^m[x_i](\alpha_{i-1}^m, \alpha_i^m).$$

---

**Алгоритм 1** Подсчет нормировочной константы  $Z$ 

---

**Вход:** факторы  $\Psi_1, \dots, \Psi_m$ , точность округления  $\varepsilon$

**Выход:** оценка нормировочной константы  $\tilde{Z}$

- 1: для  $\ell := 1$  до  $m$
  - 2: Найти ТТ-ядра  $G_1^\ell, \dots, G_n^\ell$  для тензора  $\Psi_\ell$
  - 3: Инициализировать  $\mathbf{f}_{n+1} := 1$
  - 4: для  $i := n$  до 1
  - 5: Инициализировать  $B_i := 0$
  - 6: для  $x_i := 1$  до  $d_i$
  - 7: Построить ТТ-матрицу  $A_i[x_i] = \bigotimes_{\ell=1}^m G_i^\ell[x_i]$
  - 8:  $B_i := B_i + A_i[x_i]$
  - 9:  $\overline{\mathbf{f}}_i := B_i \cdot \mathbf{f}_{i+1}$
  - 10:  $\mathbf{f}_i := \text{round}(\overline{\mathbf{f}}_i, \varepsilon)$
  - 11:  $\tilde{Z} := \mathbf{f}_1$
- 

Таким образом, матрица  $A_i[x_i]$  представлена в ТТ-формате, а её ТТ-ранг равен одному (т. к.  $G_i^\ell[x_i](\alpha_{i-1}^\ell, \alpha_i^\ell)$  — это матрица размера  $1 \times 1$ ).

Представим нормировочную константу  $Z$  в виде произведения  $n$  матриц:

$$Z = \sum_{x_1, \dots, x_n} A_1[x_1] \dots A_n[x_n] = \left( \sum_{x_1} A_1[x_1] \right) \dots \left( \sum_{x_n} A_n[x_n] \right) = B_1 \dots B_n,$$

где

$$B_i = \sum_{x_i=1}^{d_i} A_i[x_i].$$

ТТ-представление матрицы  $B_i$  можно получить просуммировав  $d_i$  матриц в ТТ-формате. Все матрицы  $B_i$  можно построить и держать в оперативной памяти, т. к. ТТ-ранги  $B_i$  не превосходят  $d_i$ .

Матрицы  $B_1$  и  $B_n$  являются вектором-строкой и вектором-столбцом соответственно, а значит результат произведения  $B_1 \dots B_n$  — это число. Построив ТТ-матрицы  $B_i$ , их можно перемножать, округляя результат после каждого умножения (см. алгоритм 1). Параметр  $\varepsilon$  контролирует баланс между точностью и скоростью работы алгоритма.

Помимо нормировочной константы  $Z$ , предложенный метод также позволяет найти маргинальные распределения на переменные графической модели. Ненормированное маргинальное распределение  $\hat{p}_i(x_i)$  вычисляется следующим образом:  $\hat{p}_i(x_i) = B_1 \dots B_{i-1} A_i[x_i] B_{i+1} \dots B_n$ . При этом произведение  $B_1 \dots B_{i-1}$  и  $B_{i+1} \dots B_n$ ,  $i = 1, \dots, n$  можно предрасчитать за  $2(n-1)$  умножений ТТ-матриц. Дополнительно рассчитав все произведения вида  $B_i \dots B_j$ ,  $1 \leq i < j \leq n$ , можно вычислять маргинальное распределение на любое подмножество переменных.

## 6.2 Анализ алгоритма 1

В этом разделе предоставлены теоретические гарантии точности оценки нормировочной константы алгоритмом 1.

Обозначим оценку произведения ТТ-матриц  $\{B_j\}_{j=i}^n$  за  $\mathbf{f}_i$ ,  $\mathbf{f}_n = B_n$ ,  $\tilde{Z} = \mathbf{f}_1$ . Умножая ТТ-матрицу на ТТ-вектор и применяя ТТ-округление, получаем  $\mathbf{f}_i$

$$\mathbf{f}_i = \text{round}(B_i \mathbf{f}_{i+1}, \varepsilon),$$

где точность ТТ-округления контролирует относительную точность по евклидовой норме<sup>3</sup>:

$$\|B_i \mathbf{f}_{i+1} - \mathbf{f}_i\|_2 \leq \varepsilon \|B_i \mathbf{f}_{i+1}\|_2. \quad (10)$$

Основной результат представлен в теореме 2, затем приведено следствие, которое легче интерпретировать.

