Convex Manifold Approximation for Tensors

by

Kazu Ghalamkari

Dissertation

submitted to the Department of Informatics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

The Graduate University for Advanced Studies, SOKENDAI
March 2023

Kazu Ghalamkari
Convex Manifold Approximation for Tensors
A doctoral dissertation, March 24, 2023
Supervisors: Mahito Sugiyama and Yuichi Yoshida

The Graduate University for Advanced Studies, SOKENDAI
School of Multidisciplinary Sciences
National Institute of Informatics (NII)
Sugiyama Lab.
2 Chome-1-2 Hitotsubashi, Chiyoda City
Tokyo and 101-8430

Committee Members

Chair Assoc. Prof. Dr. Mahito Sugiyama
National Institute of Informatics
The Graduate University for Advanced Studies, SOKENDAI

Member Prof. Dr. Yuichi Yoshida
National Institute of Informatics
The Graduate University for Advanced Studies, SOKENDAI

Member Prof. Dr. Katsumi Inoue
National Institute of Informatics
The Graduate University for Advanced Studies, SOKENDAI

Member Prof. Dr. Kazushi Mimura
Graduate School of Information Sciences
Hiroshima City University

Member Assoc. Prof. Dr. Makoto Yamada
Department of Intelligence Science and Technology
Graduate School of Informatics
Kyoto University

Abstract

This study formulates dimensionality reduction of non-negative multi-dimensional arrays as a convex problem. The key idea is to describe model manifolds containing dimensionality-reduced arrays with a dual-flat coordinate system used in information geometry. We can define a flat model manifold by mapping a multi-dimensional array to a discrete probability distribution and formulating dimensionality reduction as an operation of reducing natural parameters of the distribution. This flatness guarantees that a convex optimization can find an optimal point on the model manifold that globally minimizes the Kullback-Leibler divergence from the data.

In the first part of this dissertation, we analyze the low-rank approximation, a typical dimensionality reduction method, from an information geometric viewpoint, and propose a non-gradient-based low-rank approximation for tensors. The set of low-rank tensors is not guaranteed to be flat. Therefore, we formulate the low-rank approximation as a convex problem by extracting flat subspaces in the space of low-rank tensors and regarding them as model manifolds. As a result, we can solve the non-negative low-rank approximation more efficiently and stably than traditional gradient-based methods. In addition, by focusing on the property that projections onto the model manifold do not change the expected value of the distribution, we derive a solution formula of the best rank-1 simultaneous approximation for multiple matrices. We propose a faster method for rank-1 approximations of matrices with missing values using this formula.

Furthermore, in the second part of this dissertation, we introduce tensor many-body approximation as a novel convex dimensionality reduction method. The proposed method assumes the existence of a major interaction between modes in the tensor instead of a low-rank structure. We describe this interaction with an energy function, following the standard strategy of statistical mechanics. We can introduce the method as a natural extension of the rank-1 approximation for multi-dimensional arrays. The proposed method has the following three advantages: it does not require rank tuning, the cost function is guaranteed to be convex, and the best solution can be obtained faster by the natural gradient method. Furthermore, we propose an interaction representation that visually describes the interaction between modes in the tensor, and transform it into a tensor network to clarify the nontrivial relationship with the existing tensor low-rank approximation.

This research enables efficient dimensionality reduction through discussions across three fields: linear algebra, which deals with tensors and matrices; information geometry, the

geometry of probability distributions; and energy-based models, a methodology inspired by statistical mechanics that deals with interactions.

要約

情報幾何学を用いることで、非負の多次元配列の次元削減を凸問題として定式化する．鍵となるアイデアは，情報幾何学で用いられる双対平坦な座標系を用いて，次元削減後の配列が属する低次元の部分空間（モデル多様体）を記述することである．多次元配列を離散確率分布と対応付け，分布の自然パラメータの削減で次元削減を定式化すると，平坦なモデル多様体を定義できる．この平坦性によって，データからのカルバック・ライブラー情報量を大域的に最小化するモデル多様体上の点をユニークに多項式時間で見つけることができる．

本博士論文の前半では，代表的な次元削減手法である低ランク近似の情報幾何学的な解析によって，勾配法に基づかない，テンソルの低ランク近似を提案する．低ランクテンソルの全体空間に平坦性は保証されないが，低ランクテンソルの全体空間内の平坦な部分空間を抜き出し，これをモデル多様体とすることで，凸問題としての低ランク近似を定式化する．結果として，従来は非凸最適問題として定式化されてきた非負の低ランク近似をより効率的に安定して解けるようになった．また，モデル多様体への射影の前後で分布の期待値が保存されるという性質に着目することで，複数の行列を基底を共有して分解する同時低ランク近似の最良ランク 1 近似公式を閉じた形式で導いた．この公式を応用して，欠損を含む行列のランク 1 近似の高速な解法を提案する．

さらに本博士論文の後半では，データの低ランク構造に注目しない新たな次元削減の方法としてテンソル多体近似を導入する．提案手法では，低ランク構造の代わりに，テンソル内のモード同士の主要な相互作用の存在を仮定する．統計力学の標準的な方策に則り，この相互作用はエネルギー関数で記述する．この近似は多次元配列のランク 1 近似の自然な拡張として導入することができる．提案手法はランクフリーなモデルであり，コスト関数が凸であることが保証され，自然勾配法により高速に最良解が求まる．更に，テンソル内のモード間の相互作用を視覚的に記述する相互作用表示を提案し，これをテンソルネットワークに変換することで，既存のテンソル低ランク近似との非自明な関係についても明らかにした．

本研究は，テンソルや行列を扱う線形代数，確率分布の幾何学である情報幾何学，統計力学にインスパイアされたエネルギーベースモデルという 3 つの領域にまたがった議論により，効率的な次元削減を可能にする．

Acknowledgement

The cooperation of many people made my three years in the Ph.D. course valuable. So here, let me show my gratitude to them all.

First of all, I would like to express my great appreciation to my supervisor, Prof. Dr. Mahito Sugiyama. We certainly spent a lot of time together discussing our research. I am truly proud to have shared with him the excitement of making some of our discoveries. I learned a lot from him, not only the know-how of writing and presenting papers, numerical experiments, and the methodology of how to conduct research, but also a mindset of passion and ambition to keep going with scientific research. It was his constant encouragement and feedback that helped to form the very foundations of my attitude as a researcher.

I would also like to thank my Ph.D. committee members, Prof. Dr. Yuichi Yoshida, Prof. Dr. Katsumi Inoue, Prof. Dr. Kazushi Mimura, and Prof. Dr. Makoto Yamada. They devoted their time to reviewing my thesis and gave me valuable comments. I also had opportunities to discuss with them about my research in intermediate presentations of my doctoral program. These comments were very beneficial in helping me to plan my research direction. Through exchanges with them, I learned that active discussions with experts in various fields are necessary for innovative research.

I also had productive research discussions with my colleagues, Mr. Ryuichi Kanoh, Dr. Profir-Petru Pârțachi, and Dr. Prasad Cheema, as well as fruitful personal interactions. Interaction with these people from various professional and disciplinary backgrounds has helped me develop my research's basic ideas. Moreover, discussions with them sparked my curiosity and sometimes led me to new directions in my study.

Let me highlight that I was able to carry out my research without any environmental difficulties even with the pandemic disaster thanks to the efforts of our secretary, Ms. Hiroko Tokuda, who organized a suitable environment for us. I sincerely thank her and all the administrative staff at NII.

Finally, on a more personal note, I would like to thank my wife, Satomi Fukuda, who encouraged my crazy decision to leave the corporate world and to become a full-time doctoral student. Thanks to her support, the various difficulties and challenges were halved while the thrill of accomplishments and successes were multiplied many times over.

Kazu Ghalamkari
Tokyo, March 2023

Contents

Introduction 1

The size of data handled by computers continues to increase. In order to analyze such large-size data and extract knowledge from them, it is useful to obtain a reduced representation of given data. *Dimensionality reduction*, reducing the dimensionality of data without losing information of the original data, can provide a memory-efficient representation that captures the features of the data [127].

Low-rank approximation is one of the most basic dimensionality reduction methods that has been studied for a long time [84, 48]. Low-rank approximation assumes low-rank structure of data, i.e., that data can be described by a linear combination of a small number of bases, and approximates the data by a linear combination of the dominant bases in the data. It is memory-efficient to handle data by keeping only the dominant basis and coefficients, often called factors, that can reconstruct the original data. Low-rank approximation optimizes the reconstruction error between the input and reconstructed data, and extract dominant factors from various data formats, such as matrices with or without missing values, tensors, and multiple matrices (see Figure 1.1). Low-rank approximation has been used in various areas, including image processing [142, 38], speech recognition [18], bioinformatics [134], data mining [112], deep learning [104] and data compression [58, 61].

In low-rank approximation for matrices, there is a seminal result known as the Eckart–Young–Mirsky theorem [32, 86]. It states that a given matrix's singular value decomposition (SVD) provides the best rank-r matrix that minimizes the reconstruction error defined by the Frobenius norm. SVD can be performed in polynomial time with respect to the size of an input matrix.

CP decomposition [16, 52] and its general form Tucker decomposition [126] are well-known as low-rank approximation for data in the form of tensors. CP decomposition assumes that a tensor can be approximated by the sum of Kronecker products of multiple vectors, yet the best decomposition is known to be NP-hard. Tucker decomposition assumes that a tensor can be approximated by the product of a single core tensor and several matrices. By using high-order singular value decomposition (HOSVD), an extension of SVD, we can obtain a quasi-optimal solution of Tucker decomposition [50].

For data with missing values in the form of matrices or tensors, it has been proposed to perform a low-rank approximation while estimating missing values with the em-*algorithm* or to optimize the weighted reconstruction error based on SVD [115].

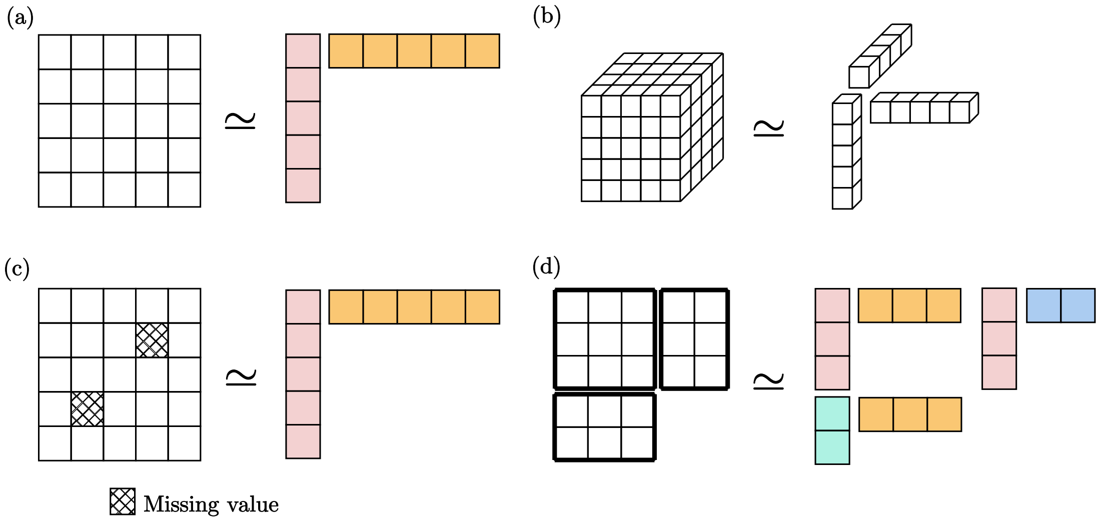


Figure 1.1 Low-rank approximations for (a) matrix, (b) tensor, (c) matrix with missing values and (d) multiple matrices with sharing bases. For simplicity, we assume the target rank is 1.

Non-negative constraints make optimization difficult.

In machine learning and data mining in several fields, such as image and audio processing, non-negative constraints are often imposed on the input and the reconstructed data. Non-negative matrix factorization (NMF) [73] and non-negative Tucker decomposition (NTD) [65] are examples of tasks in which non-negative constraints are imposed, while non-negative constraints make the problem more difficult because negative values may appear in SVD and HOSVD. Therefore, it is common to use the gradient method with the derivative of the reconstruction error. However, since the reconstruction error is usually a nonconvex function, such gradient-based methods have difficulties in setting initial values, convergence criterion, and learning rate.

Basic strategy of our study for non-negative low-rank approximation.

Therefore, this study formulates low-rank approximations of multidimensional arrays as a convex problem by using the theory of optimization established in information geometry [4]. Information geometry is the geometry of probability distributions. Data are represented as empirical distributions, and models are treated as submanifolds, which are subspaces of the overall space of probability distributions. Learning is a projection from the empirical distribution to the submanifold. We can guarantee the uniqueness and convexity of learning by defining the submanifold so that it is flat.

We regard given normalized data as probability distributions, and the set of rank-reduced multidimensional arrays as a model manifold. By describing the model manifold using natural parameters of distributions, we guarantee the flatness of the model manifold

Table 1.1 Correspondence between low-rank approximation and terms of information geometry

	Objects	Solution space	Learning
Information geometry	Distributions	Flat manifold	Orthogonal projection
Low-rank approximation	Multidimensional arrays	Rank-reduced arrays	Approximation

and formulate the low-rank approximation as a convex problem. We illustrates this correspondence in Table 1.1.

In addition to matrices and tensors, we also deal with multiple matrices and matrices with missing values. In order to discuss low-rank approximation for various data structures in a unified manner, we design a partially ordered set (poset) corresponding to the input data structure and describe the data with a log-linear model on the poset. We can see that the subspace in which some of the natural parameters of the model are reduced to zero is the set of data with reduced rank. Non-negative low-rank approximation is a projection onto this subspace. The conceptual diagram summarizing the above is shown in Figure 1.2.

Beyond low-rank approximations.

All of the above discussions have assumed a low-rank structure for data. Therefore, a hyper-parameter called rank is required, which indicates how many dominant bases exist in the data. While a larger rank improves the capability of the decomposed representation, it also increases the computational cost, so the target rank is needed to be appropriately adjusted by considering the tradeoff.

We can formulate a novel dimensionality reduction method that focuses on the mode-structure rather than the low-rank structure by capturing the above discussion by energy-based model, freeing us from rank tuning. The energy-based model, inspired by statistical physics, takes interactions between particles with terms into account in energy function. We define energy function to describe interactions between modes.

The proposed method based on convex optimization, named *many-body approximation*, assumes which tensor modes interact with each other. The proposed method is not only a rank-free model but also has the property of being globally optimizable via information geometry. We can stably obtain the solution in the proposed method because there is no initial value dependence.

As seen above, we formulate low-rank approximation for non-negative multi-dimensional arrays with focusing on convexity, and we also propose a novel convex dimensionality reduction method as an alternative to traditional low-rank approximation.

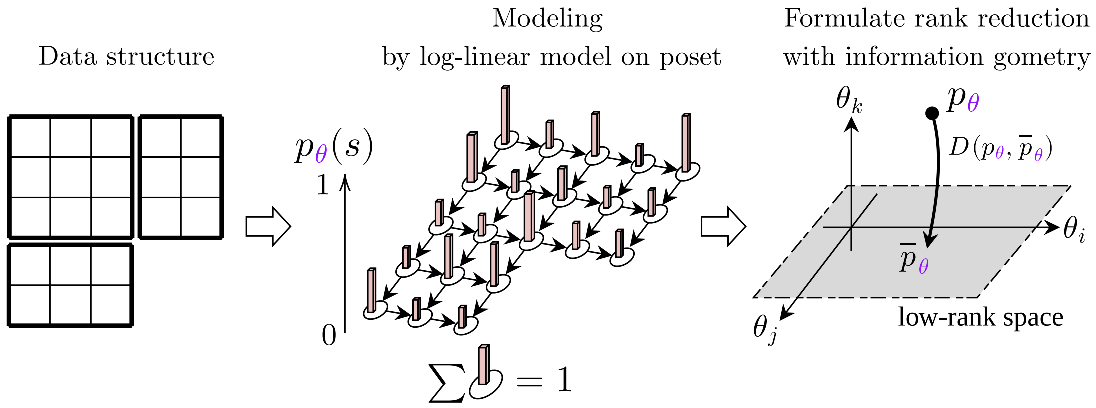


Figure 1.2 A sketch of our strategy for low-rank approximation. We design a poset corresponding to a given data structure (left) and define a discrete probability distribution on that poset (middle). The dimensionality reduction is performed by projecting some of the natural parameters of the probability distribution to zero (right).

Novel dimensionality reduction focusing on geometry

There are many variants of non-negative decomposition for tensors with better performance with modified cost functions [22, 35, 36]. It is known that overfitting can be prevented by optimizing divergence from the input tensor to the reconstructed tensor instead of the Frobenius norm [21]. It is also a popular technique to improve the robustness and generalization performance of low-rank decompositions by adding a regularization term to the cost function [98, 105]. Rather than proposing modified cost functions, we propose a fast and stable decomposition method by focusing on the metric and flat structure of the tensor space, and moreover, we provide a new decomposition that does not require a target rank, which has been difficult to tune in traditional low-rank approximation.

1.1 Main Contributions

In this section, we summarize the contributions of this study. The first major contribution is the formulation of non-negative low-rank approximation as a convex problem for matrices, tensors, multiple matrices, and matrices with missing values by utilizing log-linear models on posets and their information geometry. The specific contributions are described below.

Chapter 3 NeurIPS 2020WS [39], NeurIPS 2021 [40]

We propose a non-gradient based low-rank approximation method, called Legendre Tucker rank Reduction (LTR), for tensors.

- We treated tensors as probability distributions and succeeded in describing the tensor rank by dual flat coordinate system, which is known as the standard coordinate system in information geometry.
- By analyzing rank-1 approximation of tensors using the coordinate system, we pointed out that rank-1 approximation can be viewed geometrically as a mean-field approximation that reduces many-body problems to one-body problems, which is frequently used in statistical physics. Furthermore, by interpreting the exact solution of rank-1 approximation from an information geometrical viewpoint, we constructed an algorithm, called Legendre Tucker rank Reduction (LTR), which performs low-rank approximation for arbitrary Tucker ranks.
- LTR achieves low-rank approximation of tensors by applying the rank-1 approximation formula to each mode, making it about five times faster than traditional gradient-based non-negative low-rank approximations.

Chapter 4 AISTATS 2022 [41]

We find a solution formula of rank-1 approximation for multiple matrices and develop a faster method of rank-1 NMF for matrices with missing values as an application of the formula.

- In the same framework based on information geometry, we analyzed non-negative multiple matrices factorization (NMMF), which simultaneously decomposes multiple non-negative matrices with shared factors, and as a result, derived an analytical solution of rank-1 approximation of NMMF in a closed form that can find the globally optimal solution in polynomial time.
- By focusing on the known correspondence between NMMF and the decomposition of matrices with missing values, we developed an algorithm, called A1GM, which can rapidly find the approximated largest principal components for matrices with missing values without using the gradient method.
- We have shown through experiments on real data that A1GM can find the most dominant factor of a matrix with missing values about 10 times faster than existing methods, without significant loss of approximation accuracy. As with LTR, A1GM can obtain a solution independent of initial values and learning rate.

The above discussion focuses on low-rank structure of data. That is, it assumes that input data can be written as a linear combination of a small number of bases. In the following chapter, as the second major contribution, we formulate a novel method of dimensionality reduction that does not focus on low-rank structure, but rather on a relationship between tensor modes.

Chapter 5 Under review [42]

We develop a novel dimensional reduction method for tensors focusing on relation of tensor modes instead of low-rank structure.

Table 1.2 List of proposed methods

Chap.	Proposed method	Ref.	Implementation URL
3	LTR	[40, 39, 43]	https://github.com/gkazunii/Legendre-tucker-rank-reduction
4	A1GM	[41, 43]	https://github.com/gkazunii/A1GM
5	Many-body Approximation	[42]	https://github.com/gkazunii/MBA

- By adopting the standard methodology of statistical mechanics, which describes interactions using energy functions, for tensor decomposition, we have formulated a tensor many-body approximation that focuses on the relationships between tensor modes. Information geometric analysis enables us to comprehend the tensor many-body approximation as a natural extension of the mean-field approximation.
- While traditional low-rank approximation requires the user to determine the low-rank structure in advance, e.g., CP or Tucker decomposition, and then perform the target rank tuning, the proposed method assumes dominant interactions in a tensor for dimensionality reduction. As a result, the user is free from tuning the ranks.
- We proposed an interaction representation, a diagram that intuitively describes the interaction between modes in a tensor. By transforming this diagram into a tensor network, we reveal a nontrivial relationship between the proposed method and existing low-rank approximation.
- We formulated many-body approximation as a convex optimization problem. Therefore, the proposed method is more stable than nonnegative low-rank approximation that minimizes non-convex cost function.

The list of proposed methods is summarized in Table 1.2.

1.2 Remarks on Terminology

In this monograph, a *tensor* is a multidimensional array. We do not consider tensors as representations of multilinear maps. Each axis of a tensor is called a *mode*. The depth of the tensor, i.e., the number of modes, is called the *order*. For example, third-order tensors $\mathcal{T} \in \mathbb{R}^{I \times J \times K}$ have three modes; each length is I, J and K. Matrices and vectors are regarded as second-order and first-order tensors, respectively. Small tensors obtained by decomposing a tensor are called its *factors*.

Preliminaries

2

In this study, we perform dimensionality reduction of data with various discrete structures. For this purpose, we need a model that can flexibly handle various discrete structures. Therefore, in this chapter, we introduce a log-linear model on posets and some related topics for its optimization based on information geometry as a preparation.

The flexible domain of this model is useful for dealing with various data structures in a unified manner. In the following chapters 3 and 4, we map tensors and multiple matrices to probability distributions in this model by properly designing posets. As an example, in Section 2.1.1, we see that a log-linear model with an ordered power set in its domain can handle higher-order Boltzmann machine.

In this study, we formulate dimensionality reduction as a projection from the empirical distribution corresponding to the data onto a low-dimensional subspace. As described in Section 2.2, a certain flatness is guaranteed in the low-dimensional subspace when some natural parameters of the distribution are fixed and, as a result, the dimensionality reduction can be formulated as a convex problem. By controlling parameters that are imposed to be 0, we can specify the presence or absence of interactions or low-rankness in the data after dimensionality reduction.

In Section 2.2.1, we explain the property that some parameters are preserved in the projection onto flat subspaces, which is the key to enable efficient low-rank approximation in this study. Although the log-linear model on posets is not an invention of the author, the strategy of using this conservation law to achieve optimization without a gradient method is a significant new contribution by the author.

2.1 Log-linear Model on Posets

The log-linear model on a poset [117] is a generalization of Boltzmann machines [1], where we can flexibly design interactions between variables using partial orders. The domain of the model is a set equipped with a partial order, called a *poset*.

Definition 2.1 Poset

A poset $(\Omega, \leq)$ is a set Ω of elements associated with a partial order $\leq$ on Ω, where the relation "$\leq$" satisfies the following three properties: For all $x, y, z \in \Omega$, (1) $x \leq x$, (2) $x \leq y, y \leq x \Rightarrow x = y$, and (3) $x \leq y, y \leq z \Rightarrow x \leq z$.

We can represent a poset as a directed acyclic graph (DAG). This study equates DAG and a poset. If there are multiple DAGs representing the same poset, we suppose to choose a transitive reduced DAG. Transitive reduction is an operation to minimize the number of edges in a DAG while keeping the same poset [3].

We consider a discrete probability distribution p on a poset $(\Omega, \leq)$, which is treated as a mapping $p : \Omega \to (0, 1)$ such that $\sum_{x \in \Omega} p(x) = 1$. Each element $p(x)$ is assumed to be strictly larger than zero. We assume that the structured domain Ω has the least element $\perp$; that is, $\perp \leq x$ for all $x \in \Omega$.

Definition 2.2 Log-linear Model on Poset

The log-linear model for a distribution p on $(\Omega, \leq)$ is defined as

$$\log p(x) = \sum_{s \leq x} \theta(s) \tag{2.1}$$

for $x \in \Omega$.

In this model, $\theta(\perp)$ corresponds to the normalizing factor (partition function). The convex quantity defined as the sign inverse $\psi(\theta) = -\theta(\perp)$ is called the *Helmholtz free energy* of p.

This model belongs to the exponential family and θ corresponds to natural parameters except for $\theta(\perp)$. The exponential family is a set of probability distributions that can be represented as $\log P_{\boldsymbol{\theta}}(x) = C(x) + \sum_{i=1}^{N} \theta_i F_i(x) - \psi(\boldsymbol{\theta})$ using *natural parameters* $\boldsymbol{\theta} = (\theta_1, \dots, \theta_N) \in \mathbb{R}^N$ and normalizing factor $\psi(\boldsymbol{\theta})$. We write a distribution as p_θ to emphasize that it is determined by the natural parameter θ.

In chapters 3–5, we appropriately define the poset $(\Omega, \leq)$ for target data format to reduce their dimensions.

The log-linear model's natural parameter θ uniquely identifies the distribution p. Using θ as a coordinate system in the set of distributions, which is a typical approach in information geometry [4], we can draw the following geometric picture: Each point in the θ-coordinate system corresponds to a distribution. Moreover, because the log-linear model belongs to the exponential family, we can also identify a distribution by expectation parameters defined as follow:

Definition 2.3 Expectation Parameters

Expectation parameters of log-linear model p is given as

$$\eta(x) = \sum_{x \leq s} p(s) \tag{2.2}$$

for $x \in \Omega$.

In fact, using the Möbius function [103] inductively defined as

$$\mu(x, y) = \begin{cases} 1 & \text{if } x = y, \\ -\sum_{x \le s < y} \mu(x, s) & \text{if } x < y, \\ 0 & \text{otherwise,} \end{cases} \tag{2.3}$$

each distribution can be described as

$$p_\eta(x) = \sum_{s \in \Omega} \mu(x, s)\eta(s). \tag{2.4}$$

The above equation is often called *Möbius inversion formula* [103]. We write p_η if p is determined by the expectation parameter η. We can also identify each point in the set of distributions using the η-coordinate system. As is clear from the definition, $\eta(\perp) = 1$ always holds. Each expectation parameter $\eta(x)$ is literally consistent with the expected value $\mathbb{E}[F_x(s)] = \sum_{s \in \Omega} F_x(s)p(s)$ for the function $F_x(s)$ such that $F_x(s) = 1$ if $x \le s$ and 0 otherwise [116].

For a fixed poset $(\Omega, \le)$, the set of distributions $\mathcal{S} = \{\, p(x) \mid x \in \Omega \,\}$ is a Riemannian manifold, and its Riemannian metric is given as follows:

$$G(x, y)(\xi) = \begin{cases} \sum_{s \in \Omega} \zeta(x, s)\zeta(y, s)p(s) - \eta(x)\eta(y) & \text{if } \xi = \theta, \\ \sum_{s \in \Omega} \mu(s, x)\mu(s, y)p(s)^{-1} & \text{if } \xi = \eta, \end{cases} \tag{2.5}$$

where $x, y \in \Omega^+ = \Omega \backslash \{\, \perp \,\}$ and zeta function $\zeta : \Omega \times \Omega \to \{0, 1\}$ defined as

$$\zeta(s, x) = \begin{cases} 1 & \text{if } s \le x, \\ 0 & \text{otherwise.} \end{cases}$$

See the proof for Theorem 3 in [117]. The metric provides us with the definition of distance in $\mathcal{S}$.

Mixture Coordinate In addition, the θ-coordinate and the η-coordinate are orthogonal with each other, which guarantees that we can combine these coordinates together as a mixture coordinate and a point specified by the mixture coordinate also identifies a distribution uniquely [4]. This property is guaranteed in any exponential family. If $N = 2$ as an example, we can specify a distribution by not only θ- and η-coordinate but also mixture coordinate (θ_1, η_2) or (η_1, θ_2).

2.1.1 Boltzmann machine as an Example of Log-linear Model

It is pointed out that the log-linear model on a poset is a generalization of the Boltzmann machine with higher-order interactions [117]. We can represent mth-order Boltzmann machine as

$$p(\boldsymbol{x}) = \frac{1}{Z} \exp \left(\sum_i \theta_i x_i + \sum_{i<j} \theta_{ij} x_i x_j + \cdots + \sum_{i_1 < \cdots < i_m} \theta_{i_1,\ldots,i_m} x_{i_1} \ldots x_{i_m} \right) \tag{2.6}$$

for a binary random variable $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ and the partition function $Z = \exp(\psi(\boldsymbol{\theta}))$. We assume the natural number m is smaller than n.

We consider the log-linear model on $(\Omega_m, \leq)$, where

$$\Omega_m = \{\, \omega \mid \omega \in 2^{\{1,\ldots,n\}}, |\,\omega| \leq m \,\}, \quad \omega_1 \leq \omega_2 \overset{\text{def}}{\Longleftrightarrow} \omega_1 \subseteq \omega_2 \tag{2.7}$$

for any $\omega_1, \omega_2 \in \Omega_m$. We describe the domain $(\Omega_m, \leq)$ in Figure 2.1(a). Then the model in Equation (2.1) becomes

$$\log p(\omega) = \theta(\varnothing) + \sum_i \theta(\{i\}) + \sum_{i<j} \theta(\{i,j\}) + \cdots + \sum_{i_1 < \cdots < i_m} \theta(\{i_1, \ldots, i_m\}). \tag{2.8}$$

Once we regard $\theta(\{i_1, \ldots, i_k\})$ as $\theta_{i_1,\ldots,i_k}$ for $k \leq m$ and $\theta(\varnothing)$ as $\log Z$, the correspondence between the m-th order Boltzmann machine in Equation (2.6) and the log-linear model on $(\Omega_m, \leq)$ in Equation (2.8) is clear.

2.2 Projection Theory in Information Geometry

This section provides the theory of projection in information geometry. Optimizations in Chapters 3-5 are based on the following topics.

Let $\mathcal{S}$ be the set of discrete probability distributions with N random variables. We achieve dimensionality reduction by projection onto a subspace $\mathcal{Q} \subseteq \mathcal{S}$. The entire space $\mathcal{S}$ is a non-Euclidean space with the Fisher information matrix G as the metric. This metric measures the distance between two points. In Euclidean space, the shortest path between two points is a straight line. In a non-Euclidean space, such a shortest path is called a geodesic. In the space $\mathcal{S}$, two kinds of geodesics can be introduced, e-geodesics and m-geodesics. For two points $q_1, q_2 \in \mathcal{S}$, e- and m-geodesics can be defined as

$$\{\, r_t \mid \log r_t = (1-t) \log q_1 + t \log q_2 - \phi(t) \,\}, \quad \{\, r_t \mid r_t = (1-t) q_1 + t q_2 \,\},$$

respectively, where $0 \leq t \leq 1$ and $\phi(t)$ is a normalization factor to keep r_t to be a distribution. We can also represent these geodesics using θ- and η-coordinate as follows[1]:

$$\{\, \theta^{r_t} \mid \theta^{r_t} = (1-t)\theta^{q_1} + t\theta^{q_2} \,\}, \quad \{\, \eta^{r_t} \mid \eta^{r_t} = (1-t)\eta^{q_1} + t\eta^{q_2} \,\}$$

where θ^r and η^r are θ- and η-coordinate of a distribution $r \in \mathcal{S}$. A subspace is called e-flat when any e-geodesic connecting two points in a subspace is included in the subspace. The vertical descent of an m-geodesic from a point $p \in \mathcal{S}$ to q in an e-flat subspace $\mathcal{Q}_e$ is called m-projection. Similarly, e-projection is obtained when we replace all e with m and m with e. The flatness of subspaces guarantees the uniqueness of the projection destination. The projection destination r_m or r_e obtained by m- or e-projection onto $\mathcal{Q}_e$ or $\mathcal{Q}_m$ minimizes the following Kullback–Leibler (KL) divergence [70],

$$r_m = \operatorname*{argmin}_{q \in \mathcal{Q}_e} D(p,q), \quad r_e = \operatorname*{argmin}_{q \in \mathcal{Q}_m} D(q,p). \tag{2.9}$$

The KL divergence from discrete distributions p to q is given as

$$D(p,q) = \sum_{\omega \in \Omega} p(\omega) \log \frac{p(\omega)}{q(\omega)},$$

where Ω is the sample space of p and q. The KL divergence represents a similarity between two probabilities, and it satisfies $D(p,q) = 0 \iff p = q$. When a space is e-flat and m-flat at the same time, we say that the space is *dually-flat*. $\mathcal{S}$ is dually-flat.

$e(m)$-flatness guarantees that cost functions to be optimized in Equation (2.9) are convex. Therefore, $m(e)$-projection onto an $e(m)$-flat subspace can be implemented by a gradient method using a second-order gradient. Second-order differentiation of the KL divergence with parameters leads to a negative Fisher information metric. We call this gradient method the *natural gradient method*. The optimization in Chapter 5 is based on the natural gradient method.

2.2.1 Parameter Conservation in Projections

We assume that distributions in $\mathcal{S}$ are parameterized by N parameters. Let $\mathcal{Q}$ be the set of distributions satisfying the linear condition on $\boldsymbol{\theta}_{1:n}$, a part of the natural parameters $\boldsymbol{\theta}_{1:n} = (\theta_1, \dots, \theta_n)$, where we assume that this part is from 1 to n $(\leq N)$ without loss of generality. Since a subspace defined by linear constrains in θ-parameters is e-flat [4, Chapter 2.4], the m-projection from p onto $\mathcal{Q}$ is unique. This m-projection does not change the rest of the part of expectation parameters $\boldsymbol{\eta}_{n+1:N} = (\eta_{n+1}, \dots, \eta_N)$ [4,

[1] The following representation for continuous distributions is incorrect. This monograph concerns only discrete distributions.

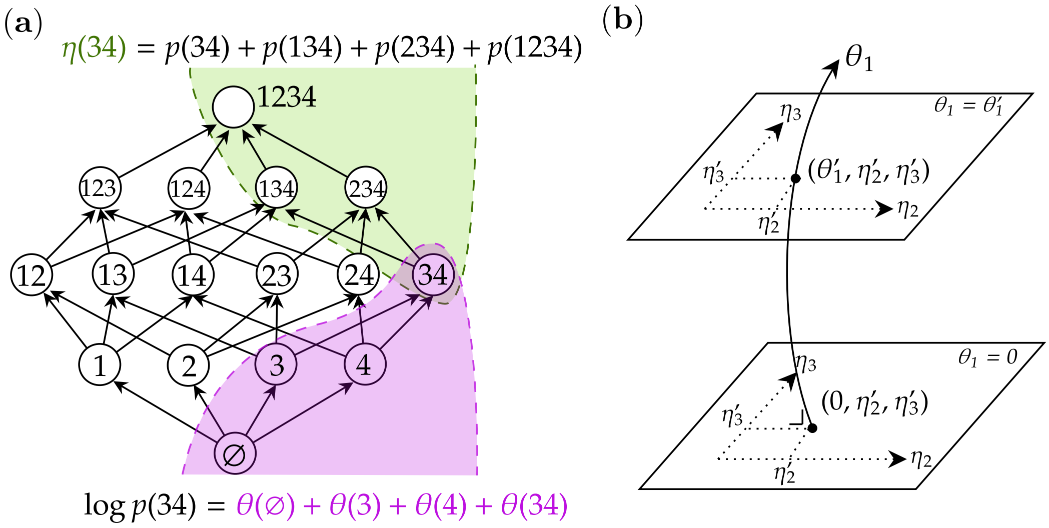


Figure 2.1 (a) The domain of log-linear model for high-order Boltzmann-machine. We described for $n = m = 4$ in Equation (2.6). Arrows indicate ordered relationships between elements in Ω. We omitted braces. For example, "34" means $\{3, 4\}$. (b) An example of expectation conservation law in m-projection for $N = 3$. To distinguish coordinate axes from coordinate values, coordinate values are marked with a symbol "′". The m-projection from a point $(\theta_1, \eta_2, \eta_3) = (\theta_1', \eta_2', \eta_3')$ to subspace satisfying $\theta_1 = 0$ keeps the value of η_2 and η_3.

Chapter 11.3]. In this paper, we call this property *expectation conservation law in m-projection*, which is a key idea to get analytical solutions of some tasks in Chapters 3 and 4. Here we provide the formal description:

Proposition 2.1 Expectation Conservation Law in m-projection [4, Chapter 11.3]

m-projection onto a subspace satisfying a linear condition on $(\theta_1, \ldots, \theta_n)$ does not change the value of $(\eta_{n+1}, \ldots, \eta_N)$, where N is the number of parameters of the distribution.

We provide a sketch of the conservation law in Figure 2.1(b). Similarly, *natural-parameter conservation law in e-projection* is obtained when we replace all θ with η and η with θ in the above discussion, which has important role in the optimization algorithm in tensor balancing [117].

3 Legendre Tucker rank Reduction

We present an efficient low-rank approximation algorithm for non-negative tensors. The algorithm is derived from our two findings: First, we show that rank-1 approximation for tensors can be viewed as a *mean-field approximation* by treating each tensor as a probability distribution. Second, we theoretically provide a sufficient condition for distribution parameters to reduce Tucker ranks of tensors and, interestingly, this sufficient condition can be achieved by iterative application of the mean-field approximation. Since the mean-field approximation is always given as a closed formula, our findings lead to a fast low-rank approximation algorithm without using a gradient method. We empirically demonstrate that our algorithm is faster than the existing non-negative Tucker rank reduction methods with achieving competitive or better approximation of given tensors.

A multidimensional array, or *tensor*, is a fundamental data structure in machine learning and statistical data analysis, and extraction of the essential information contained in tensors has been studied extensively [48, 60]. For second-order tensors – that is, matrices – *low-rank approximation* by singular value decomposition (SVD) is well established [32]. SVD always provides the best low-rank approximation in the sense of arbitrary unitarily invariant norms [86]. In contrast, the problem of low-rank approximation becomes much more challenging for tensors higher than the second order, where the question of how to define the rank of tensors is even nontrivial. To date, various types of ranks – the CP-rank [55, 68], the Tucker rank [28, 126], and the tubal rank [81] – have been proposed, and low-rank approximation of tensors in terms of one of the above two ranks has been widely studied. Furthermore, non-negative low-rank approximation has also been developed, not only for matrices such as NMF [73], but also for tensors [75]. In particular, non-negative Tucker decomposition (NTD) [63] and its efficient variant lraSNTD [141] approximate a given non-negative tensor by a tensor with the lower Tucker rank.

While these approximations have been widely used in various domains such as image classification [66], recommendation [119], and denoising [31], efficient low-rank approximation remains fundamentally challenging. Even the simplest case, the rank-1 approximation in terms of minimizing the Least Squares (LS) error between a given tensor and a low-rank tensor, is known to be NP-hard [53]. Various methods have been developed to efficiently find approximate solutions in polynomial runtime [26, 27, 29, 71, 139].

If we use the Kullback–Leibler (KL) divergence instead of the LS error as a cost function, we can alleviate the problem as the best rank-1 approximation can be obtained in the closed formula [59]. However, the general case of low-rank approximation in terms of the KL divergence is also still under development.

In this chapter, we present a fast low-Tucker-rank approximation method for non-negative tensors. To date, the majority of low-rank approximation methods are based on gradient decent using the derivative of the cost function, which often requires careful tuning of initialization and/or a tolerance threshold. In contrast, our method is not based on a gradient method; the solution is directly obtained based on a closed formula, which we derive from information geometric treatment of tensors. Through an alternative parameterization of tensors by treating them as probability distributions in a statistical manifold, we theoretically provide a sufficient condition for such parameters, called the *bingo rule*, to reduce Tucker ranks of tensors. We then show that low-rank approximation is achieved by *m-projection*; this is one of the two canonical projections in information geometry [4], where a distribution (corresponding to a given non-negative tensor) is projected onto the subspace restricted by the bingo rule (corresponding to the set of non-negative low-rank tensors).

The key idea is that rank-1 approximation for non-negative tensors can be exactly solved by a *mean-field approximation*, a well-established method in physics that approximates a joint distribution by independent distributions [129], as we can represent any non-negative rank-1 tensor by a product of independent distributions. Moreover, we show that the bingo rule, our sufficient condition for tensor Tucker rank reduction, can be achieved by iterative applications of the mean-field approximation. This, combined with the fact that mean-field approximation is computed by m-projection in the closed form, enables us to derive our fast low-Tucker-rank approximation method without using a gradient method.

Our theoretical analysis has a close relationship to [118], whose proposal, called Legendre decomposition, also uses information geometric parameterization of tensors and solves the problem of tensor decomposition by a projection onto a subspace. Although we use the same information geometric formulation of tensors, they did not provide any connection to the Tucker ranks, and Tucker rank reduction is not guaranteed by their approach.

In this chapter, we introduce Legendre Tucker Rank Reduction (LTR), which is a non-gradient method for non-negative low-Tucker-rank approximation. First, we define the task in Section 3.1. Then, we overview our fundamental ideas for LTR in Section 3.2, following the introduction to the algorithm of LTR in Section 3.3 and derive the LTR algorithm in Section 3.4 – 3.7, pointing out the relationship between the rank-1 approximation and mean field approximation[1]. Finally, we mention the relationship between the proposed LTR and related works in Section 3.9.

[1] Implementation is available at: https://github.com/gkazunii/Legendre-tucker-rank-reduction.

3.1 Low-Tucker-rank Approximation for Tensors

First we define the Tucker rank of tensors and formulate the problem of non-negative low-Tucker-rank approximation. The *Tucker rank* of a Dth-order tensor $\mathcal{P} \in \mathbb{R}^{I_1\times\cdots\times I_D}$ is defined as a tuple $(\mathrm{Rank}(\mathbf{P}^{(1)}), \dots, \mathrm{Rank}(\mathbf{P}^{(D)}))$, where each $\mathbf{P}^{(k)} \in \mathbb{R}^{I_k\times\prod_{m\neq k} I_m}$ is the mode-k expansion of the tensor $\mathcal{P}$ [30, 50, 126]. The mode-k expansion of a tensor $\mathcal{P} \in \mathbb{R}^{I_1\times\cdots\times I_D}$ is an operation to convert $\mathcal{P}$ into a matrix $\mathbf{P}^{(k)} \in \mathbb{R}^{I_k\times\prod_{m=1(m\neq k)}^{D} I_m}$. The relation between tensor $\mathcal{P}$ and its mode-k expansion $\mathbf{P}^{(k)}$ is given as,

$$\left(\mathbf{P}^{(k)}\right)_{i_k,j} = \mathcal{P}_{i_1,\dots,i_D}, \quad j = 1 + \sum_{l=1,(l\neq k)}^{D} (i_l - 1)\, J_l, \quad J_l = \prod_{m=1,(m\neq k)}^{l-1} I_m.$$

If the Tucker rank of a tensor $\mathcal{P}$ is $(r_1, \dots, r_D)$, it can always be decomposed as

$$\mathcal{P} = \sum_{i_1=1}^{r_1}\cdots\sum_{i_D=1}^{r_D} \mathcal{G}_{i_1,\dots,i_D}\, \boldsymbol{a}^{(1)}_{i_1} \otimes \boldsymbol{a}^{(2)}_{i_2} \otimes \cdots \otimes \boldsymbol{a}^{(D)}_{i_D} \tag{3.1}$$

with a tensor $\mathcal{G} \in \mathbb{R}^{r_1\times\cdots\times r_d}$, called the *core tensor* of $\mathcal{P}$, and vectors $\boldsymbol{a}^{(k)}_{i_k} \in \mathbb{R}^{I_k}$, $i_k \in [r_k]$, for each $k \in [D]$ where $\otimes$ denotes the Kronecker product [68]. The core tensor and these vectors are often called *factors*.

Task 3.1 Non-negative Low-Tucker-rank Approximation

Non-negative low-Tucker-rank approximation is approximating a given non-negative tensor $\mathcal{P}$ by a non-negative lower-Tucker-rank tensor $\mathcal{T}$, that optimizing the cost function $D(\mathcal{P}, \mathcal{T})$.

In this chapter, we use the Kullback–Leibler (KL) divergence $D(\mathcal{P}, \mathcal{T})$ as the cost function or two non-negative tensors $\mathcal{P}, \mathcal{T} \in \mathbb{R}_{\geq 0}^{I_1\times\cdots\times I_D}$ as follows: [73]

$$D(\mathcal{P}, \mathcal{T}) = \sum_{i_1=1}^{I_1}\cdots\sum_{i_D=1}^{I_D}\left\{\mathcal{P}_{i_1,\dots,i_D}\log\frac{\mathcal{P}_{i_1,\dots,i_D}}{\mathcal{T}_{i_1,\dots,i_D}} - \mathcal{P}_{i_1,\dots,i_D} + \mathcal{T}_{i_1,\dots,i_D}\right\}.$$

In this chapter, we say that a tensor is rank-1 if its Tucker rank is $(1, \dots, 1)$. Note that the task is not non-negative factorization which imposes nonnegativity on factors but low-rank approximation that allows negative factors [114, 49].

We denote by $[n] = \{1, 2, \dots, n\}$ for a positive integer n and denote by $\mathcal{P}_{a^{(k)}:b^{(k)}}$ the subtensor obtained by fixing the range of kth index to only from a to b.

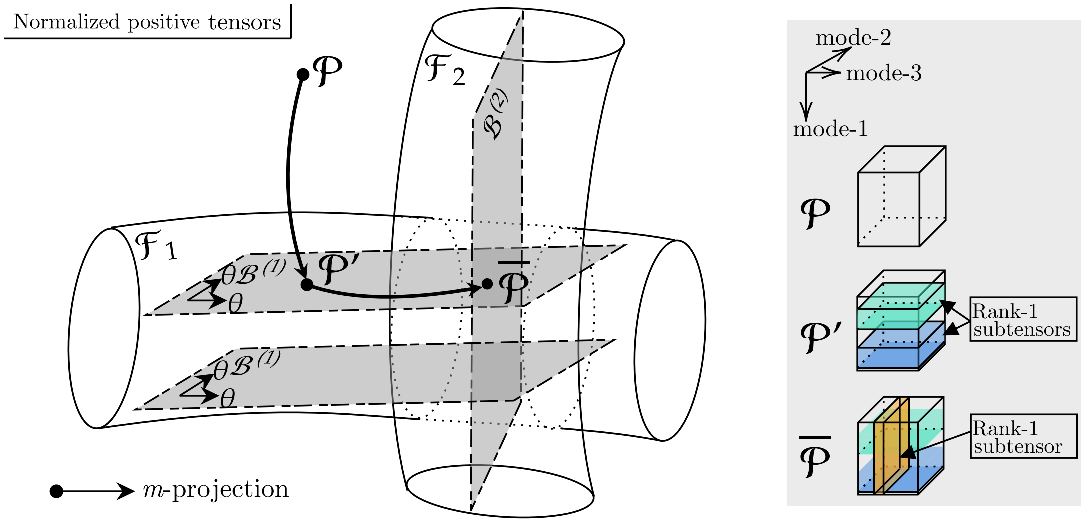


Figure 3.1 An example of reducing Tucker rank of a tensor $\mathcal{P} \in \mathbb{R}_{>0}^{I_1 \times I_2 \times I_3}$ to at most (r_1, r_2, I_3) by the proposed method LTR. $\mathcal{F}_1$ is the set of positive tensors with Tucker rank at most (r_1, I_2, I_3) and $\mathcal{F}_2$ with Tucker rank at most (I_1, r_2, I_3). The best approximation tensor exists in $\mathcal{F}_1 \cap \mathcal{F}_2$, enclosed by the dotted lines. For $m = 1, 2$, there exist e-flat bingo spaces $\mathcal{B}^{(m)} \subset \mathcal{F}_m$. The projection onto $\mathcal{B}_m$ can be performed by dividing $\mathcal{P}$ into subtensors along with mode-m direction and replacing each subtensor with its rank-1 approximation. The choice of bingo space is not unique.

3.2 Idea of LTR

In this subsection, we overview our core idea for LTR. LTR reduces Tucker rank of an input tensor $\mathcal{P}$ by known closed-formula of the best rank-1 approximation. As an example, here we reduce the Tucker rank of a positive tensor $\mathcal{P} \in \mathbb{R}_{>0}^{I_1 \times I_2 \times I_3}$ to at most (r_1, r_2, I_3) as shown in Figure 3.1. In the space of positive tensors, there exists a subspace $\mathcal{F}_1$ consisting of positive tensors of Tucker rank at most (r_1, I_2, I_3) and a subspace $\mathcal{F}_2$ consisting of positive tensors of Tucker rank at most (I_1, r_2, I_3). We want to find a low-Tucker-rank tensor in $\mathcal{F}_1 \cap \mathcal{F}_2$ that approximates $\mathcal{P}$ as close as possible.

First, we map a tensor to a probability distribution. Then, using natural parameters of the distribution, we describe sufficient conditions for reducing the Tucker rank of tensors, called *bingo rule*. For $m = 1$ and 2, we define e-flat subspace $\mathcal{B}^{(m)} \subset \mathcal{F}_m$, called *bingo space*, that satisfies bingo rule. The projection from $\mathcal{P}$ onto $\mathcal{B}^{(1)}$ can be conducted by using known rank-1 approximation formula onto the subtensor of $\mathcal{P}$. Also, the projection from the point on $\mathcal{B}^{(1)}$ onto $\mathcal{B}^{(1)} \cap \mathcal{B}^{(2)}$ can be conducted by the same way.

Bingo spaces cover only tensors generated by applying rank-1 approximations to subtensors along with each mode of a tensor. Therefore, the search space is smaller than the traditional low-rank approximation, which approximates the tensor with an appropriately chosen basis and its coefficients, and there is no guarantee that LTR finds the best approximation; however, we can guarantee that LTR finds a tensor in the selected bingo spaces that minimizes the KL divergence from an input tensor. Such a smaller search space derived by bingo rule makes our algorithm efficient without a gradient method. We discuss this point in more detail in Section 3.7.

3.3 The LTR Algorithm

In LTR, we use the rank-1 approximation method that always finds the rank-1 tensor that minimizes the KL divergence from an input tensor [59].

Theorem 3.1 Best Rank-1 Tensor Approximation Minimizing KL Divergence [59]

For any given non-negative tensor $\mathcal{P} \in \mathbb{R}_{\geq 0}^{I_1 \times \cdots \times I_D}$, its optimal rank-1 tensor $\mathcal{P}$ is given by

$$\overline{\mathcal{P}} = S\left(\mathcal{P}\right)^{1-D} \boldsymbol{s}^{(1)} \otimes \boldsymbol{s}^{(2)} \otimes \cdots \otimes \boldsymbol{s}^{(D)}. \tag{3.2}$$

where each $\boldsymbol{s}^{(k)} = (s_1^{(k)}, \ldots, s_{I_k}^{(k)})$ with $k \in [D]$ is defined as

$$s_{i_k}^{(k)} = \sum_{i_1=1}^{I_1} \cdots \sum_{i_{k-1}=1}^{I_{k-1}} \sum_{i_{k+1}=1}^{I_{k+1}} \cdots \sum_{i_D=1}^{I_D} \mathcal{P}_{i_1,\ldots,i_D}.$$

That is, it is hold that

$$\overline{\mathcal{P}} = \underset{\mathcal{Q};\mathrm{rank}(\mathcal{Q})=1}{\mathrm{argmin}} \; D(\mathcal{P}, \mathcal{Q}).$$

For given tensor $\mathcal{P}$, finding the best rank-1 tensor $\mathcal{R}$ that minimizes $\|\mathcal{P} - \mathcal{R}\|_{\mathrm{F}}$ is known as a NP-hard problem [53].

Now we introduce LTR, which iteratively applies the above rank-1 approximation to subtensors of a tensor $\mathcal{P} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$.

Legendre Tucker rank Reduction

To reduce the Tucker rank of $\mathcal{P} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ to $(r_1, \ldots, r_D)$, LTR performs the following two steps for each $k \in [D]$:

Step 1: We construct $C = \{c_1, \ldots, c_{r_k}\} \subseteq [I_k]$ by random sampling from $[I_k]$ without replacement, where we always assume that $c_1 = 1$ and $c_l < c_{l+1}$ for every $l \in [r_k - 1]$.

Step 2: For each $l \in [r_k]$, if $c_l \neq c_{l+1} - 1$ holds, we replace the subtensor $\mathcal{P}_{c_l^{(k)}:c_{l+1}^{(k)}-1}$ of $\mathcal{P}$ by its rank-1 approximation obtained by Equation (3.2).

The choice of C in **Step 1** is arbitrary, which means that another strategy can be used. For example, if we know that some parts of an input tensor are less important than others, we can directly choose these indices for C instead of random sampling to obtain a more

Algorithm 1: Legendre Tucker rank Reduction

```
input  : Tensor $\mathcal{P}$, target Tucker rank $\mathbf{r} = (r_1, \dots, r_D)$
output : Rank reduced tensor $\mathcal{Q}$
LTR($\mathcal{P}$,$\mathbf{r}$)
    $(I_1, \dots, I_D) \leftarrow$ the size of the input tensor $\mathcal{P}$
    foreach $k = 1, \dots, D$ do
        Construct $\{c_1, \dots, c_{r_k}\} \subseteq [I_k]$ by random sampling from $[I_k]$ without
          replacement, where we always assume that $c_1 = 1$ and $c_i < c_{i+1}$.
        foreach $l = 1, \dots, r_k$ do
            if $c_l \neq c_{l+1} - 1$ then
                Replace the subtensor $\mathcal{P}_{c_l^{(k)}:c_{l+1}^{(k)}-1}$ of $\mathcal{P}$ by its rank-1 approximation as
                $\mathcal{P}_{c_l^{(k)}:c_{l+1}^{(k)}-1} \leftarrow$ BESTRANK1($\mathcal{P}_{c_l^{(k)}:c_{l+1}^{(k)}-1}$)
    $\mathcal{Q} \leftarrow \mathcal{P}$
    return $\mathcal{Q}$
BESTRANK1($\mathcal{P}$)
    $(I_1, \dots, I_D) \leftarrow$ the size of the input tensor $\mathcal{P}$
    foreach $k = 1, \dots, D$ do
        foreach $i_k = 1, \dots, I_k$ do
            $s_{i_k}^{(k)} \leftarrow \sum_{i_1=1}^{I_1} \cdots \sum_{i_{k-1}=1}^{I_{k-1}} \sum_{i_{k+1}=1}^{I_{k+1}} \cdots \sum_{i_D=1}^{I_D} \mathcal{P}_{i_1,\dots i_{k-1},i_k,i_{k+1},\dots,i_D}$
    $\lambda \leftarrow$ sum of all elements of $\mathcal{P}$
    $\overline{\mathcal{P}} \leftarrow \lambda^{1-D} \boldsymbol{s}^{(1)} \otimes \boldsymbol{s}^{(2)} \otimes \cdots \otimes \boldsymbol{s}^{(D)}$
    return $\overline{\mathcal{P}}$
```

accurate reconstructed tensor. We provide the algorithm of LTR in algorithmic format in Algorithm 1.

3.3.1 Computational Complexity of LTR

Step 1 of LTR requires $O(r_1 + r_2 + \cdots + r_D)$ since we only need to sample integers from $1, 2, \dots, I_k$ for each $k \in [D]$ using the Fisher-Yates method [37]. Since the above procedure repeats the best rank-1 approximation at most $r_1 r_2 \dots r_D$ times, the worst computational complexity of LTR is $O(r_1 r_2 \dots r_D I_1 I_2 \dots I_D)$.

3.4 Posets and Modeling for LTR

We derive the LTR algorithm by information geometric formulation of low-Tucker-rank approximation. The discussion is based on the log-linear model on poset. For simplicity, we normalize input tensor beforehand so that the sum is 1. To regard any positive tensor

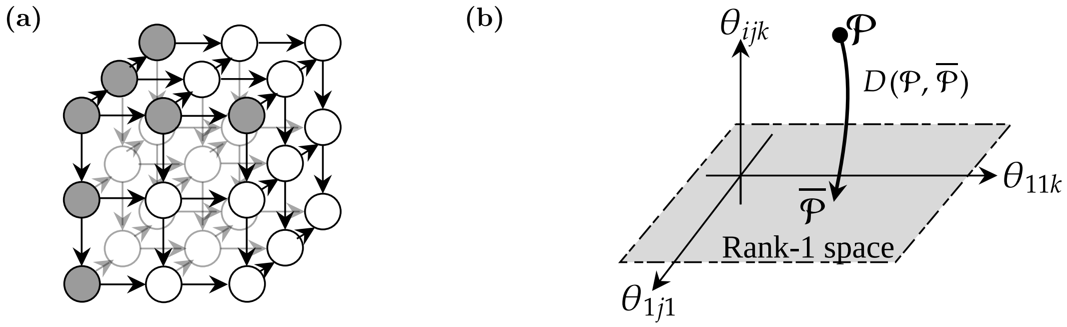


Figure 3.2 (a) A poset $(\Omega_3, \leq)$ corresponding to $3 \times 3 \times 3$ tensor. The parameters on the gray nodes are one-body parameters. (b) The non-negative rank-1 approximation is formulated as a m-projection from input tensor to rank-1 space, minimizing KL divergence.

$\mathcal{P} \in \mathbb{R}_{>0}^{I_1 \times \cdots \times I_D}$ as a distribution, we introduce the following partial order "$\leq$" between each elements $(i_1, \dots, i_D)$ in the index set $\Omega_D = [I_1] \times \cdots \times [I_D]$ of the tensor $\mathcal{P}$,

$$(i_1, \dots, i_D) \leq (i'_1, \dots, i'_D) \Leftrightarrow i_d \leq i'_d \text{ for all } d \in [D]. \tag{3.3}$$

The smallest element in $(\Omega_D, \leq)$ is $\perp = (1, 1, \dots, 1)$. See Figure 3.2 as an example for the poset Ω_D with $I_1 = I_2 = I_3 = 3$ and $D = 3$. We regard $\mathcal{P}$ as a discrete probability distribution whose sample space is the index set of $\mathcal{P}$ by log-linear model on $(\Omega_D, \leq)$. Any positive normalized tensor can be described by natural parameters $(\theta)_{i_1,\dots,i_D} = (\theta_{2,\dots,1}, \dots, \theta_{I_1,\dots,I_D})$ as

$$\mathcal{P}_{i_1,\dots,i_D} = \exp\left(\sum_{(i'_1,\dots,i'_D) \leq (i_1,\dots,i_D)} \theta_{i'_1,\dots,i'_D} \right) = \exp\left(\sum_{i'_1=1}^{i_1} \cdots \sum_{i'_D=1}^{i_D} \theta_{i'_1,\dots,i'_D} \right) \tag{3.4}$$

The condition of normalization is exposed on $\theta_\perp = \theta_{1,\dots,1}$ with $\Omega_D^+ = \Omega_D \backslash (1, \dots, 1)$ as

$$\theta_{1,\dots,1} = -\log \sum_{(i_1,\dots,i_D) \in \Omega_D^+} \exp\left(\sum_{i'_1=1}^{i_1} \cdots \sum_{i'_D=1}^{i_D} \theta_{i'_1,\dots,i'_D} \right). \tag{3.5}$$

A parameter vector $(\theta)_{i_1,\dots,i_D} = (\theta_{2,\dots,1}, \dots, \theta_{I_1,\dots,I_D})$ uniquely identifies the normalized positive tensor $\mathcal{P}$. Therefore, $(\theta)_{i_1,\dots,i_D}$ can be used as an alternative representation of $\mathcal{P}$.

In our modeling in Equation (3.4), which clearly belongs to the exponential family, each value of the vector of η-parameters $(\eta)_{i_1,\dots,i_D}$ is written as follows and uniquely identifies a normalized positive tensor $\mathcal{P}$:

$$\eta_{i_1,\dots,i_D} = \sum_{i'_1=i_1}^{I_1} \cdots \sum_{i'_D=i_D}^{I_D} \mathcal{P}_{i'_1,\dots,i'_D}. \tag{3.6}$$

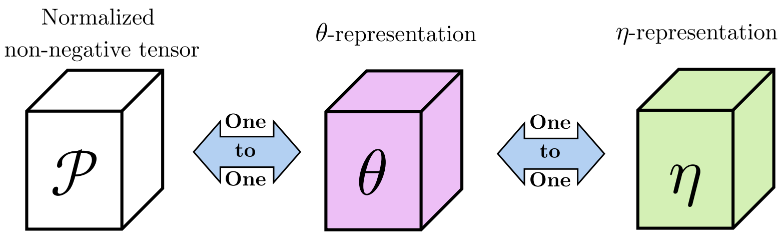


Figure 3.3 We can represent non-negative tensors whose sum is 1 by θ- and η-parameters. This is a coordinate transformation that enables us to easily design flat model manifolds.