**Теорема 2.** *Для любого набора факторов  $\Psi_1, \dots, \Psi_m$  и любого значения параметра точности округления  $\varepsilon \geq 0$  абсолютная ошибка оценки нормировочной константы, посчитанной алгоритмом 1, не превосходит:*

$$\begin{aligned} |Z - \tilde{Z}| &\leq \|B_1\|_2 \dots \|B_{n-2}\|_2 \cdot \|B_{n-1} \mathbf{f}_n - \mathbf{f}_{n-1}\|_2 + \\ &+ \|B_1\|_2 \dots \|B_{n-3}\|_2 \cdot \|B_{n-2} \mathbf{f}_{n-1} - \mathbf{f}_{n-2}\|_2 + \dots + \\ &+ \|B_1 \mathbf{f}_2 - \mathbf{f}_1\|_2 \end{aligned} \quad (11)$$

**Следствие 1.** *Для любого набора факторов  $\Psi_1, \dots, \Psi_m$  и любого значения параметра точности округления  $\varepsilon \geq 0$  абсолютная ошибка оценки нормировочной константы, посчитанной алгоритмом 1, не превосходит:*

$$|Z - \tilde{Z}| \leq \|B_1\|_2 \dots \|B_n\|_2 ((1 + \varepsilon)^{n-1} - 1) \quad (12)$$

Оценка из следствия менее точная, но позволяет по требуемой итоговой точности выбрать достаточный  $\varepsilon$ .

Чтобы пользоваться результатами теоремы 2, необходимо вычислять 2-норму векторов и матриц в ТТ-формате. 2-норма вектора совпадает с нормой Фробениуса соответствующего тензора, поэтому значения  $\|B_i \mathbf{f}_{i+1} - \mathbf{f}_i\|_2$  легко вычисляются. Хотя подсчет 2-нормы матриц в ТТ-формате является вычислительно сложной задачей, 2-норму можно оценить сверху с помощью нормы Фробениуса или с помощью эмпирически более точной оценки, использующей структуру матрицы  $B_i$ :

$$\begin{aligned} \|B_i\|_2 &= \left\| \sum_{x_i} G_i^1[x_i] \otimes \dots \otimes G_i^m[x_i] \right\|_2 \leq \sum_{x_i} \|G_i^1[x_i] \otimes \dots \otimes G_i^m[x_i]\|_2 = \\ &= \sum_{x_i} \|G_i^1[x_i]\|_2 \dots \|G_i^m[x_i]\|_2 = U_i. \end{aligned} \quad (13)$$

Здесь используется равенство:  $\|A \otimes B\|_2 = \|A\|_2 \|B\|_2$ .

<sup>3</sup>Алгоритм ТТ-округления контролирует относительную точность тензоров по норме Фробениуса, но для векторов норма Фробениуса совпадает с  $L_2$ -нормой.

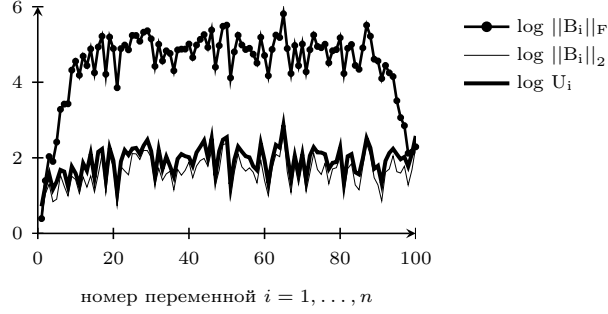


Рис. 3: Сравнение Фробениусовой и спектральной нормы матрицы  $B_i$  с верхней оценкой  $U_i$ . График построен для модели Изинга с решеткой размера  $10 \times 10$ , в которой коэффициенты унарных и парных потенциалов сгенерированы из равномерного распределения на  $[-1, 1]$ , а температура равна 1.