The normalization condition is realized as $\eta_{1,\dots,1} = 1$. As shown in [117], by using the Möbius function [103] inductively defined as

$$\mu_{i_1,\dots,i_D}^{i'_1,\dots,i'_D} = \begin{cases} 1 & \text{if } (i_1,\dots,i_D) = (i'_1,\dots,i'_D), \\ -\sum_{j_1=i_1}^{i'_1-1}\cdots\sum_{j_D=i_D}^{i'_D-1}\mu_{i_1,\dots,i_D}^{j_1,\dots j_D}, & \text{if } (i_1,\dots,i_D) \neq (i'_1,\dots,i'_D) \\ & \quad \text{and } (i_1,\dots,i_D) \leq (i'_1,\dots,i'_D), \\ 0 & \text{otherwise.} \end{cases}$$

each distribution $\mathcal{P}$ can be described as

$$\mathcal{P}_{i_1,\dots,i_D} = \sum_{(i'_1,\dots,i'_D)\in\Omega_D} \mu_{i_1,\dots,i_D}^{i'_1,\dots,i'_D}\,\eta_{i'_1,\dots,i'_D} \tag{3.7}$$

using the η-coordinate system. See more general form of Möbius function in Equation (2.3). See Figure 3.3 as a sketch of these representations.

Note that, to identify the value of $\mathcal{P}_{i_1,\dots,i_D}$, we need only $\eta_{i'_1,\dots,i'_D}$ with $(i'_1,\dots,i'_D) \in \{i_1, i_1+1\} \times \{i_2, i_2+1\} \times \cdots \times \{i_D, i_D+1\}$. For example, if $d = 2, 3$, it holds that

$$\begin{aligned} \mathcal{P}_{i_1,i_2} &= \eta_{i_1,i_2} - \eta_{i_1+1,i_2} - \eta_{i_1,i_2+1} + \eta_{i_1+1,i_2+1}, \\ \mathcal{P}_{i_1,i_2,i_3} &= \eta_{i_1,i_2,i_3} - \eta_{i_1+1,i_2,i_3} - \eta_{i_1,i_2+1,i_3} - \eta_{i_1,i_2,i_3+1} \\ &\quad + \eta_{i_1+1,i_2+1,i_3} + \eta_{i_1+1,i_2,i_3+1} + \eta_{i_1,i_2+1,i_3+1} - \eta_{i_1+1,i_2+1,i_3+1}, \end{aligned}$$

where we assume $\eta_{I_1+1,i_2} = \eta_{i_1,I_2+1} = 0$ and $\eta_{I_1+1,i_2,i_3} = \eta_{i_1,I_2+1,i_3} = \eta_{i_1,i_2,I_3+1} = 0$. As the same way, to identify the value of $\theta_{i_1,\dots,i_D}$, we need only $\mathcal{P}_{i'_1,\dots,i'_D}$ with $(i'_1,\dots,i'_D) \in \{i_1-1, i_1\} \times \{i_2-1, i_2\} \times \cdots \times \{i_D-1, i_D\}$. For example, if $D = 2, 3$, it holds that

$$\begin{aligned} \theta_{i_1,i_2} &= \log\mathcal{P}_{i_1,i_2} - \log\mathcal{P}_{i_1-1,i_2} - \log\mathcal{P}_{i_1,i_2-1} + \log\mathcal{P}_{i_1-1,i_2-1}, \\ \theta_{i_1,i_2,i_3} &= \log\mathcal{P}_{i_1,i_2,i_3} - \log\mathcal{P}_{i_1-1,i_2,i_3} - \log\mathcal{P}_{i_1,i_2-1,i_3} - \log\mathcal{P}_{i_1,i_2,i_3-1} \\ &\quad + \log\mathcal{P}_{i_1-1,i_2-1,i_3} + \log\mathcal{P}_{i_1,i_2-1,i_3-1} + \log\mathcal{P}_{i_1,i_2-1,i_3-1} - \log\mathcal{P}_{i_1-1,i_2-1,i_3-1}, \end{aligned}$$

where we assume $\mathcal{P}_{0,i_2} = \mathcal{P}_{i_1,0} = 1$ and $\mathcal{P}_{i_1,i_2,0} = \mathcal{P}_{i_1,0,i_3} = \mathcal{P}_{0,i_2,i_3} = 1$. The sign of each term is related to inclusion–exclusion principle [7, Chapter 6].

3.5 Information Geometric View of Rank-1 Approximation

Before we dive into the general case of non-negative low-Tucker-rank approximation, here we focus on the problem of rank-1 approximation for positive tensors and show the fundamental relationship with the *mean-field theory*.

We describe the necessary and sufficient condition for the rank of a tensor to be 1 using θ- and η-parameters. We formulate tensor rank-1 approximation as a projection onto a subspace consisting of positive rank-1 tensors, which is called a *rank-1 space*. We use the overline for rank-1 tensors; that is, $\overline{\mathcal{P}}$ is a rank-1 tensor, and $\overline{\theta}, \overline{\eta}$ are corresponding parameters of θ- and η-coordinates.

For the sake of clarity, we define the terms one-body and many-body parameter.

Definition 3.1 One-body and Many-body Parameter

Let a *one-body* parameter be a parameter of which at least $D-1$ indices are 1. Parameters other than one-body parameters are called *many-body* parameters.

For example, $\theta_{1,1,3,1}$ and $\eta_{1,5,1,1}$ are one-body parameters for $D=4$. These namings come from the Boltzmann machine [1], which is a special case of the log-linear model [117], where a one-body parameter corresponds to a bias and a many-body parameter to a weight. We also use the following notation for one-body parameters of a Dth-order tensor,

$$\theta_j^{(d)} \equiv \theta_{\underbrace{1,\dots,1}_{d-1},j,\underbrace{1,\dots,1}_{D-d}}, \qquad \eta_j^{(d)} \equiv \eta_{\underbrace{1,\dots,1}_{d-1},j,\underbrace{1,\dots,1}_{D-d}} \qquad \text{for each } d \in [D].$$

We will extend the above concept as n-body parameters in Chapter 5 (See Definition 5.1). The rank-1 condition for positive tensors is described as follows using many-body θ parameters, and we also have succeeded in describing the rank-1 condition using the η-parameter.

Proposition 3.1 Rank-1 Condition (θ-representation)

For any positive tensor $\overline{\mathcal{P}}$, $\mathrm{rank}(\overline{\mathcal{P}}) = 1$ if and only if its all many-body θ-parameters are 0.

Proof: First, we show that $\mathrm{rank}(\overline{\mathcal{P}}) = 1 \Rightarrow$ all many-body θ-parameters are 0. From the assumption of $\mathrm{rank}(\overline{\mathcal{P}}) = 1$, the m-th row of the mode-k expansion of $\overline{\mathcal{P}}$ have to be a constant multiple of the $(m-1)$-th row for all $m \in [2, I_k]$ and $k \in [D]$. That is,

$$
\begin{aligned}
\frac{\overline{\mathbf{P}}^{(k)}_{m,j}}{\overline{\mathbf{P}}^{(k)}_{m-1,j}} &= \frac{\overline{\mathcal{P}}_{i_1,\dots,i_{k-1},m,i_{k+1},\dots,i_D}}{\overline{\mathcal{P}}_{i_1,\dots,i_{k-1},m-1,i_{k+1},\dots,i_D}} \\
&= \exp\left(\sum_{i'_1=1}^{i_1} \cdots \sum_{i'_{k-1}=1}^{i_{k-1}} \sum_{i'_{k+1}=1}^{i_{k+1}} \cdots \sum_{i'_D=1}^{i_D} \theta_{i'_1,\dots,i'_{k-1},m,i'_{k+1},\dots,i'_D} \right)
\end{aligned}
$$

can depend on only m. If a many-body parameter $\theta_{i'_1,\dots,m,\dots i'_D}$ is not 0, the left side of the above equation depends on indices other than m. For example, if a many-body parameter $\theta_{2,1,\dots,1,m,1,\dots 1}$ is not 0, the right side of the equation depends on the value of i_1, which implies contradiction with the assumption that the m-th row is a constant multiple of the $(m-1)$-th row. Therefore, all many-body θ-parameters of rank-1 tensor are 0.

Next, we show that $\mathrm{rank}(\overline{\mathcal{P}}) = 1$ if all many-body θ-parameters are 0. If all many-body θ-parameters are 0, we have

$$
\overline{\mathcal{P}}_{i_1,\dots,i_D} = \exp\left(\theta_{1,1,\dots,1}\right) \prod_{k=1}^{D} \exp\left(\sum_{i'_k=2}^{i_k} \theta^{(k)}_{i'_k} \right).
$$

Then we can represent the tensor $\overline{\mathcal{P}}$ as the Kronecker products of D vectors $\boldsymbol{s}^{(1)} \in \mathbb{R}^{I_1}, \boldsymbol{s}^{(2)} \in \mathbb{R}^{I_2}, \dots,$ and $\boldsymbol{s}^{(D)} \in \mathbb{R}^{I_D}$, whose elements are described as

$$
s^{(k)}_{i_k} = \exp\left(\frac{\theta_{1,\dots,1}}{d} \right) \exp\left(\sum_{i'_k=2}^{i_k} \theta^{(k)}_{i'_k} \right)
$$

for each $k \in [D]$ and $i_k \in [I_k]$. Thus, $\mathrm{rank}(\overline{\mathcal{P}}) = 1$ followed by the definition of the tensor rank. □

Proposition 3.2 Rank-1 Condition (η-representation)

For any positive Dth-order tensor $\overline{\mathcal{P}} \in \mathbb{R}^{I_1 \times \dots \times I_D}_{>0}$, $\mathrm{rank}(\overline{\mathcal{P}}) = 1$ if and only if its all many-body η-parameters are factorizable as

$$
\overline{\eta}_{i_1,\dots,i_D} = \prod_{k=1}^{D} \overline{\eta}^{(k)}_{i_k}. \tag{3.8}
$$

Proof: First, we show that all many-body η-parameters are factorizable if $\mathrm{rank}(\overline{\mathcal{P}}) = 1$. Since we can decompose a rank-1 tensor as a product of normalized independent distributions $\boldsymbol{s}^{(k)} \in \mathbb{R}^{I_k}$ as shown in Equation (3.12), we can decompose many-body η parameters of $\overline{\mathcal{P}}$ as follows:

$$
\begin{aligned}
\overline{\eta}_{i_1,\dots,i_D} &\overset{(3.6)}{=} \sum_{i'_1=i_1}^{I_1} \cdots \sum_{i'_D=i_D}^{I_D} \overline{\mathcal{P}}_{i'_1,\dots,i'_D} \\
&\overset{(3.12)}{=} \sum_{i'_1=i_1}^{I_1} \cdots \sum_{i'_D=i_D}^{I_D} \left(s^{(1)}_{i'_1} s^{(2)}_{i'_2} \dots s^{(D)}_{i'_D} \right) \\
&= \left(\sum_{i'_1=i_1}^{I_1} s^{(1)}_{i'_1} \right) \left(\sum_{i'_2=i_2}^{I_2} s^{(2)}_{i'_2} \right) \cdots \left(\sum_{i'_D=i_D}^{I_D} s^{(D)}_{i'_D} \right) \\
&= \prod_{k=1}^{D} \left(\sum_{i'_k=i_k}^{I_k} s^{(k)}_{i'_k} \right) \\
&= \prod_{k=1}^{D} \left(\sum_{i'_k=i_k}^{I_k} s^{(k)}_{i'_k} \sum_{i'_1=1}^{I_1} s^{(1)}_{i'_1} \sum_{i'_2=1}^{I_2} s^{(2)}_{i'_2} \cdots \sum_{i'_D=1}^{I_D} s^{(D)}_{i'_D} \right) \\
&= \prod_{k=1}^{D} \left(\overline{\eta}^{(k)}_{i_k} \sum_{i'_k=1}^{I_k} s^{(k)}_{i'_k} \right) \\
&= \prod_{k=1}^{D} \overline{\eta}^{(k)}_{i_k},
\end{aligned}
$$

where we use the normalization condition

$$
\sum_{i'_k=1}^{I_k} s^{(k)}_{i'_k} = 1
$$

for each $k \in [D]$.

Next, we show the opposite direction. If all many-body η-parameters are factorizable, it follows that

$$
\begin{aligned}
\overline{\mathcal{P}}_{i_1,\dots,i_D} &\overset{(3.7)}{=} \sum_{(i'_1,\dots,i'_D)\in\Omega_D} \left(\mu^{i'_1,\dots,i'_D}_{i_1,\dots,i_D} \prod_{k=1}^{D} \overline{\eta}^{(k)}_{i'_k} \right) \\
&= \sum_{(i'_1\dots i'_D)\in\Omega_D} \left(\prod_{k=1}^{D} \mu^{i'_k}_{i_k} \overline{\eta}^{(k)}_{i'_k} \right) \\
&= \prod_{k=1}^{D} \left(\overline{\eta}^{(k)}_{i_k} - \overline{\eta}^{(k)}_{i_k+1} \right) \\
&\equiv \prod_{k=1}^{D} s^{(k)}_{j_k}.
\end{aligned}
$$

This formula means that the tensor $\mathcal{P}$ can be represented as a Kronecker product of vectors $\boldsymbol{s}^{(1)}, \dots, \boldsymbol{s}^{(D)}$. Thus, $\mathrm{rank}(\overline{\mathcal{P}}) = 1$ holds by the definition of the tensor rank. □

Since the rank-1 space is described by linear constraints in θ parameters, the rank-1 space is e-flat [4, Chapter 2.4]. Using both Propositions 3.1 and 3.2, we can derive the projected destination without the gradient method, which reproduces the closed formula of the best rank-1 approximation minimizing KL divergence [59] from the view of information geometry. Note that we also use the expectation conservation low in the m-projection in the following proof. In this case, one-body η-parameters do not change during the m-projection as follows:

$$\eta^{(k)}_{i_k} = \overline{\eta}^{(k)}_{i_k} \text{ for any } i_k \in [I_k] \text{ and } k \in [D]. \tag{3.9}$$

See more general statements in Proposition 2.1.

Proposition 3.3 m-projection onto Factorizable Subspace

For any positive tensor $\mathcal{P} \in \mathbb{R}^{I_1 \times \cdots \times I_D}_{>0}$, its m-projection onto the rank-1 space is given as

$$\overline{\mathcal{P}}_{i_1,\dots,i_D} = \prod_{k=1}^{D} \left(\sum_{i'_1=1}^{I_1} \cdots \sum_{i'_{k-1}=1}^{I_{k-1}} \sum_{i'_{k+1}=1}^{I_{k+1}} \cdots \sum_{i'_D=1}^{I_D} \mathcal{P}_{i'_1,\dots,i'_{k-1},i_k,i'_{k+1},\dots,i_D} \right). \tag{3.10}$$

Proof:

$$\begin{aligned}
\overline{\mathcal{P}}_{i_1,\dots,i_D} &\overset{(3.7)}{=} \sum_{(i'_1 \dots i'_D) \in \Omega_D} \mu^{i'_1,\dots,i'_D}_{i_1 \dots i_D} \overline{\eta}_{i'_1,\dots,i'_D} \\
&\overset{(3.8)}{=} \sum_{(i'_1,\dots,i'_D) \in \Omega_D} \left(\mu^{i'_1,\dots,i'_D}_{i_1,\dots,i_D} \prod_{k=1}^{D} \overline{\eta}^{(k)}_{i'_k} \right) \\
&\overset{(3.9)}{=} \sum_{(i'_1 \dots i'_D) \in \Omega_D} \left(\mu^{i'_1,\dots,i'_D}_{i_1,\dots,i_D} \prod_{k=1}^{D} \eta^{(k)}_{i'_k} \right) \\
&= \sum_{(i'_1,\dots,i'_D) \in \Omega_D} \left(\prod_{k=1}^{D} \mu^{i'_k}_{i_k} \eta^{(k)}_{i'_k} \right) \\
&= \prod_{k=1}^{D} \left(\eta^{(k)}_{i_k} - \eta^{(k)}_{i_k+1} \right) \\
&\overset{(3.6)}{=} \prod_{k=1}^{D} \left(\sum_{i'_1=1}^{I_1} \cdots \sum_{i'_{k-1}=1}^{I_{k-1}} \sum_{i'_{k+1}=1}^{I_{k+1}} \cdots \sum_{i'_D=1}^{I_D} \mathcal{P}_{i'_1,\dots,i'_{k-1},i_k,i'_{k+1},\dots,i_D} \right).
\end{aligned}$$

□

Since the m-projection minimizes the KL divergence, it is guaranteed that $\overline{\mathcal{P}}$ obtained by Equation (3.10) minimizes the KL divergence from $\mathcal{P}$ within the set of rank-1 tensors. If a given tensor is not normalized, we need to divide the right-hand side of Equation (3.10) by the $(D-1)$-th power sum of all entries of the tensor in order to match the scales of the input and the output tensors, which is consistent with Equation (3.2).

3.6 Rank-1 Approximation as Mean-field Approximation

We consider a rank-1 positive tensor $\overline{\mathcal{P}} \in \mathbb{R}_{>0}^{I_1 \times \cdots \times I_D}$ and show that it is represented as a product of independent distributions, which leads to an analogy with the mean-field theory. In the rank-1 space, the normalization condition for Dth-order tensors imposed on $\theta(\perp) = \theta_{1,\dots,1}$ is given as

$$\overline{\theta}_{1,\dots,1} = -\log \prod_{k=1}^{D} \left(1 + \sum_{i_k=2}^{I_k} \exp\left(\sum_{i'_k=2}^{i_k} \overline{\theta}^{(k)}_{i'_k} \right) \right) \tag{3.11}$$

with assigning 0 to every many-body θ parameter in Equation (3.11). Note that the empty sum is treated as zero. Note that the empty sum is treated as zero. Next, by substituting 0 for all many-body parameters in our model in Equation (3.4), we obtain

$$\begin{aligned} \overline{\mathcal{P}}_{j_1,\dots,j_D} &= \exp\left(\overline{\theta}_{1,\dots,1}\right) \prod_{k=1}^{D} \exp\left(\sum_{j'_k=2}^{j_k} \overline{\theta}^{(k)}_{j'_k} \right) \\ &= \prod_{k=1}^{D} \frac{\exp\left(\sum_{j'_k=2}^{j_k} \overline{\theta}^{(k)}_{j'_k}\right)}{1 + \sum_{i_k=2}^{I_k} \exp\left(\sum_{i'_k=2}^{i_k} \overline{\theta}^{(k)}_{i'_k}\right)} \\ &\equiv \prod_{k=1}^{D} s^{(k)}_{j_k}, \end{aligned} \tag{3.12}$$

where $\boldsymbol{s}^{(k)} \in \mathbb{R}^{I_k}$ is a positive first-order tensor normalized as

$$\sum_{j_k=1}^{I_k} s^{(k)}_{j_k} = 1, \tag{3.13}$$

then we can regard $\boldsymbol{s}^{(k)}$ as a probability distribution with a single random variable $j_k \in [I_k]$. The above discussion means that any positive a rank-1 tensor can be represented as a product of normalized independent distributions.

The operation of approximating a joint distribution as a product of independent distributions is called *mean-field approximation*. The mean-field approximation was invented in physics for discussing phase transition in ferromagnets [129]. Nowadays, it appears in a wide range of domains such as statistics [101], game theory [15, 79], and information theory [8]. From the viewpoint of information geometry, [123] developed a theory of mean-field approximation for Boltzmann machines [1], which is defined as

$$p(\boldsymbol{x}) = \exp\left(\sum_{i=1}^{N} b_i x_i + \sum_{i<j} w_{ij} x_i x_j \right)$$

for a binary random variable vector $\boldsymbol{x} \in \{0,1\}^n$ with a bias parameter $\boldsymbol{b} = (b)_i \in \mathbb{R}^n$ and an interaction parameter $\boldsymbol{W} = (w_{ij}) \in \mathbb{R}^{n \times n}$. To illustrate that a rank-1 approximation

	Minimizing KL divergence m-projection	Minimizing Inverse KL divergence e-projection
Mean-field approximation of BM Projection onto e-flat space	Impossible $O(2^n)$ unique	$\eta_i = \sigma(\theta_i + \sum_k \theta_{kj}\eta_k)$ not unique
Rank-1 approximation Projection onto e-flat space	**Closed-formula** unique	

Table 3.1 A sketch of information geometric relationship between mean-field approximation and rank-1 approximation.

can be regarded as a mean-field approximation, we prepare the mean-field theory of Boltzmann machines, as follows.

The mean-field approximation of Boltzmann machines is formulated as the projection from a given distribution onto the e-flat subspace consisting of distributions whose interaction parameters $w_{ij} = 0$ for all i and j, which is called a factorizable subspace. Since the distribution with the constraint $w_{ij} = 0$ for all i and j can be decomposed into a product of independent distributions, we can approximate a given distribution as a product of independent distribution by the projection onto a factorizable subspace. The m-projection onto the factorizable subspace requires knowing the expectation value $\eta_i \equiv \mathbb{E}[x_i]$ of an input distributions and requires $O(2^n)$ computational cost [6], so we usually approximate it by replacing the m-projection with e-projection. The e-projection is usually conducted by a self-consistent equation called *mean-field equation*. The e-projection finds the distribution $\overline{\mathcal{P}}_e$ that minimizes $D(\overline{\mathcal{P}}_e; \mathcal{P})$ for a given distribution $\mathcal{P}$ and the projection is conduced by solving the mean-field equations

$$\eta_i = \sigma\left(b_i + \sum_j w_{ij}\eta_j\right)$$

numerically, where $\sigma(\cdot)$ is a sigmoid function. Note that there is no theoretical guarantee that the e-projection destination is uniquely determined since the factorizable subspace is e-flat but not m-flat. The factorizable subspace has a special property such that we can calculate the expectation value $\eta_i \equiv \mathbb{E}[x_i]$ from a distribution as $\eta_i = \tanh^{-1} b_i$ and also can compute the distribution from the expectation value as $b_i = \frac{1}{2}\log\frac{1-\eta_i}{1-\eta_i}$.

The analogy of rank-1 approximation and mean-field theory is clear. In our modeling, a joint distribution $\mathcal{P}$ is approximated by a product of independent distributions $\boldsymbol{s}^{(k)}$ by projecting $\mathcal{P}$ onto the subspace such that all many-body θ parameters are 0, leading to the rank-1 tensor $\overline{\mathcal{P}}$. Since we can compute expectation parameters η by simply summing the input positive tensor in each axial direction, m-projection can be directly performed in our formulation with $O(I_1 \dots I_D)$, which is computationally infeasible in the case of Boltzmann machines due to $O(2^n)$ cost. Moreover, the rank-1 space has the same property as the factorizable subspace of Boltzmann machines; that is, parameters can be easily computed from the dual parameter in a closed form:

Proposition 3.4 (θ, η)-conversion in Rank-1 Space

For any positive Dth-order rank-1 tensor $\overline{\mathcal{P}} \in \mathbb{R}_{>0}^{I_1 \times \cdots \times I_D}$, its one-body θ- and η-parameters satisfy the following equations

$$\overline{\theta}_j^{(k)} = \log\left(\frac{\overline{\eta}_j^{(k)} - \overline{\eta}_{j+1}^{(k)}}{\overline{\eta}_{j-1}^{(k)} - \overline{\eta}_j^{(k)}}\right), \qquad \overline{\eta}_j^{(k)} = \frac{\sum_{i_k=j}^{I_k} \exp \sum_{i'_k=2}^{i_k} \overline{\theta}_{i'_k}^{(k)}}{1 + \sum_{i_k=2}^{I_k} \exp\left(\sum_{i'_k=2}^{i_k} \overline{\theta}_{i'_k}^{(k)}\right)},$$

where we assume $\overline{\eta}_0^{(k)} = \overline{\eta}_{I_k+1}^{(k)} = 0$.

Proof: As shown in Theorem 2 in [117], the relation between θ and η is obtained by the differentiation of Helmholtz's free energy $\psi(\theta)$, which is defined as the sign inverse normalization factor. For the rank-1 tensor $\overline{\mathcal{P}}$, Helmholtz's free energy $\psi(\theta)$ is given as

$$\psi(\overline{\theta}) = \log \prod_{k=1}^{D} \left(1 + \sum_{i_k=2}^{I_k} \exp\left(\sum_{i'_k=2}^{i_k} \overline{\theta}_{i'_k}^{(k)}\right)\right).$$

We obtain the expectation parameters η by the differentiation of Helmholtz's free energy $\psi(\theta)$ by θ, given as

$$\overline{\eta}_j^{(k)} = \frac{\partial}{\partial \overline{\theta}_j^{(k)}} \psi(\overline{\theta}) = \frac{\sum_{i_k=j}^{I_k} \exp \sum_{i'_k=2}^{i_k} \overline{\theta}_{i'_k}^{(k)}}{1 + \sum_{i_k=2}^{I_k} \exp\left(\sum_{i'_k=2}^{i_k} \overline{\theta}_{i'_k}^{(k)}\right)}.$$

By solving the above equation inverse, we obtain

$$\overline{\theta}_j^{(k)} = \log\left(\frac{\overline{\eta}_j^{(k)} - \overline{\eta}_{j+1}^{(k)}}{\overline{\eta}_{j-1}^{(k)} - \overline{\eta}_j^{(k)}}\right).$$

□

3.7 Bingo Rule for General Low-Tucker-rank Approximation

In this subsection, we extend the above discussion to arbitrary Tucker rank. First, we relax the θ-representation of the rank-1 condition, which was described in Proposition 3.1. We introduce the *bingo rule* as a sufficient condition for the tensor to be rank-reduced. Next, we formulate the low-Tucker-rank approximation as a projection onto the subspace that satisfies this bingo rule. Finally, we show the projection can be achieved by rank-1 approximations for subtensors of input tensor without gradient method.

$$\begin{array}{cccc}
\text{Rank} \leq 5 & \text{Rank} \leq 4 & \text{Rank} \leq 3 & \text{Rank} = 1 \\
\begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} & \theta_{15} \\ \theta_{21} & \theta_{22} & \theta_{23} & \theta_{24} & \theta_{25} \\ \theta_{31} & \theta_{32} & \theta_{33} & \theta_{34} & \theta_{35} \\ \theta_{41} & \theta_{42} & \theta_{43} & \theta_{44} & \theta_{45} \\ \theta_{51} & \theta_{52} & \theta_{53} & \theta_{54} & \theta_{55} \end{bmatrix} &
\begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} & \theta_{15} \\ \theta_{21} & 0 & \theta_{23} & \theta_{24} & \theta_{25} \\ \theta_{31} & 0 & 0 & 0 & 0 \\ \theta_{41} & 0 & 0 & 0 & 0 \\ \theta_{51} & 0 & \theta_{53} & \theta_{54} & \theta_{55} \end{bmatrix} &
\begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} & \theta_{15} \\ \theta_{21} & 0 & \theta_{23} & \theta_{24} & \theta_{25} \\ \theta_{31} & \theta_{32} & 0 & \theta_{34} & \theta_{35} \\ \theta_{41} & \theta_{42} & \theta_{43} & 0 & 0 \\ \theta_{51} & 0 & 0 & 0 & \theta_{55} \end{bmatrix} &
\begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} & \theta_{15} \\ \theta_{21} & 0 & 0 & 0 & 0 \\ \theta_{31} & 0 & 0 & 0 & 0 \\ \theta_{41} & 0 & 0 & 0 & 0 \\ \theta_{51} & 0 & 0 & 0 & 0 \end{bmatrix}
\end{array}$$

Figure 3.4 The relationship between matrix ranks and the bingo rule in the case of $D = 2$ and $I_1 = I_2 = 5$. Bingos on horizontal and vertical direction reduce matrix rank. Bingos on the first column, the first row, and diagonal direction does not have any effect to the matrix rank. In the case of rank-1, the bingo selection is unique.

Definition 3.2 Bingo

Let $(\theta)^{(k)}_{ij} = (\theta^{(k)}_{11}, \dots, \theta^{(k)}_{I_k K})$ with $K = \prod_{m \neq k} I_m$ be the θ-coordinate representation of the mode-k expansion of a tensor $\mathcal{P} \in \mathbb{R}^{I_1 \times \cdots \times I_D}_{>0}$. If there exists an integer $i \in [I_k] \setminus \{1\}$ such that $\theta^{(k)}_{ij} = 0$ for all $j \in [K] \setminus \{1\}$, we say that $\mathcal{P}$ has a *bingo on mode-k*.