На рис. 3 представлены результаты сравнения величины нормы Фробениуса  $\|B_i\|_F$ , спектральной нормы  $\|B_i\|_2$  и её верхней оценки  $U_i$ . Значения разных норм указаны для всех индексов  $i = 1, \dots, n$  фиксированной модели Изинга.

### 6.3 Доказательство теоремы 2

**Лемма 1.** В условиях теоремы 2 для всех индексов  $i = 1, \dots, n - 1$  верно следующее неравенство:

$$\begin{aligned} \|B_i \dots B_n - \mathbf{f}_i\|_2 &\leq \|B_i\|_2 \dots \|B_{n-2}\|_2 \cdot \|B_{n-1} \mathbf{f}_n - \mathbf{f}_{n-1}\|_2 + \\ &+ \|B_i\|_2 \dots \|B_{n-3}\|_2 \cdot \|B_{n-2} \mathbf{f}_{n-1} - \mathbf{f}_{n-2}\|_2 + \dots + \|B_i \mathbf{f}_{i+1} - \mathbf{f}_i\|_2. \end{aligned} \quad (14)$$

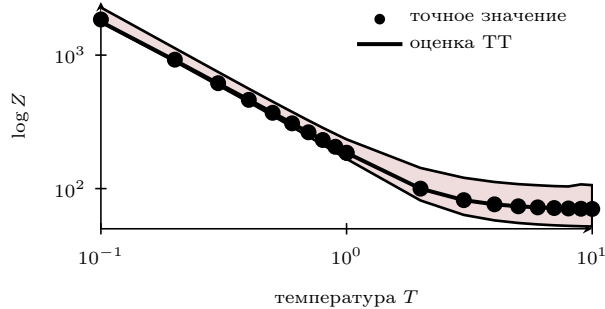


Рис. 4: Доверительный интервал на значение нормировочной константы  $Z$ , полученный из теоремы 2 и неравенства (13). Детали см. в разделе 7.3.

*Доказательство.* Проведем доказательство по индукции.

В качестве базы индукции рассмотрим  $i = n - 1$ :

$$\|B_{n-1}B_n - \mathbf{f}_{n-1}\|_2 = \|B_{n-1}\mathbf{f}_n - \mathbf{f}_{n-1}\|_2.$$

(т. к.  $\mathbf{f}_n = B_n$  по построению.)

Предположим теперь, что (14) верно  $\forall i = j + 1, \dots, n - 1$ . Тогда для  $i = j$  получаем

$$\begin{aligned} \|B_j \dots B_n - \mathbf{f}_j\|_2 &= \\ &= \|(B_j \dots B_n - B_j \mathbf{f}_{j+1}) + (B_j \mathbf{f}_{j+1} - \mathbf{f}_j)\|_2 \leq \\ &\leq \|B_j\|_2 \|B_{j+1} \dots B_n - \mathbf{f}_{j+1}\|_2 + \|B_j \mathbf{f}_{j+1} - \mathbf{f}_j\|_2 \leq \\ &\leq \|B_j\|_2 (\|B_{j+1}\|_2 \dots \|B_{n-2}\|_2 \cdot \|B_{n-1}\mathbf{f}_n - \mathbf{f}_{n-1}\|_2 + \\ &+ \|B_{j+1}\|_2 \dots \|B_{n-3}\|_2 \cdot \|B_{n-2}\mathbf{f}_{n-1} - \mathbf{f}_{n-2}\|_2 + \\ &+ \dots + \|B_{j+1}\mathbf{f}_{j+2} - \mathbf{f}_{j+1}\|_2) + \|B_j \mathbf{f}_{j+1} - \mathbf{f}_j\|_2. \end{aligned}$$

Таким образом, (14) верно для  $i = j$ .  $\square$

*Доказательство теоремы 2.* По построению,  $Z = B_1 \dots B_n$ , а  $\tilde{Z} = \mathbf{f}_1$ . Таким образом, выполнены равенства

$$|Z - \tilde{Z}| = |B_1 \dots B_n - \mathbf{f}_1| = \|B_1 \dots B_n - \mathbf{f}_1\|_2.$$