Proposition 3.5 Bingo and Tucker rank

If there are b_k bingos on mode-k of a tensor $\mathcal{P}$, it holds that

$$\mathrm{Rank}(\mathbf{P}^{(k)}) \leq I_k - b_k.$$

Proof: If there is a bingo on mode-k, the m-th row of the mode-k expansion of $\mathcal{P}$ is a constant multiple of the $(m-1)$-th row, where m is a number determined by the bingo position. We can confirm that

$$\begin{aligned}
\frac{\mathbf{P}^{(k)}_{m,j}}{\mathbf{P}^{(k)}_{m-1,j}} &= \frac{\mathcal{P}_{i_1,\dots,i_{k-1},m,i_{k+1},\dots,i_D}}{\mathcal{P}_{i_1,\dots,i_{k-1},m-1,i_{k+1},\dots,i_D}} \\
&= \frac{\exp\left(\sum_{i'_1=1}^{i_1} \cdots \sum_{i'_{k-1}=1}^{i_{k-1}} \sum_{i'_k=1}^{m} \sum_{i'_{k+1}=1}^{i_{k+1}} \cdots \sum_{i'_D=1}^{i_D} \theta_{i'_1,\dots,i'_D}\right)}{\exp\left(\sum_{i'_1=1}^{i_1} \cdots \sum_{i'_{k-1}=1}^{i_{k-1}} \sum_{i'_k=1}^{m-1} \sum_{i'_{k+1}=1}^{i_{k+1}} \cdots \sum_{i'_D=1}^{i_D} \theta_{i'_1,\dots,i'_D}\right)} \\
&= \exp\left(\sum_{i'_1=1}^{i_1} \cdots \sum_{i'_{k-1}=1}^{i_{k-1}} \sum_{i'_{k+1}=1}^{i_{k+1}} \cdots \sum_{i'_D=1}^{i_D} \theta_{i'_1,\dots,i'_{k-1},m,i'_{k+1},\dots,i'_D}\right) \\
&= \exp\left(\theta_{1,\dots,1,m,1,\dots,1}\right)
\end{aligned}$$

is just a constant that does not depend on j. When a row is a constant multiple of another row, the rank of the matrix is reduced by a maximum of one, which means $\mathrm{Rank}(\mathbf{P}^{(k)}) \leq I_k - 1$. In the same way, if there are b_k bingos, then b_k rows are constant multiple of the other rows, which means $\mathrm{Rank}(\mathbf{P}^{(k)}) \leq I_k - b_k$. □

Therefore, for any tensor $\mathcal{P} \in \mathbb{R}^{I_1 \times \cdots \times I_D}_{>0}$ such that it has b_k bingos on each mode-k, we can always guarantee that its Tucker rank is at most $(I_1 - b_1, \dots, I_D - b_d)$. We define *bingo space* as a subspace consisting of tensors with bingos. We denote a bingo space by

$\mathcal{B}$. Note that the bingo space is always e-flat since a subspace defined by linear constrains in θ-parameters is e-flat [4, Chapter 2.4]. For a given bingo space $\mathcal{B}$, the set of indices $(i_1, \dots, i_D)$ that imposes bingos $\theta_{i_1,\dots,i_D} = 0$ is called the *bingo-index set* and denoted by $\Omega_{\mathcal{B}}$.

In the case of matrix, that is $D = 2$, the relationship between bingo and matrix rank is shown in Figure 3.4. Note that since the matrix rank is given by $\min(\mathrm{Rank}(\mathbf{P}^{(1)}), \mathrm{Rank}(\mathbf{P}^{(2)}))$, it can be shown immediately from Proposition 3.5 that the matrix rank is less than or equal to $\min(I_1 - b_1, I_2 - b_2)$.

Finally, we prove that LTR successfully reduces the Tucker rank by extending the above discussion. We formulate low-Tucker-rank approximation as an m-projection onto a specific bingo space. This bingo space is constructed in **Step 1** of LTR. Then, in **Step 2**, we perform the m-projection using the closed formula of the rank-1 approximation without a gradient method. We first discuss the case in which the rank of only one mode is reduced, followed by discussing the case in which the ranks of two modes are reduced.

When the rank of only one mode is reduced Let us assume that the target Tucker rank is $(I_1, \dots, I_{k-1}, r_k, I_{k+1}, \dots, I_D)$ for an input positive tensor $\mathcal{P} \in \mathbb{R}_{>0}^{I_1 \times \cdots \times I_D}$ and $r_k < I_k$. Let $\mathcal{B}^{(k)}$ be the set of tensors having $I_k - r_k$ bingos on mode-k and $\Omega_{\mathcal{B}^{(k)}}$ be the set of the bingo indices for mode-k constructed in **Step 1** of LTR:

$$\mathcal{B}^{(k)} = \{\, \mathcal{P} \mid \theta_{i_1,\dots,i_D} = 0 \text{ for } (i_1, \dots, i_D) \in \Omega_{\mathcal{B}^{(k)}} \,\}. \tag{3.14}$$

Note that $\mathcal{P} \in \mathcal{B}^{(k)}$ implies that the Tucker rank of $\mathcal{P}$ is at most $(I_1, \dots, I_{k-1}, r_k, I_{k+1}, \dots, I_D)$. Let $\mathcal{P}_{(k)}$ be the destination of the m-projection from $\mathcal{P}$ onto $\mathcal{B}^{(k)}$ and $\tilde{\theta}$, $\tilde{\eta}$ be its corresponding parameters of θ- and η-coordinates. From the definition of m-projection and the conservation low of η, the parameters of tensor $\mathcal{P}_{(k)}$ satisfy

$$\tilde{\theta}_{i_1,\dots,i_D} = 0 \text{ for } (i_1, \dots, i_D) \in \Omega_{\mathcal{B}^{(k)}}, \quad \tilde{\eta}_{i_1,\dots,i_D} = \eta_{i_1,\dots,i_D} \text{ for } (i_1, \dots, i_D) \notin \Omega_{\mathcal{B}^{(k)}}. \tag{3.15}$$

As described in Section 3.4, we need η-parameters on only $(i'_1, \dots, i'_D) \in \{i_1, i_1 + 1\} \times \cdots \times \{i_D, i_D + 1\}$ to identify the value of $\mathcal{P}_{i'_1,\dots,i'_D}$. It leads that $\mathcal{P}_{i_1,\dots,i_D} = \mathcal{P}_{(k)}{}_{i_1,\dots,i_D}$ for $(i_1, \dots, i_D) \in \hat{\Omega}_{\mathcal{B}^{(k)}}$ for

$$\hat{\Omega}_{\mathcal{B}^{(k)}} = \{\, (i_1, \dots, i_D) \mid \{i_1, i_1 + 1\} \times \cdots \times \{i_D, i_D + 1\} \cap \Omega_{\mathcal{B}^{(k)}} = \varnothing \,\}. \tag{3.16}$$

Therefore, all we have to do to reduce the Tucker rank is to change the elements $\mathcal{P}_{i_1,\cdots,i_D}$ for only $(i_1, \dots, i_D) \notin \hat{\Omega}_{\mathcal{B}^{(k)}}$. Such adjustable parts of $\mathcal{P}$ can be divided into some contiguous blocks, and we call each of them a subtensor of $\mathcal{P}$ on mode-k. In Figure 3.5(a), for example, we can find two subtensors $\mathcal{P}_{3^{(1)}:5^{(1)}}$ and $\mathcal{P}_{7^{(1)}:8^{(1)}}$. By conducting the rank-1 approximation introduced in Section 3.3 onto each subtensor, we obtain $\mathcal{P}_{(k)}$

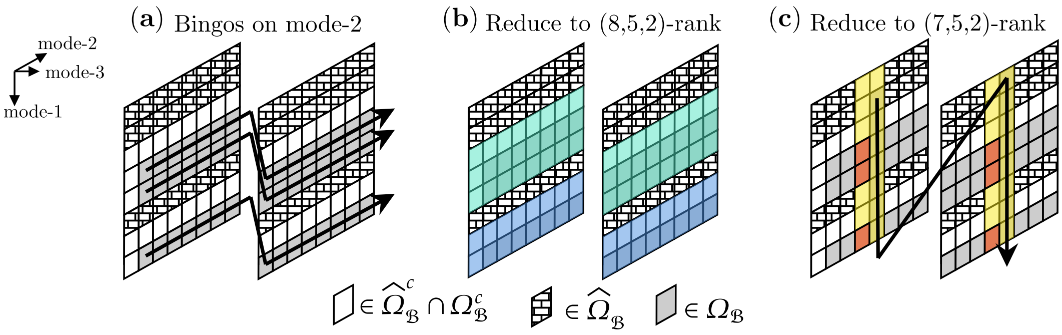

(a) Bingos on mode-2
(b) Reduce to (8,5,2)-rank
(c) Reduce to (7,5,2)-rank
mode-2
mode-3
mode-1
$\in \widehat{\Omega}^{c}_{\mathcal{B}} \cap \Omega^{c}_{\mathcal{B}}$
$\in \widehat{\Omega}_{\mathcal{B}}$
$\in \Omega_{\mathcal{B}}$

Figure 3.5 Examples of LTR for $(8, 8, 2)$ tensor. $\Omega_{\mathcal{B}}$ is bingo-index set. Tensor values and their η-parameters on $\hat{\Omega}_{\mathcal{B}}$ do not change, and the η-parameters on $\hat{\Omega}^{c}_{\mathcal{B}} \cap \Omega^{c}_{\mathcal{B}}$ also do not change, where $\hat{\Omega}^{c}_{\mathcal{B}} = \Omega_D \backslash \hat{\Omega}_{\mathcal{B}}$ and $\Omega^{c}_{\mathcal{B}} = \Omega_D \backslash \Omega_{\mathcal{B}}$. For the target rank $(8, 5, 2)$, we firstly define three bingos on mode-2 as shown in (a) since $8 - 5$ is 3, and approximate two contiguous blocks filled in green and blue in (b) by rank-1 tensor using formula (3.2). As the same way, (c) shows the case where the target rank is $(7, 5, 2)$. We additionally define single bingo on mode-1 since $8 - 7$ is 1. A subtensor approximated by formula (3.2) is filled in yellow. We assume that we project a tensor onto $\mathcal{B}^{(1)}$, followed by projecting it onto $\mathcal{B}^{(2)}$. After the second m-projection, the θ-parameters on red panels seems to be overwritten. However, these values remain to be zero after the second m-projection. Figures (a), (b), and (c) correspond to $\mathcal{P}$, $\mathcal{P}'$, and $\overline{\mathcal{P}}$ in Figure 3.1, respectively.

satisfying Equations (3.15) since Proposition 3.1 and Equation (3.9) hold in these rank-1 approximations.

When the rank of only two modes are reduced Let us assume that the target Tucker rank of mode-k is $r_k < I_k$ and that of mode-l is $r_l < I_l$. In this case, we need to consider two bingo spaces $\mathcal{B}^{(k)}$ and $\mathcal{B}^{(l)}$ associated with bingo index sets $\Omega_{\mathcal{B}^{(k)}}$ and $\Omega_{\mathcal{B}^{(l)}}$. Let $\mathcal{P}_{(k)}$ be the resulting tensor of m-projection of $\mathcal{P}$ onto $\mathcal{B}^{(k)}$ and $\mathcal{P}_{(k,l)}$ be the resulting tensor of m-projection of $\mathcal{P}_{(k)}$ onto $\mathcal{B}^{(l)}$. To get $\mathcal{P}_{(k,l)}$, let us consider m-projection from $\mathcal{P}_{(k)} \in \mathcal{B}^{(k)}$ to the bingo space $\mathcal{B}^{(l)}$. In this projection, the part of θ-parameters which are set to be 0 in the previous m-projection onto $\mathcal{B}^{(k)}$ from $\mathcal{P}$ seems to be overwritten (see red panels in Figure 3.5(b)). However, as shown in the following Proposition, after the rank-1 approximation of a tensor where some one-body θ-parameters are already zero, these parameters remain to be zero.

Proposition 3.6

Let θ denote natural parameters of given tensor $\mathcal{P} \in \mathbb{R}^{I_1 \times \cdots \times I_D}_{>0}$ and $\overline{\theta}$ denote natural parameters of $\overline{\mathcal{P}}$ which is the best rank-1 approximation that minimizes KL divergence from $\mathcal{P}$. If a one-body natural parameter $\theta^{(j)}_{i_j} = 0$, its values after the best rank-1 approximation $\overline{\theta}^{(j)}_{i_j}$ remain 0.

Proof: When $\theta^{(j)}_{i_j} = 0$, it holds that

$$\frac{\mathcal{P}_{1,\ldots,1,i_j,1,\ldots,1}}{\mathcal{P}_{1,\ldots,1,i_j-1,1,\ldots,1}} = \exp\left(\theta^{(j)}_{i_j}\right) = 1.$$

By using the closed formula of the best rank-1 approximation (3.2), we obtain

$$\begin{aligned}
\overline{\mathcal{P}}_{1,\dots,1,i_j,1,\dots,1} &= \prod_{k=1}^{D} \left(\sum_{i'_1=1}^{I_1} \cdots \sum_{i'_{k-1}=1}^{I_{k-1}} \sum_{i'_{k+1}=1}^{I_{k+1}} \cdots \sum_{i'_D=1}^{I_D} \mathcal{P}_{i'_1,\dots,i'_{k-1},1,i'_{k+1},\dots,i_j,\dots,i'_D} \right) \\
&= \prod_{k=1}^{D} \left(\sum_{i'_1=1}^{I_1} \cdots \sum_{i'_{k-1}=1}^{I_{k-1}} \sum_{i'_{k+1}=1}^{I_{k+1}} \cdots \sum_{i'_D=1}^{I_D} \mathcal{P}_{i'_1,\dots,i'_{k-1},1,i'_{k+1},\dots,i_j-1,\dots,i'_D} \right) \\
&= \overline{\mathcal{P}}_{1,\dots,1,i_j-1,1,\dots,1}.
\end{aligned}$$

It follows that

$$\frac{\overline{\mathcal{P}}_{1,\dots,1,i_j,1,\dots,1}}{\overline{\mathcal{P}}_{1,\dots,1,i_j-1,1,\dots,1}} = \exp\left(\overline{\theta}^{(j)}_{i_j}\right) = 1.$$

Finally, we obtain $\overline{\theta}^{(j)}_{i_j} = 0$. □

As a result, the θ- and η-parameters of $\mathcal{P}_{(k,l)}$ satisfy

$$\tilde{\theta}_{i_1,\dots,i_D} = 0 \text{ if } (i_1,\dots,i_D) \in \Omega_{\mathcal{B}^{(k)}} \cup \Omega_{\mathcal{B}^{(l)}}, \tag{3.17}$$

$$\tilde{\eta}_{i_1,\dots,i_D} = \eta_{i_1,\dots,i_D} \text{ if } (i_1,\dots,i_D) \notin \Omega_{\mathcal{B}^{(k)}} \cup \Omega_{\mathcal{B}^{(l)}}, \tag{3.18}$$

where η is expectation parameter of $\mathcal{P}$. That means $\mathcal{P}_{(k,l)}$ is resulting tensor of m-projection from $\mathcal{P}$ onto $\mathcal{B} = \mathcal{B}^{(k)} \cap \mathcal{B}^{(l)}$ since $\Omega_{\mathcal{B}} = \Omega_{\mathcal{B}^{(k)}} \cup \Omega_{\mathcal{B}^{(l)}}$. As a conclusion, we can obtain the projected tensor onto $\mathcal{B}$ by rank-1 approximations on each subtensor of $\mathcal{P}_{(k)}$ on mode-l. We can also immediately confirm $\mathcal{P}_{(l,k)} = \mathcal{P}_{(k,l)}$; that is, the projection order does not matter. The projection sketch is shown in Figure 3.6(b).

For general case Based on the above discussion, we can derive **Step 2** for the general case of low-Tucker-rank approximation. We formulate non-negative low-Tucker-rank approximation as a m-projection onto the intersection of bingo spaces on each mode $\mathcal{B}^{(k)}$, that is $\mathcal{B} = \mathcal{B}^{(1)} \cap \cdots \cap \mathcal{B}^{(D)}$. The m-projection destination is given as an iterative application of m-projection D times, starting from $\mathcal{P}$ onto subspace $\mathcal{B}^{(1)}$, then from there onto subspace $\mathcal{B}^{(2)}$, ..., and finally onto subspace $\mathcal{B}^{(D)}$. Note that each m-projection needs only rank-1 approximation for subtensors on each mode. The result of LTR does not depend on the projection order. Since the m-projection minimizes the KL divergence from the input onto the bingo space, LTR always provides the best low-rank approximation in the specified bingo space $\mathcal{B}$, that is, for a given non-negative tensor $\mathcal{P}$, the output $\mathcal{T}^*$ of LTR satisfies that

$$\mathcal{T}^* = \operatorname*{argmin}_{\mathcal{T} \in \mathcal{B}} D(\mathcal{P}; \mathcal{T}).$$

The usual low-rank approximation without the bingo-space requirement approximates a tensor by a linear combination of appropriately chosen bases. In contrast, our method with the bingo-space requirement approximates a tensor by scaling of bases. Therefore, our method has a smaller search space for low-rank tensors. This search space allows us to derive an efficient algorithm without a gradient method, which always outputs the globally optimal solution in the space.

3.8 Invariance of the Summation in Each Axial Direction

The definition of η in Equation (3.6) suggests that one-body η-parameters are related to the summation of elements of a tensor in each axial direction. The i_k-th summation in the k-th axis is given by

$$\sum_{i_1=1}^{I_1} \cdots \sum_{i_{k-1}=1}^{I_{k-1}} \sum_{i_{k+1}=1}^{I_{k+1}} \cdots \sum_{i_D=1}^{I_D} \mathcal{P}_{i_1,\dots,i_D} = \eta^{(k)}_{i_k} - \eta^{(k)}_{i_{k+1}}.$$

Since the one-body η-parameters do not change by the m-projection, it can be immediately proved that the best rank-1 approximation of a positive tensor in the sense of the KL divergence does not change the sum in each axial direction of the input tensor. Our information geometric insight leads to the fact that the conservation law of sums essentially comes from constant one-body η-parameters during m-projection. This property is a natural extension of the property, such that row sums and column sums are preserved in NMF, which minimizes the KL divergence [56] to tensors. Since the rank-1 reduction preserves the sum in each axial direction of the input tensor, LTR for general Tucker rank reduction also preserves it.

3.9 Relationship to Legendre Decomposition

Our theoretical analysis is closely related to Legendre decomposition [118], which also uses information geometric parameterization of tensors and solves the problem of tensor decomposition by a projection onto a subspace. However, their concept differs from ours in the following aspect. In the Legendre decomposition, any single point in a subspace that has some constraints on the θ-coordinate is taken as the initial state and moves by gradient descent inside the subspace to minimize the KL divergence from an input tensor. This operation is an e-projection, where the constrained θ-coordinates do not change from the initial state. In contrast, we employ the m-projection from the input tensor onto the low-rank space by fixing some η-coordinates. Using the conservation law for η-coordinates, we obtain an exact analytical representation of the coordinates of the projection destination without using a gradient method. Figure 3.6 illustrates the relationship between our approach and Legendre decomposition. Moreover, the Tucker rank is not discussed in the Legendre decomposition, so it is not guaranteed that Legendre

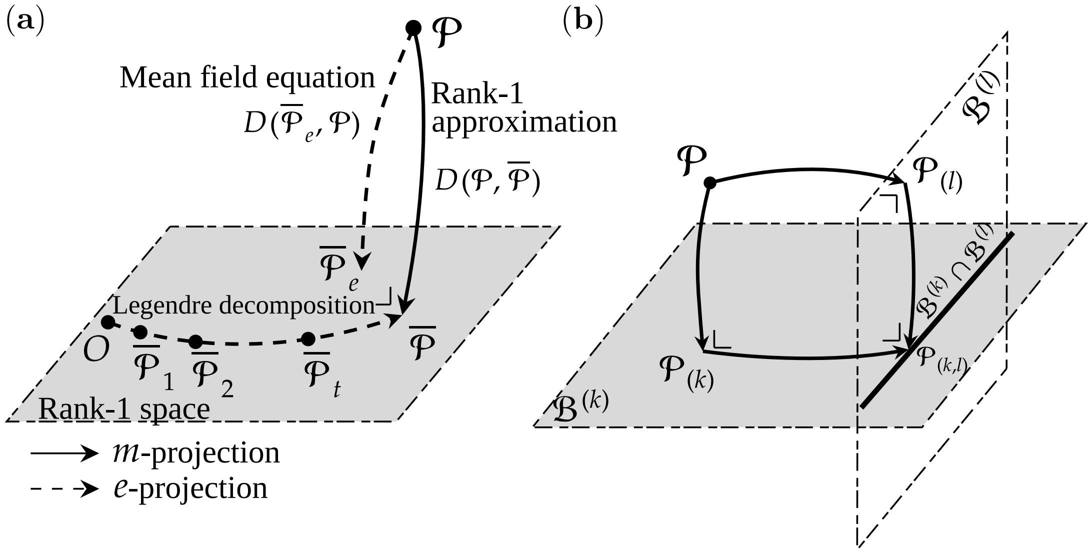


Figure 3.6 (a) The relationship among rank-1 approximation, Legendre decomposition [118], and mean-field equation, where we assume that the same bingo space is used in Legendre decomposition. A solid line illustrates m-projection with fixing one-body η parameters. $\mathcal{P}$ is an input positive tensor and $\overline{\mathcal{P}}$ is the rank-1 tensor that minimizes the KL divergence from $\mathcal{P}$. $\mathcal{O}$ is an initial point of Legendre decomposition, which is usually a uniform distribution. $\overline{\mathcal{P}}_t$ is a tensor of the t-th step of gradient descent in Legendre Decomposition. (b) The m-projection of a common space of two different bingo spaces from $\mathcal{P}$ can be achieved by m-projection into one bingo space and then m-projecting into the other bingo space.

decomposition reduces the Tucker rank, which is in contrast to our method. In addition, although we derived the algorithm based on the discussion using the dually-flat manifold on (θ, η)-coordinate, we do not have to know the value of (θ, η) during the algorithm, which also make a difference from related works in [116, 117].

3.10 Connection between Rank-1 Approximation and Balancing

So far, we have seen that the rank of tensors can be reduced by describing the low-rank condition with the many-body θ-parameters. Related to this task, it has been reported that characterizing tensors by one-body η-parameters enables an operation called tensor balancing [117], which is often solved by Sinkhorn algorithm [111] and its quantum information geometry is also studied [85]. Therefore, here we summarize the relationship between the rank-1 approximation, which constrains many-body θ-parameters, and tensor balancing, which constrains one-body η-parameters.

First, we introduce tensor balancing for a non-negative tensor $\mathcal{P} \in \mathbb{R}_{\geq 0}^{I_1 \times \cdots \times I_D}$. There are two kinds of balancing, fiber balancing and slice balancing.

3.10.1 Slice Balancing and Rank-1 Approximation

Given D vectors $\boldsymbol{c}^{(k)} \in \mathbb{R}^{I_k}_{\geq 0}$ for $k \in [D]$, the task of $\boldsymbol{c}$-slice balancing is to rescale a tensor so that the sum of each slice satisfies

$$\sum_{i_1=1}^{I_1} \cdots \sum_{i_{k-1}=1}^{I_{k-1}} \sum_{i_{k+1}=1}^{I_{k+1}} \cdots \sum_{i_D=1}^{I_D} \mathcal{P}_{i_1,\dots,i_D} = c^{(k)}_{i_k} \tag{3.19}$$

for all $i_k \in [I_k]$. Note that there is no solution when $\sum_i c^{(k)}_i \neq \sum_i c^{(l)}_i$, hence we always assume that $\sum_i c^{(k)}_i = 1$ for any $k \in [D]$ without loss of generality. The information geometric formulation of tensor balancing has already been performed in [117]. Recall the definition of expectation parameters, the above condition for $\boldsymbol{c}$-slice balancing can be expressed by one-body η-parameters of the input tensor as follows:

Proposition 3.7 $\boldsymbol{c}$-balancing Condition (η-representation) [117]

A given tensor is $\boldsymbol{c}$-balanced if and only if its one-body η-parameters satisfy

$$\eta^{(k)}_{i_k} = \sum_{i=i_k}^{I_k} c^{(k)}_i . \tag{3.20}$$

Let us define $\boldsymbol{c}$-slice balancing space $\mathcal{Q}_{\boldsymbol{c}}$ as the set of $\boldsymbol{c}$-slice balanced tensor, yielding $\mathcal{Q}_{\boldsymbol{c}} = \{p_\eta \mid \eta \text{ satisfies the condition (3.20) }\}$. By considering tensor balancing and rank-1 approximation simultaneously in the framework of information geometry, we can derive the following property: the $\boldsymbol{c}$-balanced rank-1 tensor always uniquely exists. As discussed in Section 2.1, we can identify a distribution using the mixture coordinate system (θ, η) that combines θ- and η-coordinates. On the intersection $\mathcal{Q}_{\boldsymbol{c}} \cap \mathcal{B}_1$, all one-body parameters are identified by $\boldsymbol{c}$-balancing condition in Equation (3.20) and other parameters are identified by rank-1 condition in Proposition 3.1. Now, balancing conditions and rank-1 condition specify all parameters; therefore, the mixture coordinate (θ, η) uniquely identifies the rank-1 $\boldsymbol{c}$-balanced tensor.

Theorem 3.2 $\mathcal{Q}_{\boldsymbol{c}} \cap \mathcal{B}_1$ is a singleton.

The intersection $\mathcal{Q}_{\boldsymbol{c}} \cap \mathcal{B}_1$ is a singleton.

Proof: As we discussed in Section 2.1, we can identify a distribution using the mixture coordinate system (θ, η) that combines θ- and η-coordinates. Therefore, specifying $I_1 \times \cdots \times I_D$ parameters on the mixture coordinate (θ, η) uniquely identifies a tensor $\mathcal{P} \in \mathbb{R}^{I_1 \times \cdots \times I_D}_{>0}$. There are two kinds of parameters, one-body parameters and many-body paramters. On the intersection $\mathcal{Q}_{\boldsymbol{c}} \cap \mathcal{B}_1$, the $\boldsymbol{c}$-balancing condition in Equation (3.20) determines all one-body η-parameters $\eta^{(k)}_{i_k}$ for all $i_k \in [I_k]$ and $k \in [D]$. On the other hand, the rank-1 condition in Proposition 3.1 determines all many-body θ-parameters $\theta_{i_1,\dots,i_D} = 0$. Now, the $\boldsymbol{c}$-balancing conditions and the rank-1 condition specify all $I_1 \times \cdots \times I_D$ parameters (See Figure 3.7(a)), therefore the mixture coordinate (θ, η) uniquely identifies the rank-1 $\boldsymbol{c}$-balanced tensor. □

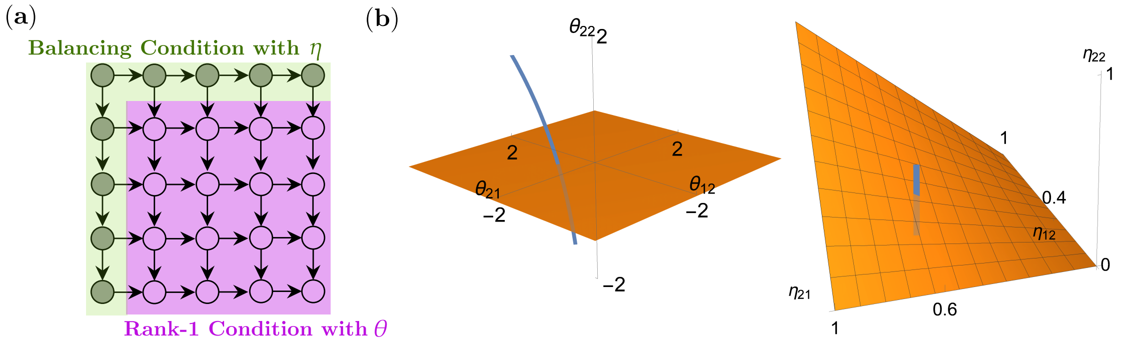


Figure 3.7 (a) All parameters are uniquely determined when the rank-1 and balancing conditions are imposed simultaneously. (b) Balancing subspace $\mathcal{Q}_{\boldsymbol{c}}$ (blue) and rank-1 subspace (orange) $\mathcal{B}_1$ in θ space (left) and η space (right) with $I_1 = I_2 = 2$ and $\boldsymbol{c}^{(1)} = \boldsymbol{c}^{(2)} = (0.4, 0.6)$.

To get the intuition of geometric structure across conditions on rank-1 approximation and balancing, we see a simple case of $I_1 = I_2 = 2$ and $d = 2$. We illustrate a simple case of $d = 2$ as 3D plots in Figure 3.7. Let us consider the $\boldsymbol{c}$-balanced matrix $\mathbf{P} \in \mathbb{R}^{2\times 2}_{\geq 0}$ with $n = 2$. We obtain the coordinate of $\mathbf{P}$ using $\mathbf{P}_{22}$:

$$\theta(\mathbf{P}) = \begin{bmatrix} \log\left(1 - \boldsymbol{c}_2^{(1)} - \boldsymbol{c}_2^{(2)} + \mathbf{P}_{22}\right) & \log \frac{\boldsymbol{c}_2^{(2)} - \mathbf{P}_{22}}{(1 - \boldsymbol{c}_2^{(1)} - \boldsymbol{c}_2^{(2)} + \mathbf{P}_{22})} \\ \log \frac{\boldsymbol{c}_2^{(1)} - \mathbf{P}_{22}}{(1 - \boldsymbol{c}_2^{(1)} - \boldsymbol{c}_2^{(2)} + \mathbf{P}_{22})} & \log \frac{\mathbf{P}_{22}(1 - \boldsymbol{c}_2^{(1)} - \boldsymbol{c}_2^{(2)} + \mathbf{P}_{22})}{(\boldsymbol{c}_2^{(2)} - \mathbf{P}_{22})(\boldsymbol{c}_2^{(1)} - \mathbf{P}_{22})} \end{bmatrix},$$

$$\eta(\mathbf{P}) = \begin{bmatrix} 1 & \boldsymbol{c}_2^{(2)} \\ \boldsymbol{c}_2^{(1)} & \mathbf{P}_{22} \end{bmatrix}.$$

Remember that θ_{11} corresponds to the normalizing factor and $\eta_{11} = 1$. The subspace consisting of $\boldsymbol{c}$-balanced matrices can be drawn as a convex curve in a 3-dimensional space by regarding $\mathbf{P}_{22}$ as a mediator variable. The curve becomes a straight line in the θ-coordinate only when $\boldsymbol{c}^{(1)} = \boldsymbol{c}^{(2)} = (0.5, 0.5)$. In contrast, the set of rank-1 matrices is identified as a plane $(\theta_{21}, \theta_{12}, 0)$ in the θ-coordinate since $\theta_{22} = 0$ ensures $\mathrm{rank}(\mathbf{P}) = 1$ and on the plane $(\eta_{21}, \eta_{12}, \eta_{21}\eta_{12})$ in the η space (See Propositions 3.1 and 3.2). We can observe that $\boldsymbol{c}$-slice balancing space $\mathcal{Q}_{\boldsymbol{c}}$ and mean-field space $\mathcal{B}_1$ cross a point, which is shown in Figure 3.7(b). It is coherent with Theorem 3.2. The cross point dynamically changes by $\boldsymbol{c}^{(1)}$ and $\boldsymbol{c}^{(2)}$. In addition, Figure 3.7(b) shows that we obtain the same matrix by rank-1 approximation of any matrix on $\mathcal{Q}_{\boldsymbol{c}}$ since rank-1 approximation does not change values of one-body η-parameters.