Здесь используется тот факт, что  $B_1 \dots B_n$  и  $\mathbf{f}_1$  — числа, а для чисел модуль совпадает с  $L_2$ -нормой одноэлементного вектора. Для завершения доказательства осталось применить лемму 1 к полученному выражению.  $\square$

## 6.4 Доказательство следствия 1

**Лемма 2.** В условиях следствия 1 для всех индексов  $i = 1, \dots, n$  верно следующее неравенство:

$$\|\mathbf{f}_i\|_2 \leq \|B_i\|_2 \dots \|B_n\|_2 (1 + \varepsilon)^{n-i}. \quad (15)$$

*Доказательство.* Проведем доказательство по индукции.

Для  $i = n$  утверждение следует из определения  $\mathbf{f}_n$ .

Пусть (15) верно  $\forall j = i + 1, \dots, n$ . Тогда из (10) следует, что

$$\begin{aligned} \|\mathbf{f}_j\|_2 &= \|\mathbf{f}_j - B_j \mathbf{f}_{j+1} + B_j \mathbf{f}_{j+1}\|_2 \leq \\ &\leq \varepsilon \|B_j\|_2 \|\mathbf{f}_{j+1}\|_2 + \|B_j\|_2 \|\mathbf{f}_{j+1}\|_2 = \\ &= \|B_j\|_2 \|\mathbf{f}_{j+1}\|_2 (1 + \varepsilon). \end{aligned}$$

Чтобы завершить доказательство, остаётся воспользоваться предположением индукции для  $\|\mathbf{f}_{j+1}\|_2$ .  $\square$

**Лемма 3.** В условиях следствия 1 для всех индексов  $i = 1, \dots, n - 1$  верно следующее неравенство:

$$\|B_i \mathbf{f}_{i+1} - \mathbf{f}_i\|_2 \leq \|B_i\|_2 \dots \|B_n\|_2 \varepsilon (1 + \varepsilon)^{n-i-1}. \quad (16)$$

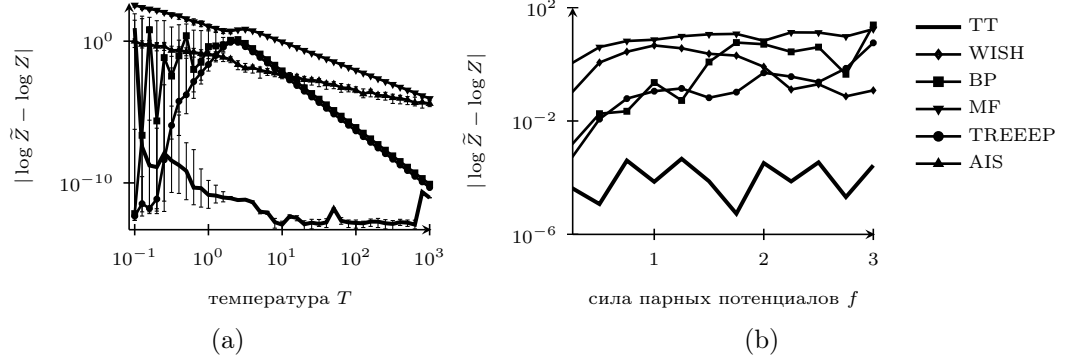


Рис. 5: На графиках представлены результаты экспериментов по подсчету нормировочной константы  $Z$ . На графике (a) предложенный метод (ТТ) сравнивается с другими на гомогенных моделях Изинга разной температуры. Для каждого значения температуры генерируется 50 моделей и указывается медиана и верхняя и нижняя квартили. На графике (b) предложенный метод сравнивается с методом WISH на гетерогенных моделях Изинга. На обоих графиках изображены ошибки оценки  $Z$  (чем меньше значение — тем лучше).

*Доказательство.* Утверждение следует из леммы 2 и неравенства

$$\|B_i \mathbf{f}_{i+1} - \mathbf{f}_i\|_2 \leq \varepsilon \|B_i \mathbf{f}_{i+1}\|_2 \leq \varepsilon \|B_i\|_2 \|\mathbf{f}_{i+1}\|_2,$$

которое в свою очередь следует из (10).  $\square$

*Доказательство следствия 1.* Применяя лемму 3 к неравенству (11), получаем

$$\begin{aligned} |Z - \tilde{Z}| &\leq \|B_1\|_2 \dots \|B_n\|_2 \varepsilon + \|B_1\|_2 \dots \|B_n\|_2 \varepsilon (1 + \varepsilon) + \dots + \\ &+ \|B_1\|_2 \dots \|B_n\|_2 \varepsilon (1 + \varepsilon)^{n-2} = \|B_1\|_2 \dots \|B_n\|_2 \varepsilon (1 + (1 + \varepsilon) + \dots + (1 + \varepsilon)^{n-2}). \end{aligned}$$

Пользуясь формулой суммы геометрической прогрессии  $1 + (1 + \varepsilon) + \dots + (1 + \varepsilon)^{n-2} = ((1 + \varepsilon)^{n-1} - 1)/\varepsilon$ , получаем (12).  $\square$

## 7 Эксперименты

Во всех экспериментах используется неоптимизированная реализация предложенных методов в среде Matlab<sup>4</sup>. Для операций связанных с ТТ-форматом используется ТТ-Toolbox<sup>5</sup>, реализованный в среде Matlab.

<sup>4</sup><https://github.com/bihaqo/TT-MRF>

<sup>5</sup><http://spring.inm.ras.ru/osel/download/tt22.zip>

## 7.1 Модель Изинга

Основная модель для экспериментов — *модель Изинга*. Функция энергии модели Изинга задаётся следующим образом:

$$\mathbf{E}(\mathbf{x}) = -\frac{1}{T} \left( \sum_{i=1}^n h_i x_i + \sum_{(i,j) \in \mathcal{E}} c_{ij} x_i x_j \right), \quad (17)$$

где переменные  $x_i$ ,  $i = 1, \dots, n$  принимают значения из множества  $\{-1, 1\}$ . Коэффициенты  $h_i$  будем называть *унарными весами*, коэффициенты  $c_{ij}$  — *парными весами*, а  $T$  — *температурой*. Парные связи зависят от выбранного множества ребер  $\mathcal{E}$ . В большинстве экспериментов будет использоваться 4-связная решетка размера  $10 \times 10$ . Будем называть модель *гомогенной*, если все парные веса равны друг другу ( $c_{ij} = c$ ), и *гетерогенной* иначе.

## 7.2 ТТ-ранги энергии и вероятности

В данном эксперименте проверяется утверждение из разделов 5.1 и 5.2: при увеличении количества переменных ТТ-ранги тензора энергии растут медленно, а ТТ-ранги тензора ненормированной вероятности — экспоненциально быстро (рис. 2). Рассматривается последовательность моделей Изинга с 4-х связной решеткой возрастающих размеров: от  $1 \times 1$  до  $12 \times 12$ . Унарные веса  $h_i$  сгенерированы из равномерного распределения  $U[-1, 1]$ , все парные веса  $c_{ij}$  равны 1, температура  $T$  равна 10. Для моделей Изинга указан максимальный ТТ-ранг энергии и ненормированной вероятности при точном ТТ-представлении и после ТТ-округления с точностью  $10^{-8}$ . Для точного ТТ-представления тензора вероятности модели Изинга с решеткой размера  $11 \times 11$  не хватает 8GB оперативной памяти.

## 7.3 Нормировочная константа

В данном эксперименте исследуется алгоритм оценки нормировочной константы MRF (алг. 1).