3.10.2 Fiber Balancing and Rank-1 Approximation

Given D tensors $\mathcal{C}^{\backslash k} \in \mathbb{R}_{\geq 0}^{I_1 \times \cdots \times I_{k-1} \times I_{k+1} \times \cdots \times I_D}$ for $k \in [D]$, the task of $\mathcal{C}$-fiber balancing is to rescale a tensor so that the sum of each fiber satisfies

$$\sum_{i_k=1}^{I_k} \mathcal{P}_{i_1,\dots,i_D} = \mathcal{C}^{\backslash k}_{i_1,\dots,i_{k-1},i_{k+1},\dots,i_D} \tag{3.21}$$

for $i_k \in [I_k]$. Recall the definition of η-parameters, the above condition for $\mathcal{C}$-fiber balancing can be expressed as follows:

Proposition 3.8 $\mathcal{C}$-balancing Condition (η-representation) [117]

A given tensor is $\mathcal{C}$-balanced if and only if its $D-1$-body η-parameters satisfy

$$\eta_{i_1,\dots,i_{k-1},1,i_{k+1},\dots,i_D} = \sum_{i'_1=i_1}^{I_1} \cdots \sum_{i'_{k-1}=i_{k-1}}^{I_{k-1}} \sum_{i'_{k+1}=i_{k+1}}^{I_{k+1}} \cdots \sum_{i'_D=i_D}^{I_D} \mathcal{C}^{\backslash k}_{i'_1,\dots,i'_{k-1},i'_{k+1},\dots,i'_D}. \tag{3.22}$$

Let us define $\mathcal{C}$-fiber balancing space $\mathcal{Q}_{\mathcal{C}}$ as the set of $\mathcal{C}$-fiber balanced tensor, yielding $\mathcal{Q}_{\mathcal{C}} = \{p_\eta \mid \eta \text{ satisfies the condition } (3.22)\ \}$. Note that fiber balancing is equivalent to slice balancing if $D = 2$, which is the case called matrix balancing.

If we impose the $\mathcal{C}$-fiber balancing condition and rank-1 condition on a tensor $\mathcal{P}$ simultaneously, then any parameter that contains only one 1 in its index has been imposed both the rank-1 and $\mathcal{C}$-balancing conditions. We can prove the following theorem by examining when these two conditions stand together.

Theorem 3.3 Singleton Condition for Rank-1 Fiber-balanced Tensor

The intersection $\mathcal{Q}_{\mathcal{C}} \cap \mathcal{B}_1$ exists and is a singleton if and only if

$$\mathcal{C}^{\backslash k}_{i_1,\dots,i_{k-1},i_{k+1},\dots,i_D} = \prod_{m \in [D] \backslash k} \left(\prod_{l \in [D] \backslash \{k,m\}} \sum_{i_l=1}^{I_l} \mathcal{C}^{\backslash k}_{i_1,\dots,i_{k-1},i_{k+1},\dots,i_D} \right).$$

Proof: A tensor $\mathcal{P} \in \mathbb{R}_{>0}^{I_1 \times \cdots \times I_D}$ on the subspace $\mathcal{Q}_{\mathcal{C}}$ satisfies $\mathcal{C}$-balancing condition that can be expressed by η-parameters as

$$\eta_{i_1,\dots,i_{k-1},1,i_{k+1},\dots,i_D} = \sum_{i'_1=i_1}^{I_1} \cdots \sum_{i'_{k-1}=i_{k-1}}^{I_{k-1}} \sum_{i'_{k+1}=i_{k+1}}^{I_{k+1}} \cdots \sum_{i'_D=i_D}^{I_D} \mathcal{C}^{\backslash k}_{i'_1,\dots,i'_{k-1},i'_{k+1},\dots,i'_D}. \tag{3.23}$$

However, in the intersection $\mathcal{Q}_{\mathcal{C}} \cap \mathcal{B}_1$, these parameters $\eta_{i_1,\dots,i_{k-1},1,i_{k+1},\dots,i_D}$ have to be satisfied rank-1 condition. Remind the necessary and sufficient conditions of a tensor to be rank-1 tensor in Equation (3.8), if and only if the condition $\mathcal{C}$ satisfies the following relation, the tensor $\mathcal{P}$ can be on $\mathcal{Q}_{\mathcal{C}} \cap \mathcal{B}_1$.

$$
\begin{aligned}
&\mathcal{C}^{\backslash k}_{i_1,\dots,i_{k-1},i_{k+1},\dots,i_D} \\
&\overset{(3.21)}{=} \sum_{i'_k=1}^{I_k} \mathcal{P}_{i_1,\dots,i_D} \\
&\overset{(3.7)}{=} \sum_{(i'_1,\dots,i'_{k-1},1,i'_{k+1},\dots,i'_D)\in\Omega_D} \mu^{i'_1,\dots,i'_{k-1},1,i'_{k+1},\dots,i'_D}_{i_1,\dots,i_{k-1},1,i_{k+1},\dots,i_D} \eta_{i'_1,\dots,i'_{k-1},1,i'_{k+1},\dots,i'_D} \\
&\overset{(3.8)}{=} \sum_{(i'_1,\dots,i'_{k-1},1,i'_{k+1},\dots,i'_D)\in\Omega_D} \mu^{i'_1,\dots,i'_{k-1},1,i'_{k+1},\dots,i'_D}_{i_1,\dots,i_{k-1},1,i_{k+1},\dots,i_D} \left(\prod_{m\in[D]\backslash k} \eta^{(m)}_{i'_m} \right) \\
&= \sum_{(i'_1,\dots,i'_{k-1},1,i'_{k+1},\dots,i'_D)\in\Omega_D} \left(\prod_{m\in[D]\backslash k} \mu^{i'_m}_{i_m} \eta^{(m)}_{i'_m} \right) \\
&= \prod_{m\in[D]\backslash k} \left(\eta^{(m)}_{i_m} - \eta^{(m)}_{i_{m+1}} \right) \\
&\overset{(3.6)}{=} \prod_{m\in[D]\backslash k} \left(\sum_{i'_1=1}^{I_1} \cdots \sum_{i'_{k-1}=1}^{I_{k-1}} \sum_{i'_{k+1}=1}^{I_{k+1}} \cdots \sum_{i'_D=1}^{I_D} \mathcal{P}_{i'_1,\dots,i'_{m-1},i_m,i'_{m+1},\dots,i_D} \right). \\
&\overset{(3.21)}{=} \prod_{m\in[D]\backslash k} \left(\prod_{l\in[D]\backslash\{k,m\}} \sum_{i_l=1}^{I_l} \mathcal{C}^{\backslash k}_{i_1,\dots,i_{k-1},i_{k+1},\dots,i_D} \right)
\end{aligned}
$$

Thus, the theorem was proved. □

3.11 Experiments for LTR

We compared LTR with two existing non-negative low-Tucker-rank approximation methods. The first method is non-negative Tucker decomposition, which is the standard non-negative tensor decomposition method [130] whose cost function is either the Least Squares (LS) error (NTD_LS) or the KL divergence (NTD_KL). The second method is sequential non-negative Tucker decomposition (lraSNTD), which is known as the faster of the two methods [141]. Its cost function is the LS error.

Implementation Details All methods are implemented in `Julia` 1.6 with `TensorToolbox`[2] library [100], hence runtime comparison is fair. We implement lraSNTD referring to the the original papers [141]. We used the `TensorLy` implementation [69] for NTDs. Experiments were conducted on CentOS 6.10 with a single core of 2.2 GHz Intel Xeon CPU E7-8880 v4 and 3TB of memory. We use default values of hyper-parameters of `tensorly` [69] for NTD. We use default values of hyper-parameters of `sklearn` [99] for NMF module in lraSNTD.

[2] MIT Expat License

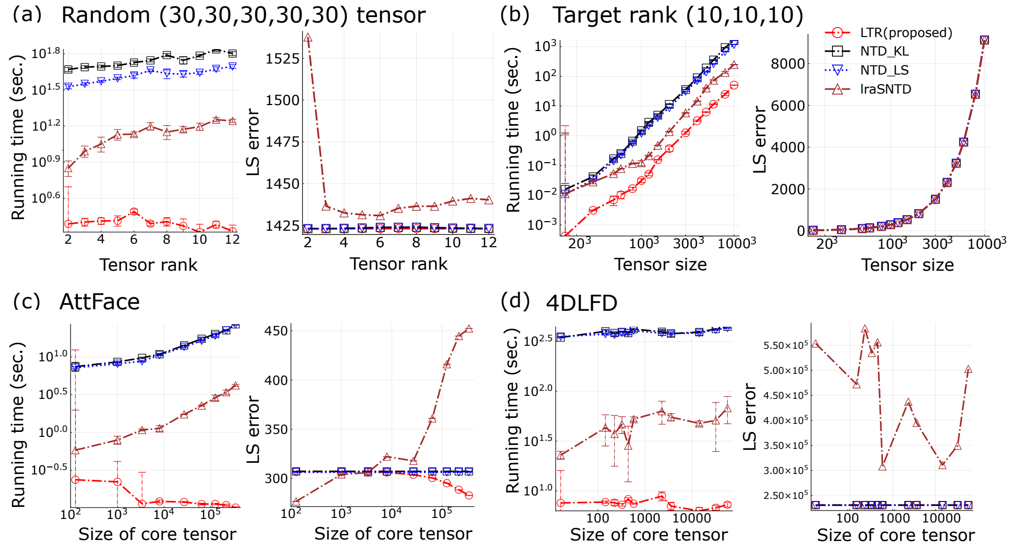


Figure 3.8 Experimental results for synthetic (a, b) and real-world (c, d) datasets. Mean errors ± standard error for 20 times iterations are plotted. (a) The horizontal axis is r for target Tucker rank (r, r, r, r, r). (b) The horizontal axis is n^3 for input (n, n, n) tensor. (c, d) The horizontal axis is the number of elements of the core tensor.

3.11.1 Results on Synthetic Data

We created tensors with $D = 3$ or $D = 5$, where every $I_k = n$. We change n to generate various sizes of tensors. Each element is sampled from the uniform continuous distribution on from 0 to 1. To evaluate the efficiency, we measured the running time of each method. To evaluate the accuracy, we measured the LS reconstruction error, defined as the Frobenius norm between input and output tensors. Figure 3.8(a) shows the running time and the LS reconstruction error for randomly generated tensors with $D = 3$ and $n = 30$ varying the target Tucker tensor rank. Figure 3.8(b) shows the running time and the LS reconstruction error for the target Tucker rank $(10, 10, 10)$ with varying the input tensor size n. These plots clearly show that our method is faster than other methods while keeping the competitive approximation accuracy.

3.11.2 Results on Real Data

We evaluated running time and the LS reconstruction error for two real-world datasets. 4DLFD is a $(9, 9, 512, 512, 3)$ tensor [57] and AttFace is a $(92, 112, 400)$ tensor [107]. AttFace is commonly used in tensor decomposition experiments [63, 65, 141]. For the 4DLFD dataset, we chose the target Tucker rank as (1,1,1,1,1), (2,2,2,2,1), (3,3,4,4,1), (3,3,5,5,1), (3,3,6,6,1), (3,3,7,7,1), (3,3,8,8,1), (3,3,16,16,1), (3,3,20,20,1), (3,3,40,40,1), (3,3,60,60,1), and (3,3,80,80,1). For the AttFace dataset, we chose (1,1,1), (3,3,3), (5,5,5), (10,10,10), (15,15,15), (20,20,20), (30,30,30), (40,40,40), (50,50,50), (60,60,60), (70,70,70), and (80,80,80). In both datasets, LTR is always faster than the comparison

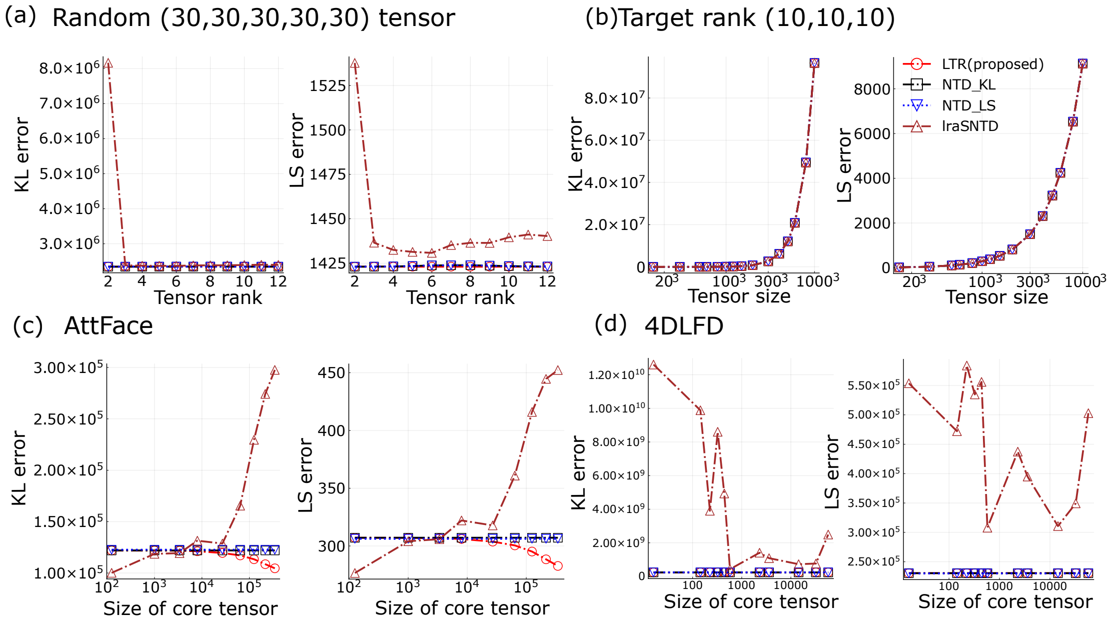


Figure 3.9 Experimental results for synthetic (a, b) and real-world (c, d) datasets. The left-hand panels are KL reconstruction error and the right-hand panels are LS reconstruction error. (a) The horizontal axis is r for target tensor rank (r, r, r, r, r). (b) The horizontal axis is n^3 for input (n, n, n) tensor.

methods, as shown in Figure 3.8(c, d), with competitive or better approximation accuracy in terms of the LS error.

We also obtained almost the same results as in Figure 3.8 with the KL reconstruction error in Figure 3.9 and we provided the experimental results as a tabular format in Table 3.2 and 3.3.

As described in Section 3.7, the search space of LTR is smaller than that of NTD and lraSNTD. Nevertheless, our experiments show that the approximation accuracy of LTR is competitive with other methods. This means that NTD and lraSNTD do not effectively treat linear combinations of bases.

Datasets details We describe the details of each dataset in the following. 4DLFD is a $(9, 9, 512, 512, 3)$ tensor, which is produced by 4D Light Field Dataset described in [57]. Its license is Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. We use dino images and their depth and disparity map in training scenes. We concatenate them to produce a tensor. AttFace is a $(92, 112, 400)$ tensor, which is produced by the entire data in The Database of Faces (AT&T) [107], which includes 400 grey-scale face photos. The size of each image is $(92, 112)$. AttFace is public on kaggle but the license is not specified.

3.12 Conclusion

We have derived a new probabilistic perspective to rank-1 approximation for tensors using information geometry and shown that is can be viewed as mean-field approximation. Our new geometric understanding leads to a novel fast non-negative low-Tucker-rank approximation method, called LTR, which does not use any gradient method. Our research will not only lead to applications of faster tensor decomposition but also can be a milestone of the research of tensor decomposition to further development of interdisciplinary field around information geometry and the mean-field theory.

Table 3.2 Experimental results of LTR on AttFace dataset.

	Relative running time			Relative LS error			Relative KL error		
Rank	NTD_KL	NTD_LS	lraSNTD	NTD_KL	NTD_LS	lraSNTD	NTD_KL	NTD_LS	lraSNTD
(1,1,1)	4.7533	3.8874	0.5510	1.0000	0.9971	0.9971	1.0000	1.0056	1.0059
(3,3,3)	33.4752	26.3030	2.7253	0.9755	0.9743	0.9280	0.9440	0.9549	0.8685
(5,5,5)	38.2118	29.9251	4.0683	0.9777	0.9784	0.8757	0.9508	0.9633	0.7786
(10,10,10)	28.6684	22.7941	4.4291	0.9826	0.9834	0.9915	0.9621	0.9736	0.9795
(15,15,15)	34.6129	27.5794	6.2854	0.9865	0.9867	1.0090	0.9714	0.9819	1.0006
(20,20,20)	32.0121	25.9077	6.6242	0.9916	0.9912	1.0548	0.9817	0.9913	1.0890
(30,30,30)	43.9135	36.6030	13.2826	1.0007	0.9996	1.0710	0.9996	1.0082	1.1222
(40,40,40)	45.2404	38.8275	15.1266	1.0164	1.0149	1.2127	1.0296	1.0378	1.4427
(50,50,50)	62.7553	52.3623	25.9060	1.0385	1.0367	1.3627	1.0707	1.0786	1.8738
(60,60,60)	74.8940	60.5661	32.1785	1.0617	1.0595	1.5309	1.1150	1.1228	2.5023
(70,70,70)	41.5551	35.3710	21.7593	1.0888	1.0863	1.6287	1.1676	1.1755	2.9325
(80,80,80)	58.4794	47.3544	30.3658	1.1186	1.1160	1.5663	1.2278	1.2358	2.6274

Table 3.3 Experimental results of LTR on 4DLFD dataset.

	Relative running time			Relative LS error			Relative KL error		
Rank	NTD_KL	NTD_LS	lraSNTD	NTD_KL	NTD_LS	lraSNTD	NTD_KL	NTD_LS	lraSNTD
(1,1,1,1,1)	13.3018	13.3018	1.0019	1.0000	0.9985	0.9985	1.0000	1.0027	1.0027
(2,2,2,2,1)	22.9786	22.9786	2.8540	0.9870	0.9842	3.0129	0.9748	0.9766	14.3603
(3,3,4,4,1)	22.6299	22.6299	8.2270	0.9319	0.9210	1.1570	0.8765	0.8682	1.3131
(3,3,5,5,1)	22.9552	22.9552	4.9963	0.9303	0.9201	3.2399	0.8745	0.8672	28.4056
(3,3,6,6,1)	22.4911	22.4911	7.0107	0.9279	0.9183	2.6687	0.8704	0.8642	19.1505
(3,3,7,7,1)	23.7567	23.7567	6.2986	0.9294	0.9202	1.0066	0.8730	0.8674	1.0927
(3,3,8,8,1)	23.7998	23.7998	9.0809	0.9284	0.9198	1.3867	0.8710	0.8663	1.9389
(3,3,16,16,1)	23.6797	23.6797	7.1864	0.9256	0.9211	1.1852	0.8640	0.8659	1.5274
(3,3,20,20,1)	26.1782	26.1782	6.4697	0.9304	0.9265	1.4345	0.8719	0.8746	2.0486
(3,3,40,40,1)	26.4613	26.4613	4.6566	0.9426	0.9414	1.5224	0.8933	0.8996	2.2700
(3,3,60,60,1)	27.9254	27.9254	10.5136	0.9532	0.9528	1.3116	0.9128	0.9197	1.7012
(3,3,80,80,1)	26.4166	26.4166	8.1281	0.9619	0.9615	1.2247	0.9288	0.9354	1.8635

4 Fast Rank-1 NMF for Missing Data with KL Divergence

We propose a fast non-gradient-based method of rank-1 non-negative matrix factorization (NMF) for missing data, called A1GM, that minimizes the KL divergence from an input matrix to the reconstructed rank-1 matrix. Our method is based on our new finding of an analytical closed-formula of the best rank-1 non-negative multiple matrix factorization (NMMF), a variety of NMF. NMMF is known to exactly solve NMF for missing data if positions of missing values satisfy a certain condition, and A1GM transforms a given matrix so that the analytical solution to NMMF can be applied. We empirically show that A1GM is more efficient than a gradient method with competitive reconstruction errors.

The tabular form is one of the most common data types across different fields, and includes purchasing data, processed sensor data, and images. Tabular datasets are often treated as matrices for analysis. To date, many methods have been developed to extract essential information from matrices [124, 131, 25, 12]. Non-negative matrix factorization (NMF) is one of the most popular techniques [74]. NMF extracts factors in a dataset by decomposing a given non-negative matrix $\mathbf{X} \in \mathbb{R}_{\geq 0}^{I \times J}$ into a product $\mathbf{AB}$ of two matrices $\mathbf{A} \in \mathbb{R}_{\geq 0}^{I \times r}$ and $\mathbf{B} \in \mathbb{R}_{\geq 0}^{r \times J}$ so that the predetermined cost becomes smaller. The matrix rank of the product $\mathbf{AB}$ is less than or equal to the hyper-parameter $r \in \mathbb{N}$.

Standard NMF, which uses the least squares error $\|\mathbf{X} - \mathbf{AB}\|_{\mathrm{F}}$ as the cost function, is widely used in various applications, including face recognition [102], recommender systems [120], and text analysis [133]. Although it is an NP-hard problem to find the best decomposition that minimizes the cost function exactly for any $r > 1$ [128], in an exceptional case of $r = 1$, the best decomposition is obtained in polynomial time [46]. Therefore it is possible to obtain the best decomposition in a reasonable time if $r = 1$ and a number of NMF algorithms are developed based on rank-1 NMF [9, 78]. Rank-1 NMF approximates an input matrix $\mathbf{X}$ by the Kronecker product $\boldsymbol{a} \otimes \boldsymbol{b}$ of two non-negative vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, called dominant factors. Since the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ correspond to the largest principal components restricted to the first quadrant in eigenvalue decomposition, they are representative features that roughly describe the input matrix.

However, it is problematic if a matrix includes missing values, which often occurs in real-world datasets in practice. When the cost function is the least squares error, NMF for missing data — that is, a matrix with missing values, which we call *missing NMF* [64] —

is known to become an NP-hard problem, even for $r = 1$ [44]. Although this is a crucial drawback of missing NMF, missing NMF with other cost functions has not been well studied to date. In this chapter, we show that there are certain cases in which rank-1 missing NMF can be exactly solved in polynomial time when the cost function is defined as the KL divergence.

Our key idea is that, instead of directly solving missing NMF, we focus on non-negative multiple matrix factorization (NMMF) [121], a variant of NMF. We derive a closed formula that globally minimizes the cost function of NMMF when the target rank $r = 1$, which is our main theoretical contribution. NMMF conducts simultaneous factor-sharing decomposition of multiple matrices, which has been used in purchase forecast systems [67] and recommender systems [137]. Interestingly, if the cost function is given as the KL divergence, NMMF is equivalent to missing NMF when missing values are clustered in the lower right corner of the input matrix [121]. Using this relationship between missing NMF and NMMF, we can efficiently compute the exact solution of rank-1 missing NMF with the KL divergence by our solution to NMMF when we can locate missing values in a rectangular region by permuting rows and columns.

Moreover, to treat matrices in which we cannot locate missing values in a rectangular region by permutations of rows and columns, we provide a method of finding an approximate solution of rank-1 missing NMF by adding more missing values so that we can use the closed formula of the best rank-1 NMMF. We call our novel method *A1GM* (Analytical solution for rank-1 NMF with Grid-based Missing values). We empirically show that A1GM is more efficient than an existing gradient-based method for rank-1 missing NMF with the competitive reconstruction error.

We summarize our contribution as follows:

- We derive a closed formula of the best rank-1 NMMF, which extracts the most dominant factors faster than the existing gradient method.
- We develop a novel efficient method to solve rank-1 missing NMF, called A1GM, and prove that A1GM globally minimizes a cost function under an assumption about the position of missing values.
- We empirically show that A1GM is more efficient than an existing method for missing NMF with competitive reconstruction error.

First, we define the rank-1 NMMF in Section 4.1. Then, we provide the best rank-1 approximation formula in Section 4.2. In subsections 4.3 – 4.4, we introduce information geometric formulation of NMMF using the log-linear model and derive the closed-form solution. Finally, we introduce A1GM using the closed formula in Section 4.5.

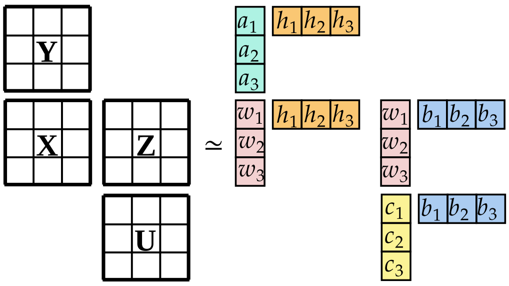


Figure 4.1 A sketch of Rank-1 NMMF with four inputs matrices for $I = J = N = M = L = 3$. The task approximates four input matrices with shared factors.

Notations Matrices are denoted by bold capital letters like $\mathbf{X}$ and $\mathbf{Y}$, and vectors are denoted by lower-case bold alphabets like $\boldsymbol{a}$ and $\boldsymbol{b}$. The total sum of a matrix or a vector is represented as $S(\cdot)$. The ith component of a vector $\boldsymbol{a}$ is written in its non-bold letter as a_i. The Kronecker product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is denoted by $(\boldsymbol{a} \otimes \boldsymbol{b})$, which is a rank-1 matrix, and each element is defined as $(\boldsymbol{a} \otimes \boldsymbol{b})_{ij} = a_i b_j$. The $I \times J$ all-one and all-zero matrices are denoted by $\mathbf{1}_{IJ}$ and $\mathbf{0}_{IJ}$, respectively. The identity matrix is denoted by $\mathbf{I}$. The transpose of a matrix $\mathbf{X}$ is denoted by $\mathbf{X}^\top$. The element-wise product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A} \circ \mathbf{B}$. For a pair of natural numbers n and m $(\geq n)$, we denote by $[n, m] = \{n, n+1, \dots, m-1, m\}$. We abbreviate $[1, m]$ as $[m]$. The set difference of B and A is denoted by $B \setminus A$. When we use the Kullback–Leibler (KL) divergence $D(\mathbf{X}, \mathbf{Y})$ for matrices $\mathbf{X}$ and $\mathbf{Y}$, it is defined as follows: [73]

$$D(\mathbf{X}, \mathbf{Y}) = \sum_{i,j} \left\{ \mathbf{X}_{ij} \log \frac{\mathbf{X}_{ij}}{\mathbf{Y}_{ij}} - \mathbf{X}_{ij} + \mathbf{Y}_{ij} \right\}.$$

In this chapter, we treat two tasks, rank-1 NMMF and rank-1 missing NMF. We consistently assume that these cost functions are defined by the above KL divergence throughout the chapter.

4.1 Rank-1 Non-negative Multiple Matrix Factorization

We provide the definition of the rank-1 NMMF as follows:

Task 4.1 Rank-1 Non-negative Multiple Matrix Factorization (Rank-1 NMMF)

Rank-1 NMMF simultaneously decomposes four matrices $\mathbf{X} \in \mathbb{R}^{I\times J}_{\geq 0}$, $\mathbf{Y} \in \mathbb{R}^{N\times J}_{\geq 0}$, $\mathbf{Z} \in \mathbb{R}^{I\times M}_{\geq 0}$ and $\mathbf{U} \in \mathbb{R}^{L\times M}_{\geq 0}$ into four rank-1 matrices $\boldsymbol{w} \otimes \boldsymbol{h}$, $\boldsymbol{a} \otimes \boldsymbol{h}$, $\boldsymbol{w} \otimes \boldsymbol{b}$ and $\boldsymbol{c} \otimes \boldsymbol{b}$, respectively, using non-negative vectors $\boldsymbol{w} \in \mathbb{R}^{I}_{\geq 0}$, $\boldsymbol{h} \in \mathbb{R}^{J}_{\geq 0}$, $\boldsymbol{a} \in \mathbb{R}^{N}_{\geq 0}$, $\boldsymbol{b} \in \mathbb{R}^{M}_{\geq 0}$ and $\boldsymbol{c} \in \mathbb{R}^{L}_{\geq 0}$. The cost function of NMMF is defined as

$$D(\mathbf{X}, \boldsymbol{w} \otimes \boldsymbol{h}) + \alpha D(\mathbf{Y}, \boldsymbol{a} \otimes \boldsymbol{h}) + \beta D(\mathbf{Z}, \boldsymbol{w} \otimes \boldsymbol{b}) + \gamma D(\mathbf{U}, \boldsymbol{c} \otimes \boldsymbol{b}), \tag{4.1}$$

and the task of rank-1 NMMF is to find vectors $\boldsymbol{w}$, $\boldsymbol{h}$, $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ that minimize the above cost.

We assume that the scaling parameters α, β and γ are non-negative real numbers. We provide a sketch of the task in Figure 4.1.

4.2 A Closed Formula of the Best Rank-1 NMMF

We give the following closed-form of the best rank-1 NMMF that exactly minimizes the cost function in Equation (4.1), which is one of our main theoretical contributions. This formula efficiently extracts only the most dominant shared factors from four input matrices. While the standard NMMF requires only three input matrices, we analyze the extended-NMMF that requires four input matrices. This enables us to exactly solve the missing NMF with a rank-2 weighted matrix as shown in Section 4.5.3.

Theorem 4.1 Closed Formula for Best Rank-1 NMMF

For any four positive matrices $\mathbf{X} \in \mathbb{R}^{I\times J}_{>0}$, $\mathbf{Y} \in \mathbb{R}^{N\times J}_{>0}$, $\mathbf{Z} \in \mathbb{R}^{I\times M}_{>0}$, and $\mathbf{U} \in \mathbb{R}^{L\times M}_{>0}$ and three parameters $\alpha, \beta, \gamma \geq 0$, five non-negative vectors $\boldsymbol{w} \in \mathbb{R}^{I}_{\geq 0}$, $\boldsymbol{h} \in \mathbb{R}^{J}_{\geq 0}$, $\boldsymbol{a} \in \mathbb{R}^{N}_{\geq 0}$, $\boldsymbol{b} \in \mathbb{R}^{M}_{\geq 0}$, and $\boldsymbol{c} \in \mathbb{R}^{L}_{\geq 0}$ that minimize the cost function in Equation (4.1) is given as

$$w_i = \frac{\sqrt{S(\mathbf{X})}}{S(\mathbf{X}) + \beta S(\mathbf{Z})} \left(\sum_{j=1}^{J} \mathbf{X}_{ij} + \beta \sum_{m=1}^{M} \mathbf{Z}_{im} \right),$$

$$h_j = \frac{\sqrt{S(\mathbf{X})}}{S(\mathbf{X}) + \alpha S(\mathbf{Y})} \left(\sum_{i=1}^{I} \mathbf{X}_{ij} + \alpha \sum_{n=1}^{N} \mathbf{Y}_{nj} \right),$$

$$a_n = \frac{1}{\sqrt{S(\mathbf{X})}} \left(\sum_{j=1}^{J} \mathbf{Y}_{nj} \right), \quad c_l = \frac{\sqrt{S(\mathbf{X})}}{S(\mathbf{Z})} \left(\sum_{m=1}^{M} \mathbf{U}_{lm} \right),$$

$$b_m = \frac{S(\mathbf{Z})}{\beta S(\mathbf{Z}) + \gamma S(\mathbf{U})} \frac{1}{\sqrt{S(\mathbf{X})}} \left(\beta \sum_{i=1}^{I} \mathbf{Z}_{im} + \gamma \sum_{l=1}^{L} \mathbf{U}_{lm} \right).$$

We provide complete proof of Theorem 4.1 on page 52.

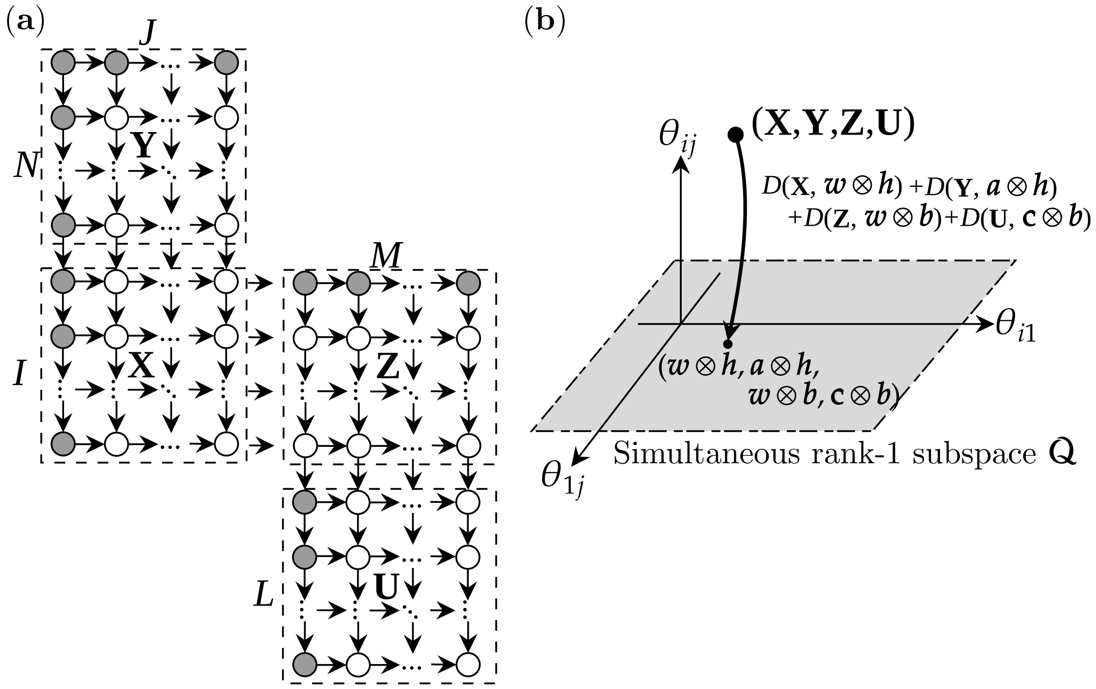

(a)
J
N
Y
M
I
X
Z
L
U
(b)
θ_{ij}
(X,Y,Z,U)
$D(\mathbf{X}, w \otimes h) + D(\mathbf{Y}, a \otimes h)$
$+D(\mathbf{Z}, w \otimes b) + D(\mathbf{U}, c \otimes b)$
θ_{i1}
$(w \otimes h, a \otimes h,$
$w \otimes b, c \otimes b)$
θ_{1j}
Simultaneous rank-1 subspace Q

Figure 4.2 (a) A partial order structure for NMMF for three input matrices $\mathbf{X} \in \mathbb{R}_{>0}^{I \times J}$, $\mathbf{Y} \in \mathbb{R}_{>0}^{N \times J}$, $\mathbf{Z} \in \mathbb{R}_{>0}^{I \times M}$ and $\mathbf{U} \in \mathbb{R}_{>0}^{L \times M}$. Only θ-parameters on gray-colored nodes can have non-zero values if and only if $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ is simultaneously rank-1 decomposable. (b) Information geometric view of rank-1 NMMF. Rank-1 NMMF is m-projection onto simultaneous rank-1 subspace from a tuple of input four matrices, where one-body η-parameters do not change.

The time complexity to obtain the best rank-1 NMMF is $O(IJ + NJ + IM + LM)$ because all we need is to take the summation of each column and row of the matrices $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ and $\mathbf{U}$. Note that, if $N = M = 0$, our result in Theorem 4.1 coincides with the best rank-1 NMF minimizing the KL divergence from an input matrix $\mathbf{X}$ shown in [56], which also coincides with the best rank-1 approximation provided in Equation (3.2) for $d = 2$.

4.3 Posets for NMMF

The input of NMMF is a tuple $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$, where $\mathbf{X} \in \mathbb{R}_{>0}^{I \times J}$, $\mathbf{Y} \in \mathbb{R}_{>0}^{N \times J}$, $\mathbf{Z} \in \mathbb{R}_{>0}^{I \times M}$, and $\mathbf{U} \in \mathbb{R}_{>0}^{L \times M}$. For simplicity, we normalize them beforehand so that their sum is 1; that is, $S(\mathbf{X}) + S(\mathbf{Y}) + S(\mathbf{Z}) + S(\mathbf{U}) = 1$. It is straightforward to eliminate this assumption using the property of the KL divergence, $\lambda D(\mathbf{X}, \mathbf{Y}) = D(\lambda \mathbf{X}, \lambda \mathbf{Y})$, for any non-negative number λ. We model these four matrices using a single discrete distribution on a partial ordered sample space.

To make a one-to-one mapping from $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ to a probability mass function p, we prepare the index set Ω as

$$\begin{aligned}\Omega &= \Omega_{\mathbf{X}} \cup \Omega_{\mathbf{Y}} \cup \Omega_{\mathbf{Z}} \cup \Omega_{\mathbf{U}}, \text{ where} \\ \Omega_{\mathbf{X}} &= [N+1, I+N] \times [J], \quad \Omega_{\mathbf{Y}} = [N] \times [J], \\ \Omega_{\mathbf{Z}} &= [N+1, I+N] \times [J+1, J+M], \\ \Omega_{\mathbf{U}} &= [N+I+1, N+I+L] \times [J+1, J+M],\end{aligned}$$

where the subspace $\Omega_{\mathbf{X}}$ corresponds to the index of $\mathbf{X}$, $\Omega_{\mathbf{Y}}$ to $\mathbf{Y}$, $\Omega_{\mathbf{Z}}$ to $\mathbf{Z}$, and $\Omega_{\mathbf{U}}$ to $\mathbf{U}$. Then, we define the following partial order "$\leq$" between each element (s,t) in the index set Ω

$$(s,t) \leq (s', t') \Leftrightarrow s \leq s' \text{ and } t \leq t'. \tag{4.2}$$

The smallest element in $(\Omega, \leq)$ is $\perp = (1,1)$. We regard the multiple matrices $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ as a distribution on the log-linear model on the poset $(\Omega, \leq)$.

$$p(s,t) = \exp\left(\sum_{(s',t') \leq (s,t)} \theta_{s't'} \right), \tag{4.3}$$

where $(s,t) \in \Omega$. The natural parameter θ_{11}, in which $(1,1)$ is the smallest element in the poset Ω, corresponds to the normalization factor. The θ-parameters $\{\theta_{21}, \dots, \theta_{N+I+L, J+M}\}$ are identified so that they satisfy

$$\begin{gathered}\mathbf{X}_{ij} = p(N+i, j), \ \ \mathbf{Y}_{nj} = p(n, j), \\ \mathbf{Z}_{im} = p(N+i, J+m), \ \ \mathbf{U}_{lm} = p(N+I+l, J+m),\end{gathered}$$

for $i \in [I]$, $j \in [J]$, $n \in [N]$,$m \in [M]$ and $l \in [L]$. Figure 4.2 illustrates the partial order for the input triple $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$.

There are other possible ways to model $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ as a probability distribution using different partial order structure. However, the solution formula that we obtain in Theorem 4.1 does not depend on the modeling.

Using results provided in [117] based on the incidence algebra between θ- and η-parameters, we can also obtain the η-parameters using the following formula:

$$\eta_{st} = \sum_{(s,t) \leq (s',t')} p(s', t').$$

To make the following discussion clear, for all $i \in [I], j \in [J], n \in [N], m \in [m]$ and $l \in [L]$, we define

$$\eta^{\mathbf{Y}}_{nj} = \eta_{nj}, \quad \eta^{\mathbf{X}}_{ij} = \eta_{N+i,j}, \quad \eta^{\mathbf{Z}}_{im} = \eta_{N+i,J+m}, \quad \eta^{\mathbf{U}}_{lm} = \eta_{N+I+l,J+m},$$
$$\theta^{\mathbf{Y}}_{nj} = \theta_{nj}, \quad \theta^{\mathbf{X}}_{ij} = \theta_{N+i,j}, \quad \theta^{\mathbf{Z}}_{im} = \theta_{N+i,J+m}, \quad \theta^{\mathbf{U}}_{lm} = \theta_{N+I+l,J+m}.$$

4.4 Derivation of the Exact Solution of Rank-1 NMMF

For simplicity, we define simultaneous rank 1 decomposability as follows:

Definition 4.1 Simultaneously Rank-1 Decomposable

Let $\boldsymbol{w} \in \mathbb{R}^{I}_{\geq 0}, \boldsymbol{h} \in \mathbb{R}^{J}_{\geq 0}, \boldsymbol{a} \in \mathbb{R}^{N}_{\geq 0}$, $\boldsymbol{b} \in \mathbb{R}^{M}_{\geq 0}$, and $\boldsymbol{c} \in \mathbb{R}^{L}_{\geq 0}$. If four positive matrices $\mathbf{X} \in \mathbb{R}^{I\times J}_{>0}$, $\mathbf{Y} \in \mathbb{R}^{N\times J}_{>0}$, $\mathbf{Z} \in \mathbb{R}^{I\times M}_{>0}$, and $\mathbf{U} \in \mathbb{R}^{L\times M}_{>0}$ can be decomposed into a form $\boldsymbol{w} \otimes \boldsymbol{h}, \boldsymbol{a} \otimes \boldsymbol{h}$, $\boldsymbol{w} \otimes \boldsymbol{b}$, and $\boldsymbol{c} \otimes \boldsymbol{b}$, we say that $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ is *simultaneously rank-1 decomposable*.

To describe the necessary and sufficient conditions for $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ to be simultaneously rank-1 decomposable, we define *one-body* parameters and *many-body* parameters as well as we define in Section 3.5. We call $\theta^{\mathbf{Y}}_{1j}, \theta^{\mathbf{Y}}_{n1}, \theta^{\mathbf{X}}_{i1}, \theta^{\mathbf{Z}}_{1m}, \theta^{\mathbf{U}}_{l1}$ as one-body θ-parameters and $\eta^{\mathbf{Y}}_{1j}, \eta^{\mathbf{Y}}_{n1}, \eta^{\mathbf{X}}_{i1}, \eta^{\mathbf{Z}}_{1m}, \eta^{\mathbf{U}}_{l1}$ as one-body η-parameters for any $i \in [I]$, $j \in [2, J]$, $n \in [2, N]$, $m \in [M]$ and $l \in [L]$. Gray-colored nodes in Figure 4.2(a) corresponds to one-body parameters. Parameters which are not one-body parameters are called many-body parameters. Using these parameters, we obtain the following proposition.

Proposition 4.1 Simultaneous Rank-1 θ-condition

A tuple $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ is simultaneously rank-1 decomposable if and only if its all many-body θ-parameters are 0.

Proof: First, we show a tuple $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ is simultaneously rank-1 decomposable $\Rightarrow$ its all many-body natural parameters are 0. If a tuple $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ is simultaneously rank-1 decomposable, we can immediately confirm that $\mathrm{Rank}\,(\mathbf{X}) = \mathrm{Rank}\,(\mathbf{Y}) = \mathrm{Rank}\,(\mathbf{Z}) = \mathrm{Rank}\,(\mathbf{U}) = 1$. Then, from the Proposition 3.1 for $d = 2$,

$$\begin{aligned}
\theta^{\mathbf{Y}}_{nj} &= 0 \quad \text{if } n \neq 1 \text{ and } j \neq 1,\\
\theta^{\mathbf{X}}_{ij} &= 0 \quad \text{if } i \neq 1 \text{ and } j \neq 1,\\
\theta^{\mathbf{Z}}_{im} &= 0 \quad \text{if } i \neq 1 \text{ and } m \neq 1,\\
\theta^{\mathbf{U}}_{lm} &= 0 \quad \text{if } l \neq 1 \text{ and } m \neq 1.
\end{aligned}$$

Then, from the definition of the model in Equation (4.3), it holds that

$$
\begin{aligned}
\mathbf{Y}_{nj} &= e^{\theta^{\mathbf{Y}}_{11}} \exp\left(\sum_{n'=2}^{n} \theta^{\mathbf{Y}}_{n'1}\right) \exp\left(\sum_{j'=2}^{j} \theta^{\mathbf{Y}}_{1j'}\right) \\
\mathbf{X}_{ij} &= e^{\theta^{\mathbf{X}}_{11}+\theta^{\mathbf{Y}}_{11}} \exp\left(\sum_{n'=2}^{N} \theta^{\mathbf{Y}}_{n'1} + \sum_{i'=2}^{i} \theta^{\mathbf{X}}_{i'1}\right) \exp\left(\sum_{j'=2}^{j} \theta^{\mathbf{Y}}_{1j'} + \sum_{j'=2}^{j} \theta^{\mathbf{X}}_{1j'}\right) \\
\mathbf{Z}_{im} &= e^{\theta^{\mathbf{X}}_{11}+\theta^{\mathbf{Y}}_{11}+\theta^{\mathbf{Z}}_{11}} \exp\left(\sum_{n'=2}^{N} \theta^{\mathbf{Y}}_{n'1} + \sum_{i'=2}^{i} \theta^{\mathbf{X}}_{i'1} + \sum_{i'=2}^{i} \theta^{\mathbf{Z}}_{i'1}\right) \\
&\quad \times \exp\left(\sum_{j'=2}^{J} \theta^{\mathbf{Y}}_{1j'} + \sum_{j'=2}^{J} \theta^{\mathbf{X}}_{1j'} + \sum_{m'=2}^{m} \theta^{\mathbf{Z}}_{1m'}\right) \\
\mathbf{U}_{lm} &= e^{\theta^{\mathbf{X}}_{11}+\theta^{\mathbf{Y}}_{11}+\theta^{\mathbf{Z}}_{11}+\theta^{\mathbf{U}}_{11}} \exp\left(\sum_{n'=2}^{N} \theta^{\mathbf{Y}}_{n'1} + \sum_{i'=2}^{I} \theta^{\mathbf{X}}_{i'1} + \sum_{i'=2}^{I} \theta^{\mathbf{Z}}_{i'1} + \sum_{l'=2}^{l} \theta^{\mathbf{U}}_{l'1}\right) \\
&\quad \times \exp\left(\sum_{j'=2}^{J} \theta^{\mathbf{Y}}_{1j'} + \sum_{j'=2}^{J} \theta^{\mathbf{X}}_{1j'} + \sum_{m'=2}^{m} \theta^{\mathbf{Z}}_{1m'} + \sum_{m'=2}^{m} \theta^{\mathbf{U}}_{1m'}\right)
\end{aligned}
$$

If it does not hold that

$$
\begin{aligned}
\theta^{\mathbf{X}}_{1j} &= 0 \quad \text{if } j \neq 1, \\
\theta^{\mathbf{Z}}_{i1} &= 0 \quad \text{if } i \neq 1, \\
\theta^{\mathbf{U}}_{1m} &= 0 \quad \text{if } m \neq 1,
\end{aligned}
$$

matrices $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ have never shared factors. Then, the above condition holds if the matrices $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ is simultaneously rank-1 decomposable.

Next, we show that its all many-body natural parameters are $0 \Rightarrow$ a tuple $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ is simultaneously rank-1 decomposable. We put all many-body θ-parameters as 0 in Equation (4.3) and obtain

$$
\begin{aligned}
\mathbf{Y}_{nj} &= e^{\theta^{\mathbf{Y}}_{11}} \exp\left(\sum_{n'=2}^{n} \theta^{\mathbf{Y}}_{n'1}\right) \exp\left(\sum_{j'=2}^{j} \theta^{\mathbf{Y}}_{1j'}\right), \\
\mathbf{X}_{ij} &= e^{\theta^{\mathbf{X}}_{11}+\theta^{\mathbf{Y}}_{11}} \exp\left(\sum_{n'=2}^{N} \theta^{\mathbf{Y}}_{n'1} + \sum_{i'=2}^{i} \theta^{\mathbf{X}}_{i'1}\right) \exp\left(\sum_{j'=2}^{j} \theta^{\mathbf{Y}}_{1j'}\right), \\
\mathbf{Z}_{im} &= e^{\theta^{\mathbf{X}}_{11}+\theta^{\mathbf{Y}}_{11}+\theta^{\mathbf{Z}}_{11}} \exp\left(\sum_{n'=2}^{N} \theta^{\mathbf{Y}}_{n'1} + \sum_{i'=2}^{i} \theta^{\mathbf{X}}_{i'1}\right) \exp\left(\sum_{j'=2}^{J} \theta^{\mathbf{Y}}_{1j'} + \sum_{m'=2}^{m} \theta^{\mathbf{Z}}_{1m'}\right), \\
\mathbf{U}_{lm} &= e^{\theta^{\mathbf{X}}_{11}+\theta^{\mathbf{Y}}_{11}+\theta^{\mathbf{Z}}_{11}+\theta^{\mathbf{U}}_{11}} \exp\left(\sum_{n'=2}^{N} \theta^{\mathbf{Y}}_{n'1} + \sum_{i'=2}^{I} \theta^{\mathbf{X}}_{i'1} + \sum_{l'=2}^{l} \theta^{\mathbf{U}}_{l'1}\right) \exp\left(\sum_{j'=2}^{J} \theta^{\mathbf{Y}}_{1j'} + \sum_{m'=2}^{m} \theta^{\mathbf{Z}}_{1m'}\right).
\end{aligned}
$$

Then, we can define the following shared factors on the tuple $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$,

$$\begin{cases} \boldsymbol{a}_n = \exp\left(\sum_{n'=2}^{n} \theta^{\mathbf{Y}}_{n'1}\right), \\ \boldsymbol{h}_j = e^{\theta^{\mathbf{Y}}_{11}} \exp\left(\sum_{j'=2}^{j} \theta^{\mathbf{Y}}_{1j'}\right), \\ \boldsymbol{w}_i = e^{\theta^{\mathbf{X}}_{11}} \exp\left(\sum_{n'=2}^{N} \theta^{\mathbf{Y}}_{n'1} + \sum_{i'=2}^{i} \theta^{\mathbf{X}}_{i'1}\right) \\ \boldsymbol{b}_m = e^{\theta^{\mathbf{Y}}_{11}+\theta^{\mathbf{Z}}_{11}} \exp\left(\sum_{j'=2}^{J} \theta^{\mathbf{Y}}_{1j'} + \sum_{m'=2}^{m} \theta^{\mathbf{Z}}_{1m'}\right), \\ \boldsymbol{c}_l = e^{\theta^{\mathbf{X}}_{11}+\theta^{\mathbf{U}}_{11}} \exp\left(\sum_{n'=2}^{N} \theta^{\mathbf{Y}}_{n'1} + \sum_{i'=2}^{I} \theta^{\mathbf{X}}_{i'1} + \sum_{l'=2}^{l} \theta^{\mathbf{U}}_{l'1}\right), \end{cases}$$

which satisify $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U}) = (\boldsymbol{w} \otimes \boldsymbol{h}, \boldsymbol{a} \otimes \boldsymbol{h}, \boldsymbol{w} \otimes \boldsymbol{b}, \boldsymbol{c} \otimes \boldsymbol{b})$. Thus, the proposition was proved. □

We call a subspace that satisfies simultaneous rank-1 condition *simultaneous rank*-1 *subspace*. From the viewpoint of information geometry, we can understand the best rank-1 NMMF as follows (shown in Figure 4.2(b)). The input of NMMF $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ corresponds to a point in the space described by the θ-coordinate system. The best rank-1 NMMF is an m-projection onto the simultaneous rank-1 subspace from the input point.

Since the m-projection is a convex optimization, we can get the projection destination by a gradient method. However, it requires appropriate settings for initial values, stopping criterion, and learning rates.

Our closed analytical formula of the projection destination in Theorem 4.1 solves all the drawbacks of the gradient-based optimization. According to the expectation conservation law in this m-projection onto simultaneous rank-1 subspace, one-body η-parameters do not change in the m-projection (See more general statements in Proposition 2.1). That is, for any $i \in [I]$, $j \in [J]$, $n \in [N]$, $m \in [M]$ and $l \in [L]$,

$$\eta^{\mathbf{Y}}_{n1} = \overline{\eta}^{\mathbf{Y}}_{n1}, \quad \eta^{\mathbf{Y}}_{1j} = \overline{\eta}^{\mathbf{Y}}_{1j}, \quad \eta^{\mathbf{X}}_{i1} = \overline{\eta}^{\mathbf{X}}_{i1}, \quad \eta^{\mathbf{Z}}_{1m} = \overline{\eta}^{\mathbf{Z}}_{1m}, \quad \eta^{\mathbf{U}}_{l1} = \overline{\eta}^{\mathbf{U}}_{l1} \tag{4.4}$$

where η is the expectation parameter of input, and $\overline{\eta}$ is the expectation parameter after the m-projection onto simultaneous rank-1 subspace. By the definition of expectation parameters, we obtain

$$\begin{aligned} \eta^{\mathbf{Y}}_{n1} - \eta^{\mathbf{Y}}_{n+1,1} &= a_n S(\boldsymbol{h}), \quad \eta^{\mathbf{X}}_{i1} - \eta^{\mathbf{X}}_{i+1,1} = w_i \left(S(\boldsymbol{h}) + S(\boldsymbol{b})\right), \\ \eta^{\mathbf{U}}_{l1} - \eta^{\mathbf{U}}_{l+1,1} &= d_l S(\boldsymbol{b}), \quad \eta^{\mathbf{Y}}_{1j} - \eta^{\mathbf{Y}}_{1,j+1} = \left(S(\boldsymbol{a}) + S(\boldsymbol{w})\right) h_j, \\ \eta^{\mathbf{Z}}_{1m} - \eta^{\mathbf{Z}}_{1,m+1} &= \left(S(\boldsymbol{w}) + S(\boldsymbol{d})\right) b_m. \end{aligned}$$

The expectation conservation law in Equation (4.4) guarantees that the values of the left-hand sides do not change before the m-projection, and they also do not change after the m-projection. Since the sum of each matrix $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ and $\mathbf{U}$ is represented by one-body η-parameters, the sum of each matrix does not change in the m-projection. Using these facts and multiplying these equations together, we can derive Theorem 4.1. Complete proof of Theorem 4.1 is as follows.

Proof: First, we show this theorem with $\alpha = \beta = \gamma = 1$, followed by generalizing the result for any non-negative α, β and γ. Hereinafter, we use overline for quantities on the simultaneous rank-1 subspace; for example, $(\overline{\mathbf{X}}, \overline{\mathbf{Y}}, \overline{\mathbf{Z}}, \overline{\mathbf{U}})$ as rank-1 matrices sharing factors, $\overline{\eta}$ as expectation parameters for distributions on simultaneous rank-1 subspace. We decompose input matrices $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ and $\mathbf{U}$ as $\overline{\mathbf{X}} = \boldsymbol{w} \otimes \boldsymbol{h}$, $\overline{\mathbf{Y}} = \boldsymbol{a} \otimes \boldsymbol{h}$, $\overline{\mathbf{Z}} = \boldsymbol{w} \otimes \boldsymbol{b}$, and $\overline{\mathbf{U}} = \boldsymbol{c} \otimes \boldsymbol{b}$, respectively, so that they minimize the cost function

$$D(\mathbf{X}, \boldsymbol{w} \otimes \boldsymbol{h}) + \alpha D(\mathbf{Y}, \boldsymbol{a} \otimes \boldsymbol{h}) + \beta D(\mathbf{Z}, \boldsymbol{w} \otimes \boldsymbol{b}) + \gamma D(\mathbf{U}, \boldsymbol{c} \otimes \boldsymbol{b}). \tag{4.5}$$

To simplify, we define

$$\eta^{\mathbf{Y}}_{nj} = \eta_{nj}, \quad \eta^{\mathbf{X}}_{ij} = \eta_{N+i,j}, \quad \eta^{\mathbf{Z}}_{im} = \eta_{N+i,J+m}, \quad \eta^{\mathbf{U}}_{lm} = \eta_{N+I+l,J+m},$$

for all $i \in [I], j \in [J], n \in [N], m \in [m]$ and $l \in [L]$. According to the expectation conservation law in this m-projection, it holds that

$$\eta^{\mathbf{Y}}_{n1} = \overline{\eta}^{\mathbf{Y}}_{n1}, \quad \eta^{\mathbf{Y}}_{1j} = \overline{\eta}^{\mathbf{Y}}_{1j}, \quad \eta^{\mathbf{X}}_{i1} = \overline{\eta}^{\mathbf{X}}_{i1}, \quad \eta^{\mathbf{Z}}_{1m} = \overline{\eta}^{\mathbf{Z}}_{1m}, \quad \eta^{\mathbf{U}}_{l1} = \overline{\eta}^{\mathbf{U}}_{l1}$$

where $(\eta^{\mathbf{X}}, \eta^{\mathbf{Y}}, \eta^{\mathbf{Z}}, \eta^{\mathbf{U}})$ is the expectation parameter of input, and $(\overline{\eta}^{\mathbf{X}}, \overline{\eta}^{\mathbf{Y}}, \overline{\eta}^{\mathbf{Z}}, \overline{\eta}^{\mathbf{U}})$ is the expectation parameter after the m-projection. By the definition of expectation parameters and the conservation law, we obtain

$$\begin{cases} \eta^{\mathbf{Y}}_{n1} - \eta^{\mathbf{Y}}_{n+1,1} = a_n S(\boldsymbol{h}), \\ \eta^{\mathbf{X}}_{i1} - \eta^{\mathbf{X}}_{i+1,1} = w_i \left(S(\boldsymbol{h}) + S(\boldsymbol{b})\right), \\ \eta^{\mathbf{U}}_{l1} - \eta^{\mathbf{U}}_{l+1,1} = d_l S(\boldsymbol{b}), \\ \eta^{\mathbf{Y}}_{1j} - \eta^{\mathbf{Y}}_{1,j+1} = \left(S(\boldsymbol{a}) + S(\boldsymbol{w})\right) h_j, \\ \eta^{\mathbf{Z}}_{1m} - \eta^{\mathbf{Z}}_{1,m+1} = \left(S(\boldsymbol{w}) + S(\boldsymbol{d})\right) b_m. \end{cases}$$

We multiply these equations together and simplify them, resulting in

$$\begin{aligned} \overline{\mathbf{X}}_{ij} &= w_i h_j = \frac{\left(\eta^{\mathbf{X}}_{i1} - \eta^{\mathbf{X}}_{i+1,1}\right)\left(\eta^{\mathbf{Y}}_{1j} - \eta^{\mathbf{Y}}_{1,j+1}\right)}{\left(S(\boldsymbol{w}) + S(\boldsymbol{a})\right)\left(S(\boldsymbol{h}) + S(\boldsymbol{b})\right)}, \\ \overline{\mathbf{Y}}_{nj} &= a_n h_j = \frac{\left(\eta^{\mathbf{Y}}_{n1} - \eta^{\mathbf{Y}}_{n+1,1}\right)\left(\eta^{\mathbf{Y}}_{1j} - \eta^{\mathbf{Y}}_{1,j+1}\right)}{\left(S(\boldsymbol{w}) + S(\boldsymbol{a})\right) S(\boldsymbol{h})}, \\ \overline{\mathbf{Z}}_{im} &= w_i b_m = \frac{\left(\eta^{\mathbf{X}}_{i1} - \eta^{\mathbf{X}}_{i+1,1}\right)\left(\eta^{\mathbf{Z}}_{1m} - \eta^{\mathbf{Z}}_{1,m+1}\right)}{\left(S(\boldsymbol{h}) + S(\boldsymbol{b})\right)\left(S(\boldsymbol{w}) + S(\boldsymbol{d})\right)}, \\ \overline{\mathbf{U}}_{lm} &= d_l b_m = \frac{\left(\eta^{\mathbf{U}}_{l1} - \eta^{\mathbf{U}}_{l+1,1}\right)\left(\eta^{\mathbf{Z}}_{1m} - \eta^{\mathbf{Z}}_{1,m+1}\right)}{S(\boldsymbol{b})\left(S(\boldsymbol{w}) + S(\boldsymbol{d})\right)}. \end{aligned}$$