В первом эксперименте метод, основанный на ТТ-разложении (ТТ), сравнивается со следующими методами из библиотеки LibDAI [15]: Belief Propagation (BP) [13], Tree Expectation Propagation (TREEEP) [14], и методом Mean Field (MF) [23]. Так же проводится сравнение с методом annealed importance sampling method (AIS) [16] — представителем методов MCMC. Результаты представлены на рис. 5а. Рассматривается набор моделей Изинга с 4-х связной решеткой размера  $10 \times 10$ , где унарные веса  $h_i$  сгенерированы из равномерного распределения  $U[-1, 1]$ , все парные веса  $c_{ij}$  равны 1, температура  $T$  меняется в пределах от  $10^{-1}$  до  $10^3$ . Для каждого значения температуры сгенерировано 50 моделей Изинга и указана медиана и верхняя и нижняя квартили абсолютной ошибки оценки логарифма нормировочной константы. Для метода AIS были выбраны параметры (1000



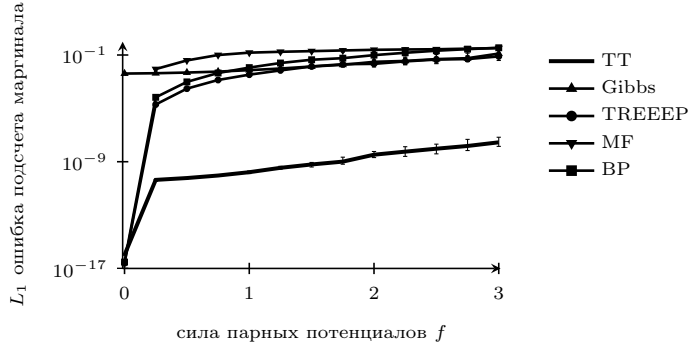


Рис. 6: Ошибка подсчета маргинальных распределений на наборе гетерогенных моделей Изинга. На графике указана среднемодульная ошибка вероятности класса “-1”, усредненная по всем вершинам. Для каждого значения  $f$  генерируется 50 моделей Изинга и указывается медиана и верхняя и нижняя квартили.

промежуточных распределений с выборкой размером 70), максимизирующие точность, достигнутую за 60 секунд. Для подсчета правильных ответов был использован метод junction tree [23].

Во втором эксперименте проводится сравнение с недавно предложенным методом WISH на данных из оригинальной статьи Ермона и др. [3]. Ермон рассматривал набор моделей Изинга с решеткой размера  $10 \times 10$ , унарными весами  $h_i$  сгенерированными из равномерного распределения  $U[-1, 1]$ , температурой  $T$  равной 1, и парными весами  $c_{ij}$  сгенерированными из равномерного распределения  $U[-f, f]$ , где параметр  $f$  меняется от 0.25 до 3. Результаты сравнения представлены на рис. 5б. Отметим, что метод WISH запускался на кластере и каждое вычислительное ядро работало не менее 15 минут. Наш неоптимизированный Matlab код работал не более 53 секунд на каждом примере (32 секунды в среднем).

На рис. 4 изображен доверительный интервал, полученный путем оценивания результата теоремы 2 неравенством (13).

## 7.4 Унарные маргинальные распределения

В этом разделе исследуется качество подсчета маргинальных распределений предлагаемым методом (TT). Метод сравнивается с Belief Propagation (BP) [13], Tree Expectation Propagation (TREEEP) [14], Mean Field (MF) [23] и Gibbs sampling [23], реализованными в библиотеке LibDAI (рис. 6). Рассматривается набор моделей Изинга с решеткой размера  $10 \times 10$ , унарными весами  $h_i$  сгенерированными из равномерного распределения  $U[-1, 1]$ , температурой  $T$  равной 1, и парными весами  $c_{ij}$  сгенерированными из равномерного распределения  $U[-f, f]$ , где параметр  $f$  меняется от 0 до 3. Для

Таблица 2: Относительные значения энергии, достигнутые методом минимизации энергии, основанном на ТТ-разложении. За 0% взята нижняя граница энергии полученная методом TRW-S, за 100% — значение прямой энергии TRW-S.

Задача	$n$	$d$	Энергия ТТ
GEO-SURF-3/GM6	320	3	48.41%
GEO-SURF-3/GM20	348	3	95.83%
GEO-SURF-3/GM203	187	3	98.69%
GEO-SURF-7/GM11	125	7	1769.65%
MATCHING/MATCHING1	19	19	135.47%

каждого значения параметра  $f$  приводятся медиана и квантили сформированные по 50 моделям.