Since the sum of each matrix $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ and $\mathbf{U}$ are represented by conserved quantities, $S(\mathbf{X}) = \eta_{11}^{\mathbf{X}} - \eta_{11}^{\mathbf{Z}}$, $S(\mathbf{Y}) = \eta_{11}^{\mathbf{Y}} - \eta_{11}^{\mathbf{X}}$, $S(\mathbf{Z}) = \eta_{11}^{\mathbf{Z}} - \eta_{11}^{\mathbf{U}}$, and $S(\mathbf{U}) = \eta_{11}^{\mathbf{U}}$,the sum of each matrix also do not change in this projection, that is,

$$
\begin{aligned}
S(\mathbf{X}) = S(\overline{\mathbf{X}}) = S(\boldsymbol{w} \otimes \boldsymbol{h}) = S(\boldsymbol{w})S(\boldsymbol{h}),\\
S(\mathbf{Y}) = S(\overline{\mathbf{Y}}) = S(\boldsymbol{a} \otimes \boldsymbol{h}) = S(\boldsymbol{a})S(\boldsymbol{h}),\\
S(\mathbf{Z}) = S(\overline{\mathbf{Z}}) = S(\boldsymbol{w} \otimes \boldsymbol{b}) = S(\boldsymbol{w})S(\boldsymbol{b}),\\
S(\mathbf{U}) = S(\overline{\mathbf{U}}) = S(\boldsymbol{c} \otimes \boldsymbol{b}) = S(\boldsymbol{c})S(\boldsymbol{b}),
\end{aligned}
$$

and we obtain

$$
\begin{cases}
\overline{\mathbf{X}}_{ij} = \dfrac{S(\mathbf{X}) \left(\sum_{j'}^{J} \mathbf{X}_{ij'} + \sum_{m}^{M} \mathbf{Z}_{im}\right) \left(\sum_{i'}^{I} \mathbf{X}_{i'j} + \sum_{n}^{N} \mathbf{Y}_{nj}\right)}{\left(S(\mathbf{X}) + S(\mathbf{Y})\right) \left(S(\mathbf{X}) + S(\mathbf{Z})\right)},\\
\overline{\mathbf{Y}}_{nj} = \dfrac{\left(\sum_{i}^{I} \mathbf{X}_{ij} + \sum_{n}^{N} \mathbf{Y}_{nj}\right) \left(\sum_{j'}^{J} \mathbf{Y}_{ij'}\right)}{S(\mathbf{X}) + S(\mathbf{Y})},\\
\overline{\mathbf{Z}}_{im} = \dfrac{S(\mathbf{Z}) \left(\sum_{j}^{J} \mathbf{X}_{ij} + \sum_{m'}^{M} \mathbf{Z}_{im'}\right) \left(\sum_{i'}^{I} \mathbf{Z}_{i'm} + \sum_{l}^{L} \mathbf{U}_{lm}\right)}{\left(S(\mathbf{Z}) + S(\mathbf{U})\right) \left(S(\mathbf{X}) + S(\mathbf{Z})\right)}.\\
\overline{\mathbf{U}}_{lm} = \dfrac{\left(\sum_{i}^{I} \mathbf{Z}_{im} + \sum_{l'}^{L} \mathbf{U}_{l'j}\right) \left(\sum_{j'}^{J} \mathbf{U}_{ij'}\right)}{S(\mathbf{Z}) + S(\mathbf{U})}.
\end{cases}
$$

Note that we used relations

$$
\begin{aligned}
\frac{1}{S(\mathbf{X}) + S(\mathbf{Y}) + S(\mathbf{Z}) + S(\boldsymbol{a})S(\boldsymbol{b})} &= \frac{S(\mathbf{X})}{\left(S(\mathbf{X}) + S(\mathbf{Y})\right)\left(S(\mathbf{X}) + S(\mathbf{Z})\right)},\\
\frac{1}{S(\mathbf{X}) + S(\mathbf{U}) + S(\mathbf{Z}) + S(\boldsymbol{h})S(\boldsymbol{c})} &= \frac{S(\mathbf{Z})}{\left(S(\mathbf{Z}) + S(\mathbf{U})\right)\left(S(\mathbf{Z}) + S(\mathbf{X})\right)}.
\end{aligned}
$$

Using the general property of the KL divergence, $\lambda D(\mathbf{P}, \mathbf{Q}) = D(\lambda \mathbf{P}, \lambda \mathbf{Q})$ for any matrices $\mathbf{P}, \mathbf{Q}$ and non-negative number λ, the above result with general α, β and γ is obtained by shifting matrices $\mathbf{Y} \to \alpha \mathbf{Y}$, $\mathbf{Z} \to \beta \mathbf{Z}$ and $\mathbf{U} \to \gamma \mathbf{U}$ in the both sides of the above equation as

$$
\begin{cases}
\overline{\mathbf{X}}_{ij} = \dfrac{S(\mathbf{X}) \left(\sum_{j'}^{J} \mathbf{X}_{ij'} + \beta \sum_{m}^{M} \mathbf{Z}_{im}\right) \left(\sum_{i'}^{I} \mathbf{X}_{i'j} + \alpha \sum_{n}^{N} \mathbf{Y}_{nj}\right)}{\left(S(\mathbf{X}) + \alpha S(\mathbf{Y})\right) \left(S(\mathbf{X}) + \beta S(\mathbf{Z})\right)},\\
\overline{\mathbf{Y}}_{nj} = \dfrac{\left(\sum_{i}^{I} \mathbf{X}_{ij} + \alpha \sum_{n}^{N} \mathbf{Y}_{nj}\right) \left(\sum_{j'}^{J} \mathbf{Y}_{ij'}\right)}{S(\mathbf{X}) + \alpha S(\mathbf{Y})},\\
\overline{\mathbf{Z}}_{im} = \dfrac{S(\mathbf{Z}) \left(\sum_{j}^{J} \mathbf{X}_{ij} + \beta \sum_{m'}^{M} \mathbf{Z}_{im'}\right) \left(\beta \sum_{i'}^{I} \mathbf{Z}_{i'm} + \gamma \sum_{l}^{L} \mathbf{U}_{lm}\right)}{\left(\beta S(\mathbf{Z}) + \gamma S(\mathbf{U})\right) \left(S(\mathbf{X}) + \beta S(\mathbf{Z})\right)}.\\
\overline{\mathbf{U}}_{lm} = \dfrac{\left(\beta \sum_{i}^{I} \mathbf{Z}_{im} + \gamma \sum_{l'}^{L} \mathbf{U}_{l'j}\right) \left(\sum_{j'}^{J} \alpha \mathbf{U}_{ij'}\right)}{\beta S(\mathbf{Z}) + \gamma S(\mathbf{U})}.
\end{cases}
$$

Thus, the theorem was proved. □

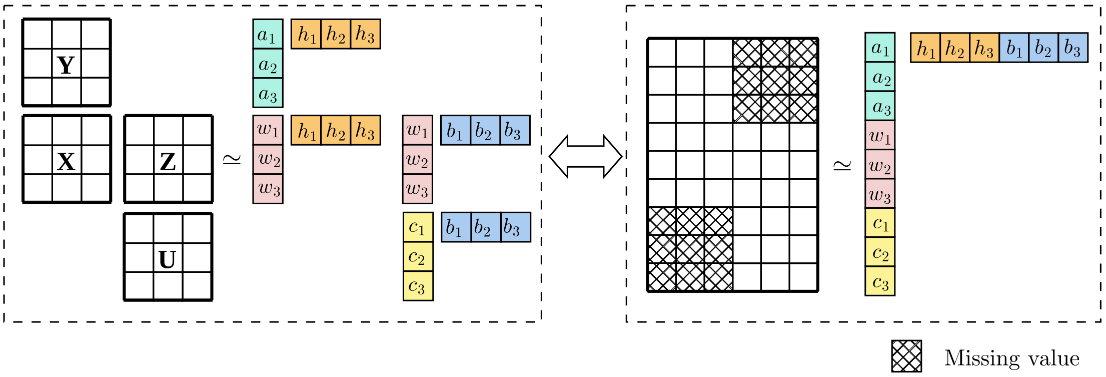


Figure 4.3 A sketch of relationship between Rank-1 NMMF with four inputs matrices and NMF with missing values for $I = J = N = M = L = 3$. The cost functions of these two tasks are equivalent.

4.5 Rank-1 Missing NMF based on Rank-1 NMMF

As an application of the closed-form in Theorem 4.1, we develop an efficient method to solve rank-1 NMF for missing data. Here we provide the definition of the task.

Task 4.2 Rank-1 NMF with Missing Values (Rank-1 missing NMF)

Rank-1 NMF for a given matrix $\mathbf{X} \in \mathbb{R}_{\geq 0}^{I \times J}$ with missing values (rank-1 missing NMF) is the task of finding two non-negative vectors $\boldsymbol{w} \in \mathbb{R}_{\geq 0}^{I}$ and $\boldsymbol{h} \in \mathbb{R}_{\geq 0}^{J}$ that minimize a weighted cost function $D_{\Phi}(\mathbf{X}, \boldsymbol{w} \otimes \boldsymbol{h})$ defined as

$$D_{\mathbf{\Phi}}(\mathbf{X}, \boldsymbol{w} \otimes \boldsymbol{h}) = D(\mathbf{\Phi} \circ \mathbf{X}, \mathbf{\Phi} \circ (\boldsymbol{w} \otimes \boldsymbol{h})) \tag{4.6}$$

for a binary weight matrix $\mathbf{\Phi} \in \{0, 1\}^{I \times J}$. The weight matrix indicates the position of missing values; that is, $\mathbf{\Phi}_{ij} = 0$ if the entry $\mathbf{X}_{ij}$ is missing, $\mathbf{\Phi}_{ij} = 1$ otherwise.

Note that the above cost function (4.6) is always convex in $\boldsymbol{w}$ and $\boldsymbol{h}$.

If the binary weight matrix $\mathbf{\Phi}$ satisfies $\mathrm{Rank}\,(\mathbf{\Phi}) \leq 2$, we can find the exact solution for rank-1 missing NMF. After we mention the relationship between NMMF and missing NMF in Section 4.5.1, we demonstrate a way to find the best rank-1 missing NMF when it holds $\mathrm{Rank}\,(\mathbf{\Phi}) \leq 2$ in Sections 4.5.2 – 4.5.3. In addition, we develop an efficient method for the general cases to find an approximate solution based on the closed formula. The proposed method is described in Section 4.5.4 [1].

4.5.1 Connection between NMMF and missing NMF

Our discussion is based on the following fundamental two facts. First, we can regard NMMF as a special case of missing NMF. We assume that a binary weight matrix $\mathbf{\Phi} \in \{0,1\}^{N+I+L,J+M}$ and an input matrix $\mathbf{K} \in \mathbb{R}_{\geq 0}^{N+I+L,J+M}$ are given in the form of

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{1}_{NJ} & \mathbf{0}_{NM} \\ \mathbf{1}_{IJ} & \mathbf{1}_{IM} \\ \mathbf{0}_{LJ} & \mathbf{1}_{LM} \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} \mathbf{Y} & \mathbf{F} \\ \mathbf{X} & \mathbf{Z} \\ \mathbf{E} & \mathbf{U} \end{bmatrix}, \tag{4.7}$$

where $\mathbf{X} \in \mathbb{R}_{>0}^{I\times J}$, $\mathbf{Y} \in \mathbb{R}_{>0}^{N\times J}$, $\mathbf{Z} \in \mathbb{R}_{>0}^{I\times M}$, $\mathbf{U} \in \mathbb{R}_{>0}^{L\times M}$, $\mathbf{E} \in \mathbb{R}_{>0}^{L\times J}$, and $\mathbf{F} \in \mathbb{R}_{>0}^{N\times M}$. The all of elements of $\mathbf{E}$ and $\mathbf{F}$ are missing. We consider the rank-1 approximation of $\mathbf{K}$ as

$$\mathbf{K_1} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{w} \\ \boldsymbol{c} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}^\top & \boldsymbol{b}^\top \end{bmatrix}. \tag{4.8}$$

In this situation, the cost function of missing NMF is equivalent to that of NMMF [121]:

$$\begin{aligned} &\underset{\mathbf{K}_1;\mathrm{rank}(\mathbf{K}_1)=1}{\mathrm{argmin}} \quad D_{\mathbf{\Phi}}(\mathbf{K},\mathbf{K}_1) \\ &= \underset{\boldsymbol{w},\boldsymbol{h},\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}{\mathrm{argmin}}\, D(\mathbf{X},\boldsymbol{w}\otimes\boldsymbol{h})+D(\mathbf{Y},\boldsymbol{a}\otimes\boldsymbol{h})+D(\mathbf{Z},\boldsymbol{w}\otimes\boldsymbol{b})+D(\mathbf{U},\boldsymbol{c}\otimes\boldsymbol{b}). \end{aligned}$$

The second fundamental fact is the homogeneity of rank-1 missing NMF, which ensures a factorization after row or column permutations can be reproduced by permutations after the factorization.

Proposition 4.2 Homogeneity of Rank-1 Missing NMF

Let $\mathrm{NMF}_1(\mathbf{\Phi},\mathbf{X})$ be the best rank-1 matrix $\boldsymbol{w}\otimes\boldsymbol{h}$, which minimizes the cost function in Equation (4.6). For any permutation matrices $\mathbf{G}$ and $\mathbf{H}$, it holds that

$$\mathrm{NMF}_1\left(\mathbf{G\Phi H},\mathbf{GKH}\right) = \mathbf{G}^\top \mathrm{NMF}_1\left(\mathbf{\Phi},\mathbf{K}\right)\mathbf{H}^\top.$$

Proof: We assume $\mathbf{K} \in \mathbb{R}_{\geq 0}^{n\times m}$ and $\mathbf{\Phi} \in \{0,1\}^{n\times m}$. Let $\mathbf{G} \in \{0,1\}^{n\times n}$ and $\mathbf{H} \in \{0,1\}^{m\times m}$ be the permutation matrices corresponding to the mappings $\mathcal{G} : i \mapsto g(i)$ and $\mathcal{H} : j \mapsto h(j)$, respectively. This means that, for a given matrix $\mathbf{X}$ and its row and column permutation $\mathbf{X}' = \mathbf{GXH}$, we have $\mathbf{X}'_{g(i),h(j)} = \mathbf{X}_{i,j}$. We can also apply the permutation matrices $\mathbf{G}$ and $\mathbf{H}$ to vectors

[1] Implementation is available at: https://github.com/gkazunii/A1GM

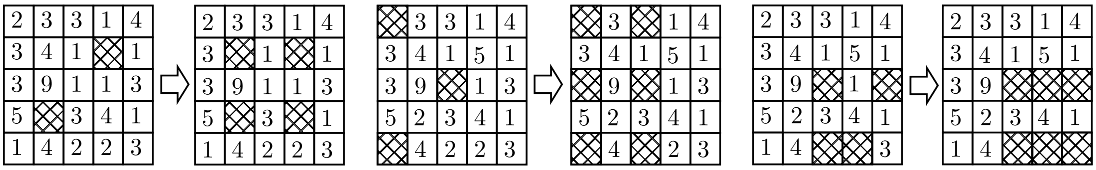


Figure 4.4 Examples of matrices with non-grid-like missing values (left) and grid-like missing values (right). Meshed entries are missing values. We can create grid-like missing values by increasing missing values.

$\boldsymbol{w} \in \mathbb{R}^n_{\geq 0}$ and $\boldsymbol{h} \in \mathbb{R}^m_{\geq 0}$. For $\boldsymbol{w}' = \mathbf{G}\boldsymbol{w}$ and $\boldsymbol{h}' = \mathbf{H}\boldsymbol{h}$, it holds that $w'_{g(i)} = w_i$ and $h'_{h(j)} = h_j$. We define $\boldsymbol{w}^*$ and $\boldsymbol{h}^*$ as

$$\begin{aligned}\boldsymbol{w}^*, \boldsymbol{h}^* &\equiv \underset{\boldsymbol{w},\boldsymbol{h}}{\operatorname{argmin}}\, D_{\mathbf{G\Phi H}}\left(\mathbf{GXH}, \boldsymbol{w} \otimes \boldsymbol{h}\right) \\ &= \underset{\boldsymbol{w},\boldsymbol{h}}{\operatorname{argmin}} \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\mathbf{\Phi}_{g(i)h(j)} a_i b_j - \mathbf{\Phi}_{g(i)h(j)} \mathbf{X}_{g(i)h(j)} \log a_i b_j\right).\end{aligned}$$

We replace $\boldsymbol{w}$ with $\mathbf{G}\boldsymbol{w}$ and $\boldsymbol{h}$ with $\mathbf{H}\boldsymbol{h}$ and we get

$$\begin{aligned}\mathbf{G}\boldsymbol{w}^*, \mathbf{H}\boldsymbol{h}^* &= \underset{\boldsymbol{w},\boldsymbol{h}}{\operatorname{argmin}} \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\mathbf{\Phi}_{g(i)h(j)} a_{g(i)} b_{h(j)} - \mathbf{\Phi}_{g(i)h(j)} \mathbf{X}_{g(i)h(j)} \log a_{g(i)} b_{h(j)}\right) \\ &= \underset{\boldsymbol{w},\boldsymbol{h}}{\operatorname{argmin}} \sum_{g(i)=1}^{n}\sum_{h(j)=1}^{m}\left(\mathbf{\Phi}_{g(i)h(j)} a_{g(i)} b_{h(j)} - \mathbf{\Phi}_{g(i)h(j)} \mathbf{X}_{g(i)h(j)} \log a_{g(i)} b_{h(j)}\right) \\ &= \underset{\boldsymbol{w},\boldsymbol{h}}{\operatorname{argmin}} \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\mathbf{\Phi}_{ij} a_i b_j - \mathbf{\Phi}_{ij} \mathbf{X}_{ij} \log a_i b_j\right) \\ &= D_{\mathbf{\Phi}}\left(\mathbf{X}, \boldsymbol{w} \otimes \boldsymbol{h}\right).\end{aligned}$$

Thus, it holds that $\boldsymbol{w}^* \otimes \boldsymbol{h}^* = \mathrm{NMF}_1\left(\mathbf{G\Phi H}, \mathbf{GXH}\right)$ and $\mathbf{G}\left(\boldsymbol{w}^* \otimes \boldsymbol{h}^*\right)\mathbf{H} = \mathrm{NMF}_1\left(\mathbf{\Phi}, \mathbf{X}\right)$. Therefore, we have

$$\mathrm{NMF}_1\left(\mathbf{G\Phi H}, \mathbf{GTH}\right) = \mathbf{G}^\top \mathrm{NMF}_1\left(\mathbf{\Phi}, \mathbf{K}\right)\mathbf{H}^\top.$$

We use the fact that permutation matrix is always orthogonal; that is, $\mathbf{G}^{-1} = \mathbf{G}^\top$ and $\mathbf{H}^{-1} = \mathbf{H}^\top$. □

Therefore, using the closed formula of the best rank-1 NMMF in Theorem 4.1, we can solve the rank-1 missing NMF when we can relocate the position of missing values to the form Equation (4.7) by row and column permutations.

4.5.2 Rank-1 Missing NMF for Grid-like Missing

We introduce the term *grid-like*, defined as follows. As we describe in this section, we can regard missing NMF as NMMF that requires only three matrices $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ as input, which corresponds to the case $L = 0$ in Theorem 4.1.

Definition 4.2 Grid-like Binary Weight Matrix

Let $\mathbf{\Phi} \in \{0,1\}^{I\times J}$ be a binary weight matrix. If there exist two sets $S^{(1)} \subset [I]$ and $S^{(2)} \subset [J]$ such that

$$\mathbf{\Phi}_{ij} = \begin{cases} 0 & \text{if } \ i \in S^{(1)} \text{ and } j \in S^{(2)}, \\ 1 & \text{otherwise}, \end{cases}$$

$\mathbf{\Phi}$ is called grid-like.

It holds that $\mathrm{rank}\,(\mathbf{\Phi}) = 2$ if $\mathbf{\Phi}$ is grid-like. But the converse is not true. We discuss the case $\mathrm{rank}\,(\mathbf{\Phi}) = 2$ but not grid-like in Section 4.5.3.

Real-world tabular datasets tend to have missing values on only certain rows or columns. Therefore, the binary weight matrix $\mathbf{\Phi}$ often becomes grid-like in practice (we show example datasets in Section 4.5.7). Figure 4.4 illustrates examples of matrices with grid-like missing values.

When $\mathbf{\Phi} \in \{0,1\}^{I+N,J+M}$ is grid-like, we can transform it in the form given in Equation (4.7) with $L = 0$ using row and column permutations. Let $S^{(1)} \subset [I+N]$ and $S^{(2)} \subset [J+M]$ with $\mid S^{(1)} \mid = C^{(1)}$ and $\mid S^{(2)} \mid = C^{(2)}$ be the row and column index sets for zero entries in $\mathbf{\Phi}$. For the block at the upper right of $\mathbf{\Phi}$ whose row and column indices are specified as

$$B^{(1)} = [C^{(1)}], \quad B^{(2)} = [J+M-C^{(2)}+1, J+M],$$

we can collect all the zero entries of $\mathbf{\Phi}$ in the rectangular region $B^{(1)} \times B^{(2)}$ using row and column permutations. Formally, for a grid-like binary weight matrix $\mathbf{\Phi}$, there are row $\mathcal{G} : [I+N] \to [I+N]$ and column $\mathcal{H} : [J+M] \to [J+M]$ permutations satisfying

$$(\mathbf{G}\mathbf{\Phi}\mathbf{H})_{ij} = \begin{cases} 0 & \text{if } \ i \in B^{(1)} \text{ and } j \in B^{(2)} \\ 1 & \text{otherwise} \end{cases}$$

where $\mathbf{G}$ and $\mathbf{H}$ are corresponding permutation matrices to $\mathcal{G}$ and $\mathcal{H}$, respectively.

We can obtain $\mathbf{G}$ and $\mathbf{H}$ as follows. First, we focus on row permutation $\mathcal{G}$. We want to include each row $j \in S^{(1)} \cap B^{(1)c}$, where $B^{(1)c} = [I+N] \backslash B^{(1)}$, in $B^{(1)}$ by row permutation $\mathcal{G}$, which can be achieved by any one-to-one mapping from $S^{(1)} \cap B^{(1)c}$ to $S^{(1)c} \cap B^{(1)}$, where $S^{(1)c} = [I+N] \setminus S^{(1)}$. Note that $\mid S^{(1)} \cap B^{(1)c} \mid = \mid S^{(1)c} \cap B^{(1)} \mid$ always holds. The corresponding permutation matrix is given as

$$\mathbf{G} = \prod_{k \in S^{(1)} \cap B^{(1)c}} \mathbf{R}^{k \leftrightarrow \mathcal{G}(k)}$$

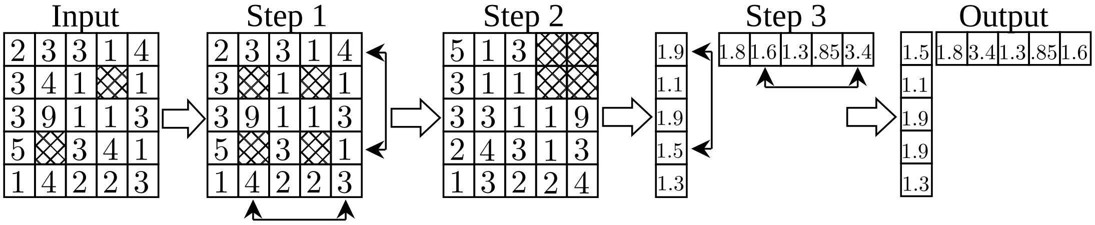


Figure 4.5 Sketch of the algorithm of A1GM. Meshed entries are missing values. In Step 1, we increase missing values so that they become grid-like. In Step 2, we gather missing values in the block at the upper right by low and column permutations. In Step 3, we use the closed formula of the best rank-1 NMMF in Theorem 4.1 with $L = 0$. In this example, we get $\boldsymbol{w} = (1.9, 1.5, 1.3)^\top, \boldsymbol{a} = (1.9, 1.1)^\top, \boldsymbol{h} = (1.8, 1.6, 1.3)^\top, \boldsymbol{b} = (0.85, 3.4)^\top$. Finally, we get two vectors as the output by the repermutation. We use two significant digits in this figure.

where $\mathbf{R}^{k\leftrightarrow l}$ is a permutation matrix, which switches the k-th row and the l-th row; that is,

$$\mathbf{R}^{k\leftrightarrow l}_{ij} = \begin{cases} 0 & \text{if } (i,j) = (k,k) \text{ or } (l,l), \\ 1 & \text{if } (i,j) = (k,l) \text{ or } (l,k), \\ \mathbf{I}_{ij} & \text{otherwise.} \end{cases}$$

Since $S^{(1)} \cap B^{(1)^c}$ and $S^{(1)c} \cap B^{(1)}$ are disjoint, it holds that $\mathbf{G} = \mathbf{G}^\top$.

In the same way, any one-to-one mapping from $S^{(2)} \cap B^{(2)c}$ to $S^{(2)c} \cap B^{(2)}$ can be $\mathcal{H}$, where $S^{(2)c} = [J+M] \setminus S^{(2)}$ and $B^{(2)c} = [J+M] \setminus B^{(2)}$. The corresponding permutation matrix is given as

$$\mathbf{H} = \prod_{k \in S^{(2)} \cap B^{(2)c}} \mathbf{R}^{k\leftrightarrow \mathcal{H}(k)},$$

which is also a symmetric matrix.

The above discussion leads to the following procedure of the best rank-1 missing NMF for an input matrix $\mathbf{K}$ if a binary weight matrix $\mathbf{\Phi}$ is grid-like. The first step is to find proper permutations $\mathbf{G}$ and $\mathbf{H}$ to collect the missing values in the upper right corner. In the next step, we obtained $\mathrm{NMF}_1\left(\mathbf{G\Phi H}, \mathbf{GKH}\right)$ using the closed formula of the best rank-1 NMMF. In the final step, we operated the inverse permutations of $\mathbf{G}$ and $\mathbf{H}$ to the result of the previous step; that is, $\mathbf{G}^{-1}\mathrm{NMF}_1\left(\mathbf{G\Phi H}, \mathbf{GKH}\right)\mathbf{H}^{-1}$. Note that $\mathbf{G}^{-1} = \mathbf{G}^\top = \mathbf{G}$ and $\mathbf{H}^{-1} = \mathbf{H}^\top = \mathbf{H}$ always holds since these permeation matrices are orthogonal and symmetrical.

4.5.3 Rank-1 Missing NMF with $\mathrm{Rank}\left(\mathbf{\Phi}\right) \leq 2$

In this subsection, we show that we can relocate missing values into the form of Equation (4.7) by column and row permutations if the binary weight satisfies $\mathrm{rank}\left(\mathbf{\Phi}\right) = 2$ and solve the rank-1 missing NMF as a rank-1 NMMF.

As we can confirm immediately, there are only two cases when the rank of a binary matrix $\mathbf{\Phi}$ is 1. In the first case, all of the elements $\mathbf{\Phi}_{ij}$ are 1. This case does not happen in our case because the number of missing values is assumed to be strictly larger than 0, resulting in $\mathbf{\Phi} \neq \mathbf{1}$. In the second case, if there are rows or columns with all zero elements in $\mathbf{\Phi}$, the matrix rank of $\mathbf{\Phi}$ can be 1. However, since rows and columns with all zero elements do not contribute to the cost function (4.6), we ignore such rows and columns. As a result, we discuss only the case of $\mathrm{Rank}(\mathbf{\Phi}) = 2$.

We consider a rank-2 weight matrix $\mathbf{\Phi} \in \{0,1\}^{N+I+L, J+M}$. There are two linear independent column bases if and only if $\mathrm{Rank}\,(\mathbf{\Phi}) = 2$ since row-rank and column-rank are always the same. We define them as $\boldsymbol{a} \in \{0,1\}^{N+I+L}$ and $\boldsymbol{b} \in \{0,1\}^{N+I+L}$. We also assume that $\boldsymbol{a} \neq \mathbf{0}$ and $\boldsymbol{b} \neq \mathbf{0}$ since a zero-vector cannot be a basis. Then, for the binary matrix $\mathbf{\Phi} = [\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(J+M)}]$, any column $\boldsymbol{c}^{(i)}$ should be able to be written as $\boldsymbol{c}^{(i)} = \alpha_i \boldsymbol{a} + \beta_i \boldsymbol{b}$ using two bases $\boldsymbol{a}$ and $\boldsymbol{b}$. Since $\boldsymbol{c}^{(i)}$ is a binary vector, possible domains of α_i and β_i are limited, and we analyze the domains in the following by separating them into three cases. In all of three cases, we can rearrange $\mathbf{\Phi}$ into the form of Equation (4.7) by permutations. To consider the possible values of the pair (α_i, β_i), we firstly define *disjoint* between binary vectors as follows:

Definition 4.3 Disjoint Binary Vectors

We say that two binary vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are *disjoint* with each other if $a_k \neq b_k$ for all k.