## 7.5 Минимизация энергии

В данном эксперименте исследуется алгоритм минимизации энергии основанный на поиске минимального элемента тензора  $\mathbf{E}$ . Для поиска минимального элемента тензора, представленного в ТТ-формате, строится диагональная матрица в ТТ-формате которая содержит все элементы исходного тензора на своей диагонали. Затем используется DMRG алгоритм Хоромского и Оселедца [11] для поиска минимального собственного значения ТТ-матрицы. Данный метод сравнивается с популярным алгоритмом TRW-S [12] на реальных задачах из бенчмарка OpenGM [10]. Часть результатов представлена в таблице 2. Качество алгоритма DMRG сравнимо с качеством TRW-S и существенно выше на большинстве задач из раздела geo-surf-3. Более эффективные алгоритмы минимизации энергии ещё предстоит разработать.

## 8 Заключение

В работе показано, как использование современных тензорных методов позволяет решать важные задачи в вероятностных графических моделях. Предложенные методы рассмотрены на графических моделях с потенциалами низких порядков, но обобщаются для применения к важным случаям потенциалов высоких порядков.

## Список литературы

- [1] J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart-Young

- decomposition. *Psychometrika*, 35:283–319, 1970.
- [2] S. V. Dolgov and D. V. Savostyanov. Alternating minimal energy methods for linear systems in higher dimensions. Part II: Faster algorithm and application to nonsymmetric systems. arXiv preprint 1304.1222, 2013.
  - [3] S. Ermon, C. Gomes, A. Sabharwal, and B. Selman. Taming the curse of dimensionality: Discrete integration by hashing and optimization. In *International Conference on Machine Learning (ICML)*, 2013.
  - [4] L. Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM J. Matrix Anal. Appl.*, 31:2029–2054, 2010.
  - [5] R. Grosse, C. Maddison, and R. Salakhutdinov. Annealing between distributions by averaging moments. In *Neural Information Processing Systems (NIPS)*, 2013.
  - [6] W. Hackbusch and S. Kühn. A new scheme for the tensor representation. *J. Fourier Anal. Appl.*, 15:706–722, 2009.
  - [7] T. Hazan, S. Maji, and T. Jaakkola. On sampling from the Gibbs distribution with random maximum a posteriori perturbations. In *Neural Information Processing Systems (NIPS)*, 2013.
  - [8] M. Ishteva, H. Park, , and L. Song. Unfolding latent tree structures using 4th order tensors. In *International Conference on Machine Learning (ICML)*, 2013.
  - [9] Y. Jernite, Y. Halpern, and D. Sontag. Discovering hidden variables in noisy-or networks using quartet tests. In *Neural Information Processing Systems (NIPS)*, 2013.
  - [10] J. Kappes, B. Andres, F. Hamprecht, C. Schnoerr, S. Nowozin, D. Batra, S. Kim, B. Kausler, J. Lellmann, N. Komodakis, and C. Rother. A comparative study of modern inference techniques for discrete energy minimization problems. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
  - [11] B. N. Khoromskij and I. V. Oseledets. DMRG+QTT approach to computation of the ground state for the molecular Schrödinger operator. Preprint 69, MPI MIS, Leipzig, 2010.
  - [12] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(10):1568–1583, 2006.
  - [13] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

- [14] T. Minka and Y. Qi. Tree-structured approximations by expectation propagation. In *Neural Information Processing Systems (NIPS)*, 2004.
- [15] J. M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, August 2010.
- [16] R. Neal. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.
- [17] I. V. Oseledets. Tensor-Train decomposition. *SIAM J. Scientific Computing*, 33(5):2295–2317, 2011.
- [18] L. Song, M. Ishteva, A. Parikh, E. Xing, and H. Park. Hierarchical tensor decomposition of latent tree graphical models. In *International Conference on Machine Learning (ICML)*, 2013.
- [19] B. Taskar, C. Guestrin, and D. Koller. Max-Margin Markov networks. In *Neural Information Processing Systems (NIPS)*, 2003.
- [20] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6:1453–1484, 2005.
- [21] L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.
- [22] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51(7):2313–2335, 2005.
- [23] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.