For example, two vectors $\boldsymbol{a} = (0,0,1,0)^\top$ and $\boldsymbol{b} = (1,1,0,1)^\top$ are disjoint and $\boldsymbol{a} = (1,0,0,1)^\top$ and $\boldsymbol{b} = (1,1,0,1)^\top$ are not disjoint. Using this concept, we divide possible pairs (α_i, β_i) into three cases for rank-2 $\mathbf{\Phi}$ as follows:

Case 1: the bases are disjoint If the bases $\boldsymbol{a}$ and $\boldsymbol{b}$ are disjoint, it holds that $(\alpha_i, \beta_i) \in \{\,(1,0),(0,1),(1,1)\,\}$, that is, $\boldsymbol{c}^{(i)}$ can be $\boldsymbol{a}$, $\boldsymbol{b}$, or $\boldsymbol{a}+\boldsymbol{b}$. Since $\boldsymbol{a}$ and $\boldsymbol{b}$ are disjoint, $\boldsymbol{a}+\boldsymbol{b} = \mathbf{1}$ follows. Then by only column permutations, we can arrange the binary weight matrix $\mathbf{\Phi} = [\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(J+M)}]$ in the form of

$$\mathbf{\Phi}\mathbf{H} = [\boldsymbol{a}, \ldots, \boldsymbol{a}, \mathbf{1}, \ldots, \mathbf{1}, \boldsymbol{b}, \ldots, \boldsymbol{b}],$$

where $\mathbf{H}$ is a permutation matrix corresponding to the column permutation. After the column permutation, we conduct row permutation as follows. First, we define $S = \{\, i \mid a_i = 0\,\}$, $C = \mid S \mid$ and $B = [J+M-C+1, J+M]$. There is a one-to-one mapping $\mathcal{G}$ from $S \cap B^c$ to $S^c \cap B$. The corresponding permutation matrix is given as

$$\mathbf{G} = \prod_{k \in S \cap B^c} \mathbf{R}^{k \leftrightarrow \mathcal{G}(k)},$$

where $B^c = [J+M] \backslash B$. By operating the permutation $\mathbf{G}$ on bases, we obtain,

$$\tilde{\boldsymbol{a}} \equiv \mathbf{G}\boldsymbol{a} = \begin{pmatrix} 1, & \dots, & 1, & 0, & \dots, & 0 \end{pmatrix}^\top, \qquad \tilde{\boldsymbol{b}} \equiv \mathbf{G}\boldsymbol{b} = \begin{pmatrix} 0, & \dots, & 0, & 1, & \dots, & 1 \end{pmatrix}^\top.$$

Using the fact $\mathbf{G}\mathbf{1} = \mathbf{1}$, finally, we obtain

$$\mathbf{G}\boldsymbol{\Phi}\mathbf{H} = [\tilde{\boldsymbol{a}}, \dots, \tilde{\boldsymbol{a}}, \mathbf{1}, \dots, \mathbf{1}, \tilde{\boldsymbol{b}}, \dots, \tilde{\boldsymbol{b}}],$$

which means $\mathbf{G}\boldsymbol{\Phi}\mathbf{H}$ is in the form of Equation (4.7).

Case 2: the bases are not disjoint, but one of them is one vector If the bases $\boldsymbol{a}$ and $\boldsymbol{b}$ are not disjoint but $\boldsymbol{a} = \mathbf{1}$, it holds that $(\alpha_i, \beta_i) \in \{ (1,0), (0,1), (1,-1) \}$. That is, $\boldsymbol{c}^{(i)}$ can be $\mathbf{1}$, $\boldsymbol{b}$, or $\mathbf{1} - \boldsymbol{b}$. As we can confirm immediately, $\boldsymbol{b}$ and $\mathbf{1} - \boldsymbol{b}$ are disjoint since the sum of them is $\mathbf{1}$. Then, we can rearrange $\boldsymbol{\Phi}$ in the form of Equation (4.7) as in the same way as in Case 1. If the bases $\boldsymbol{a}$ and $\boldsymbol{b}$ are not disjoint but $\boldsymbol{b} = \mathbf{1}$, it is also the same situation as Case 1.

Case 3: the bases are not disjoint and not one vector If the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are not disjoint and $\boldsymbol{a} \neq \mathbf{1}$ and $\boldsymbol{b} \neq \mathbf{1}$, it holds that $(\alpha_i, \beta_i) \in \{ (1,0), (0,1) \}$ since $\boldsymbol{a} + \boldsymbol{b}$ includes 2 and $\pm(\boldsymbol{a} - \boldsymbol{b})$ makes rows such that all the elements are 0, which is contrary to the assumption. By only column permutations, we can arrange the binary weight matrix $\boldsymbol{\Phi} = [\boldsymbol{c}^{(1)}, \dots, \boldsymbol{c}^{(J+M)}]$ in the form of

$$\boldsymbol{\Phi}\mathbf{H} = [\boldsymbol{a}, \dots, \boldsymbol{a}, \boldsymbol{b}, \dots, \boldsymbol{b}].$$

In the same way with Case 1, we obtain $\mathbf{G}\boldsymbol{\Phi}\mathbf{H} = [\tilde{\boldsymbol{a}}, \dots, \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{b}}, \dots, \tilde{\boldsymbol{b}}]$, which corresponds to the form of Equation (4.7) with $I = 0$, corresponding to NMMF for only two matrices $\mathbf{Z}$ and $\mathbf{U}$.

The above discussion leads to the following procedure of the best rank-1 missing NMF for an input matrix $\mathbf{K}$ if the rank of the binary weight matrix $\boldsymbol{\Phi}$ is 2. The first step is to find the bases $\boldsymbol{a}$ and $\boldsymbol{b}$ of $\boldsymbol{\Phi}$ and find proper permutations $\mathbf{G}$ and $\mathbf{H}$ as described above to collect the missing values in the corners. The rest step is the same as described in the final paragraph in Section 4.5.2.

4.5.4 Rank-1 Missing NMF for the General Case

If the rank of the binary weight matrix $\boldsymbol{\Phi}$ is strictly larger than 2, the above procedure cannot be directly applied. To treat any matrices with missing values, our idea is *increase* missing values so that the rank of $\boldsymbol{\Phi}$ becomes 2. However, the optimal way to increase missing values is not obvious to make $\boldsymbol{\Phi}$ rank-2. Then, we increase missing values so that $\boldsymbol{\Phi}$ becomes grid-like since the optimal way to increase missing values is clear.

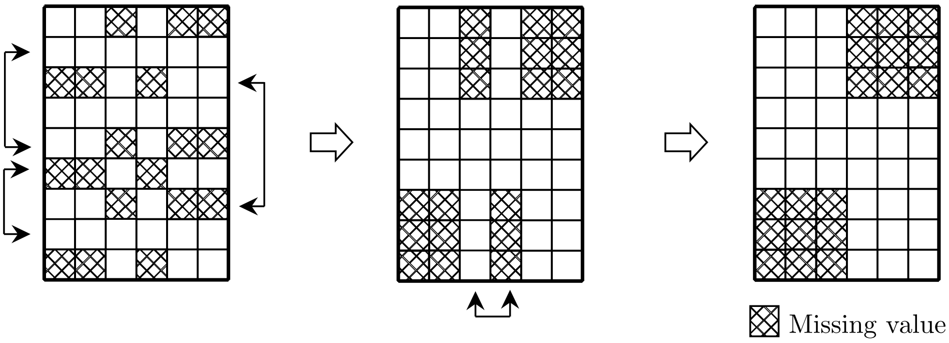


Figure 4.6 An examples of matrix with missing values where $\mathrm{rank}(\mathbf{\Phi}) = 2$. We can collect missing vlaues as a form of Equation (4.7) by column and row permutations if $\mathrm{rank}(\mathbf{\Phi}) \leq 2$ holds.

Although this strategy is counter-intuitive because we lose some information, which may cause a larger reconstruction error, we gain the efficiency instead of using our closed-form solution in Theorem 4.1 with $L = 0$, and, as we empirically show in the Section 4.5.7, the error increase is not significant in many datasets. Examples of this step are demonstrated in Figure 4.4.

In the worst case, the number of missing values after this step becomes k^2 for k missing values. If every row or column has at least one missing value, all indices are missing after this step, for which our algorithm does not work. Thus, our method is not suitable if there are too many missing values in a matrix.

We illustrate an example of the overall procedure of A1GM in Figure 4.5 and show its algorithm in Algorithm 2. Since the time complexity of each process of A1GM is at most linear with respect to the number of entries of an input matrix, the time complexity is $O((I+N)(J+M))$ for input matrix $\mathbf{X} \in \mathbb{R}^{(I+N)\times(J+M)}$.

4.5.5 Relation to *em*-algorithm

As a method to solve rank-1 NMF with missing values, the em-algorithm repeats the following e- and m-steps after filling missing values of an input matrix $\mathbf{T}$ with arbitrary values [138].

em-algorithm

m-step : Get the rank-1 approximation of the input matrix $\mathbf{T}$ that minimizes the KL divergence from $\mathbf{T}$.

e-step : Overwrite missing values of $\mathbf{T}$ by the obtained matrix in the m-step with keeping other values.

Algorithm 2: A1GM

```
input  : A binary weight matrix $\mathbf{\Phi} \in \{0,1\}^{I \times J}$, a matrix $\mathbf{X} \in \mathbb{R}_{\geq 0}^{I \times J}$
output : Dominant factors $\boldsymbol{c} \in \mathbb{R}_{\geq 0}^{I}$ and $\boldsymbol{d} \in \mathbb{R}_{\geq 0}^{J}$
A1GM($\mathbf{\Phi}$, $\mathbf{X}$)
    $S^{(1)} \leftarrow \varnothing$
    $S^{(2)} \leftarrow \varnothing$
    for $(i, j) \in [I] \times [J]$ do
        if $\mathbf{\Phi}_{ij} = 0$ then
            $S^{(1)} \leftarrow S^{(1)} \cup \{i\}$
            $S^{(2)} \leftarrow S^{(2)} \cup \{j\}$
    $B^{(1)} \leftarrow \{I- \mid S^{(1)} \mid +1, I- \mid S^{(1)} \mid +2, \ldots, I\}$
    $B^{(2)} \leftarrow \{J- \mid S^{(2)} \mid +1, J- \mid S^{(2)} \mid +2, \ldots, J\}$
    $\mathrm{perm1} \leftarrow (1, \ldots, I)$
    for $k \in \{1, 2, \ldots, \mid S^{(1)} \cap B^{(1)c} \mid\}$ do
        $i \leftarrow k$th smallest element of $S^{(1)} \cap B^{(1)c}$
        $j \leftarrow k$th smallest element of $S^{(1)c} \cap B^{(1)}$
        swap(perm1[$i$],perm1[$j$])
    $\mathrm{perm2} \leftarrow (1, \ldots, J)$
    for $k \in \{1, 2, \ldots, \mid S^{(2)} \cap B^{(2)c} \mid\}$ do
        $i \leftarrow k$th smallest element of $S^{(2)} \cap B^{(2)c}$
        $j \leftarrow k$th smallest element of $S^{(2)c} \cap B^{(2)}$
        swap(perm2[$i$],perm2[$j$])
    $\mathbf{X} \leftarrow \mathbf{X}[\mathrm{perm1}, \mathrm{perm2}]$
    $\boldsymbol{w}, \boldsymbol{h}, \boldsymbol{a}, \boldsymbol{b} \leftarrow$ the best rank-1 NMMF of $\mathbf{X}$ in Theorem 4.1 with $L = 0$.
    $\boldsymbol{c} \leftarrow$ concate $\boldsymbol{w}$ and $\boldsymbol{a}$
    $\boldsymbol{d} \leftarrow$ concate $\boldsymbol{h}$ and $\boldsymbol{b}$
    $\boldsymbol{c} \leftarrow \boldsymbol{c}[\mathrm{perm1}]$
    $\boldsymbol{d} \leftarrow \boldsymbol{d}[\mathrm{perm2}]$
    return $\boldsymbol{c}, \boldsymbol{d}$
```

This algorithm also minimizes the cost function (4.1) indirectly [4]. For grid-like data, A1GM directly finds the convergence point of the em-algorithm without performing the above iterations. We describe a sketch of em-algorithm in Figure 4.7.

4.5.6 Relation between A1GM and LTR

We summarize the relationship between the two proposed algorithms A1GM and LTR, which are based on the log-linear model on posets and its convex optimization via information geometry. The difference between A1GM and LTR is the structure of posets behind algorithms. After designing proper posets, these two algorithms perform m-projection in common, where some natural parameters become zero. Interestingly, θ- and η-parameters are not computed explicitly during the procedure of both algorithms. This is because the trick of describing the low-rank condition in a dual-flat coordinate

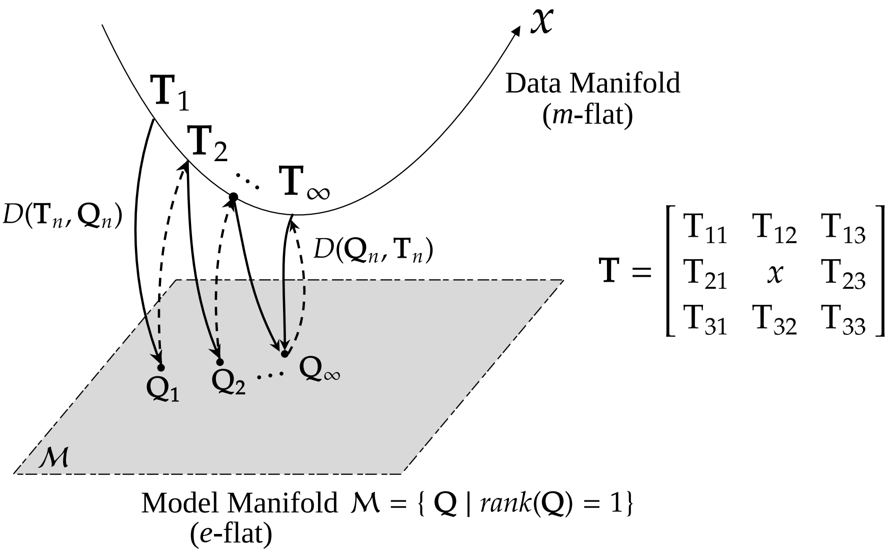

x
Data Manifold
(m-flat)
$\mathbf{T}_1$
$\mathbf{T}_2$
$\mathbf{T}_\infty$
$D(\mathbf{T}_n, \mathbf{Q}_n)$
$D(\mathbf{Q}_n, \mathbf{T}_n)$
$\mathbf{Q}_1$
$\mathbf{Q}_2$
$\mathbf{Q}_\infty$
$\mathcal{M}$
$\mathbf{T} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & x & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix}$
Model Manifold $\mathcal{M} = \{\, \mathbf{Q} \mid rank(\mathbf{Q}) = 1 \}$
(e-flat)

Figure 4.7 A sketch of em-algorithm for missing values estimation for a matrix $\mathbf{T} \in \mathbb{R}^{3\times3}$ and single missing value x. The algorithm estimates the value of x by repeating e-step and m-step. e-step is e-projection from model manifold to data manifold. m-step is m-projection from data manifold to model manifold. In this example, the model manifold is a set of rank-1 matrices. The dimension of the data manifold is the number of missing values. When the model manifold is e-flat and the data manifold is m-flat, the convergence of the algorithm and its uniqueness are guaranteed [4, Chapter 8.1]. Theorem 4.1 finds the convergence point $\mathbf{Q}_\infty$ without iteration if the rank of the weight matrix of $\mathbf{T}$ is less than or equal to 2.

system and using a conservation law for the parameters allows us to know the projection destination in a closed form.

It is also known that the constraints of tensor balancing can be described in terms of expectation parameters, as we see in Section 3.10. In our framework, the task to be solved, such as low-rank approximation or balancing, is described as a constraint in a dual-flat coordinate system. We expect that higher-rank approximations for multiple matrices could also be possible by defining bingos as well as LTR.

In summary, our approach formulates tasks as convex optimizations by taking the input data structure as proper poset and describing the constraints of the task in a dual-flat coordinate system.

4.5.7 Experiments for A1GM

We use three types of data to empirically investigate the efficiency and effectiveness of A1GM: (i) synthetic data with missing values at the upper right corner, (ii) synthetic data with random grid-like missing values, and (iii) real data with grid-like and non-grid-like missing values. It is guaranteed that A1GM always finds the best solution for any data of (i) and (ii). Thus, we only investigate efficiency in our experiments for (i) and (ii). By contrast, for data (iii), the reconstruction error can be worse than the existing methods due to increased missing values in A1GM. Therefore, in the experiment for data (iii), we investigate both efficiency (running time) and effectiveness (reconstruction error).

We use KL-WNMF as a comparison method [64]. KL-WNMF is a commonly used gradient method that reduces the KL-based cost in Equation (4.6) by multiplicative updates. Although faster NMF methods, such as ALS [62] and ADMM [51], have been developed, they are just as fast as a multiplicative update when the target rank is small [113]. Moreover, as we will show in below, KL-WNMF converges within only 2–4 iterations in our experiments. In addition, the em-algorithm needs more iterations since it minimizes the cost function indirectly. Thus, these techniques are considered ineffective for speeding up rank-1 KL-WNMF. This is why we only compared A1GM with simple KL-WNMF.

We implemented KL-WNMF by referring to the original paper [64]. The stopping criterion of KL-WNMF follows the implementation of the standard NMF in scikit-learn [99]. The initial values of KL-WNMF are determined by sampling from a uniform continuous distribution from 0 to 1. All methods are implemented in `Julia 1.6`. We used `BenchmarkTools` to measure the running time [17]. Experiments were conducted on Ubuntu 20.04.1 with a single core of 2.1GHz Intel Xeon CPU Gold 5218 and 128GB of memory.

Synthetic datasets

Missing values in the top right corner We prepared synthetic matrices $\mathbf{X} \in \mathbb{R}^{N \times N}$ and their weights $\mathbf{\Phi} \in \{0,1\}^{N \times N}$. We assumed that each input weight $\mathbf{\Phi}$ is in the form of Equation (4.7) with $L = 0$. We measured the running time to obtain rank-1 decomposition of $\mathbf{X}$ with varying the matrix size N. Figure 4.8(a) shows that A1GM is an order of magnitude faster than the existing gradient method. The number of iterations of the existing method until convergence was between 2 and 4. A1GM just applies the closed formula in Theorem 4.1 to parts of input matrices.

Random grid-like missing values We also prepared synthetic matrices and its binary weight matrices $\mathbf{\Phi} \in \{0,1\}^{N \times N}$. We assumed that every input weight matrix $\mathbf{\Phi}$ is grid-like, and we set the ratio of missing values to be 5 percent. We measured the running time of A1GM to complete the best rank-1 missing NMF compared with KL-WNMF by varying the matrix size N. Figure 4.8(b) shows that our method is always faster than the gradient method. The number of iterations of the existing method required for convergence was between 2 and 4. Note that in these datasets, A1GM does not need to increase missing values.

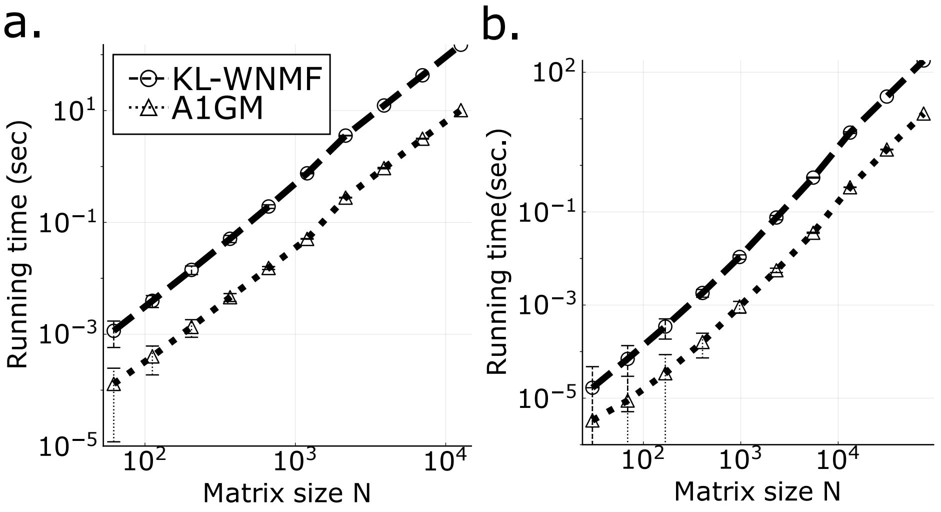


Figure 4.8 Running time comparison of the proposed method A1GM (triangle, dots line) and KL-WNMF (circle, dashed line) with respect to the matrix size N. (a) Missing values are at the top right corner. (b) Missing value positions are grid-like. We plot the mean ± S.D. of five trials.

Real datasets

We used 20 real datasets. We downloaded tabular datasets that have missing values from the Kaggle databank[2] or UCI dataset.[3] If a dataset contains negative values, we converted them to their absolute values. Zero values in a matrix were replaced with the average value of the matrix to make them all positive. We evaluate the relative error as

$$D_{\mathbf{\Phi}}(\mathbf{X}, \mathrm{A1GM}(\mathbf{X}))/D_{\mathbf{\Phi}}(\mathbf{X}, \mathrm{WNMF}(\mathbf{X})),$$

where $\mathrm{WNMF}(\mathbf{X})$ and $\mathrm{A1GM}(\mathbf{X})$ are the rank-1 reconstructed matrices by KL-WNMF and A1GM, respectively, and the binary weight matrix $\mathbf{\Phi}$ indicates locations of missing values of $\mathbf{X}$. We also compared the relative running time of A1GM to KL-WNMF.

The results are summarized in Table 4.1. In the table, the column `increase rate` means the ratio of the number of missing values after addition in A1GM to the original number of missing values. If `increase rate` is 1, it means that the location of missing values of the dataset is originally grid-like. For such datasets, it is theoretically guaranteed that our method A1GM always provides the best rank-1 missing NMF, which minimizes the KL divergence in Equation (4.6). It is reasonable that the reconstructed matrix by KL-WNMF and that by A1GM are the same since the cost function (4.6) is convex in $\boldsymbol{w}$ and $\boldsymbol{h}$. The number of iterations of the existing method required for convergence was between 2 and 4 for real datasets.

We can see that A1GM is much faster than KL-WNMF for all the datasets. Moreover, the relative error remains low even if missing values of datasets are not grid-like for most of

[2] https://www.kaggle.com/datasets
[3] https://archive.ics.uci.edu/ml/

Table 4.1 Performance of A1GM compared to KL-WNMF on 20 real datasets.

DataSet	size	# missing	increase rate	relative error	relative runtime
IndianPop	(24,13)	1	1	1	0.19784
Autompg	(398, 8)	6	1	1	0.12957
DailySunSpot	(73718, 9)	3247	1	1	0.12845
CaliforniaHousing	(20640, 9)	207	1	1	0.11821
MTSLibrary	(1533078, 4)	1247722	1	1	0.18327
BigMartSaleForecas	(8522, 5)	1463	1	1	0.12699
BoardGameGeekData	(101375, 17)	21	1	1	0.14625
CreditCardApproval	(590, 7)	25	1.92	1.0018	0.12212
HumanResourceAnaly	(14999, 7)	519	1.96	1.0168	0.11858
concretemiss	(1030,9)	99	2	1.0010	0.11108
heartdisease	(303, 14)	6	2	1	0.12259
lungcancer	(32, 57)	5	2	1.0001	0.13803
PerthHousePrice	(33656, 14)	16585	2.61	1.0004	0.15382
SleepData	(62, 8)	12	2.75	1.0211	0.18208
HCVData	(615,11)	31	4.19	1.0068	0.11246
arrhythmia	(452, 280)	408	4.71	1.0148	0.11387
Bostonhousing	(506, 14)	120	5.60	1.0030	0.10970
LifeExpectancyData	(2938, 19)	2563	7.04	5.7983	0.09577
HCCSurvivalDataSet	(165, 50)	826	8.36	3.2898	0.07113
wiki4HE	(913, 53)	1995	18.1	1.2363	0.06626

datasets. In some real data, missing values are likely to be biased towards a particular row or column. As a result, they become grid-like by just adding a small number of missing values. In these cases, our proposed method can conduct rank-1 missing NMF rapidly with competitive errors to KL-WNMF. By contrast, a large amount of information is lost after the increasing missing value step for some datasets (large `increase rates`). As a result, our method is not suitable for obtaining an accurately reconstructed rank-1 matrix, even though it is much faster than the existing method.

Datasets for A1GM We provide the list of the source of the real datasets in Table 4.2.

4.6 Conclusion

In this chapter, we have derived the closed analytical formula of the best rank-1 NMMF. To obtain this formula, we have used the conservation law in m-projection in information geometry by modeling matrices as a log-linear model on a poset. Using the formula, we have developed a novel method of rank-1 NMF for missing data, called A1GM. We have shown that A1GM obtains the best rank-1 NMF when missing values are located in a grid-like manner. When the location of missing values is not grid-like, we increase the number of missing values so that they become grid-like, to which again we can use the closed formula of the best rank-1 NMMF. We empirically show that A1GM, which is not based on the gradient descent method, is more efficient than the existing gradient method for rank-1 missing NMF.

Table 4.2 Real detaset details for A1GM

DataSet	# zeros	# negatives	URL
IndianPop	0	19	https://onl.bz/bcu3qCR
Autompg	0	0	https://www.kaggle.com/uciml/autompg-dataset
DailySunSpot	16994	3247	https://www.kaggle.com/abhinand05/daily-sun-spot-data-1818-to-2019
CaliforniaHousing	0	20640	https://www.kaggle.com/harrywang/housing?select=housing.csv
MTSLibrary	200932	0	https://www.kaggle.com/sharthz23/mts-library?select=interactions.csv
BigMartSaleForecas	526	0	https://www.kaggle.com/arashnic/big-mart-sale-forecast?select=train.csv
BoardGameGeekData	520624	21	https://www.kaggle.com/mandshaw/games-0918
CreditCardApproval	797	0	https://www.kaggle.com/redwuie/credit-card-approval?select=train.csv
HumanResourceAnaly	27510	0	https://www.kaggle.com/cezarschroeder/human-resource-analytics-dataset
concretemiss	1390	0	https://www.kaggle.com/datasets/izemdemirci/concrete-missing
heartdisease	1149	0	https://archive.ics.uci.edu/ml/datasets/Heart+Disease
lungcancer	107	0	https://archive.ics.uci.edu/ml/datasets/Lung+Cancer
PerthHousePrice	0	33656	https://www.kaggle.com/syuzai/perth-house-prices
SleepData	0	0	https://www.kaggle.com/mathurinache/sleep-dataset
HCVData	0	0	https://archive.ics.uci.edu/ml/datasets/HCV+data
arrhythmia	67256	14250	https://archive.ics.uci.edu/ml/datasets/Arrhythmia
Bostonhousing	812	0	https://www.kaggle.com/altavish/boston-housing-dataset
LifeExpectancyData	3385	0	https://www.kaggle.com/kumarajarshi/life-expectancy-who
HCCSurvivalDataSet	2416	0	https://archive.ics.uci.edu/ml/datasets/HCC+Survival
wiki4HE	1801	0	https://archive.ics.uci.edu/ml/datasets/wiki4HE

As noted, our method has two main limitations. First, because our modeling uses a log-linear model, we cannot handle zero values in a matrix. Second, the performance of A1GM is not expected to be convincing if there are a huge number of missing values. NMMF and NMF for missing data have been extended to tensors as NMTF [122] and WNTF [95], respectively. Generalization of our study to tensors is an interesting area for future work.

5 Many-Body Approximation for Nonnegative Tensors

We present an alternative approach to decompose non-negative tensors, called *many-body approximation*. Traditional decomposition methods assume low-rankness in the representation, resulting in difficulties in global optimization and target rank selection. We avoid these problems by energy-based modeling of tensors, where a tensor and its mode correspond to a probability distribution and a random variable, respectively, and many-body approximation is performed on it by taking the *interaction between variables, i.e. modes*, into account. Our model can be globally optimized in polynomial time in terms of the KL divergence minimization, which is empirically faster than low-rank approximations while keeping comparable reconstruction error. Furthermore, we visualize interactions between modes as *tensor networks* and reveal a nontrivial relationship between many-body approximation and low-rank approximation.

Tensors are generalization of vectors and matrices. Data in various fields such as neuroscience [33], bioinformatics [82], signal processing [24], and computer vision [96] are often stored in the form of tensors, and features are extracted from them. *Tensor decomposition* and its non-negative version [108] are popular methods that extract features by approximating tensors by the sum of products of smaller tensors. These smaller tensors are often called factors. It usually tries to minimize the difference between the tensor reconstructed from obtained smaller tensors and an original tensor, called the reconstruction error.

In most of tensor decomposition approaches, a *low-rank structure* is typically assumed, where a given tensor is approximated by a linear combination of a small number of bases. Such decomposition requires the following two information. First, it requires the structure, which specifies the type of decomposition such as CP decomposition [54] and Tucker decomposition [126]. In recent years, *tensor networks* [23] have been introduced, which can intuitively and flexibly design the structure including tensor train decomposition [93], tensor ring decomposition [140], and tensor tree decomposition [89]. Second, it requires the number of bases used in the decomposition, often called the rank. Since larger ranks increase the capability of the model while increasing the computational cost, the user is required to find the appropriate rank in this tradeoff problem (See figure 5.1). Since the above tensor decomposition via minimization of the reconstruction error is

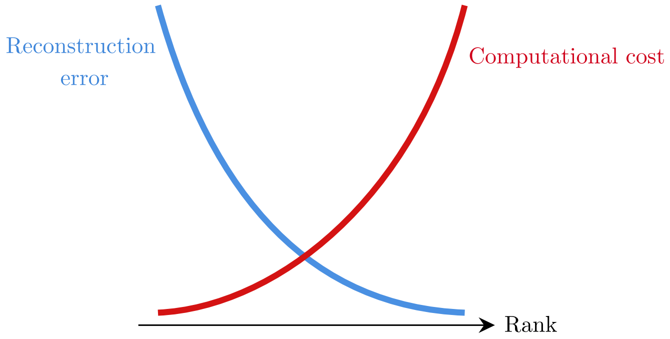


Figure 5.1 In tensor decomposition, larger target ranks increase the capability of the model and reduce reconstruction errors, while increasing computational cost. We need to face this trade-off problem to set the appropriate rank.

non-convex, which causes initial value dependence [68, Chapter 3], the problem of finding an appropriate setting of the low-rank structure is highly nontrivial in practice as it is hard to locate the cause if the decomposition does not perform well. As a result, to find proper structure and rank, the user often needs to perform decomposition multiple times with various settings, which is time and memory consuming.

Instead of the low-rank structure that has been the focus of attention in the past, in this paper, we propose a novel formulation of tensor decomposition, called *many-body approximation*, that focuses on the relationship among modes of tensors. We determine the structure of decomposition based on the existence of the interactions between modes. The proposed method requires only the decomposition structure naturally determined by the interactions between the modes and does not require the rank value, which traditional decomposition methods also require and often suffer to determine.

To describe interactions between modes, we follow the standard strategy in statistical mechanics that uses an energy function $\mathcal{H}(\cdot)$ to treat interactions and considers the corresponding distribution $\exp\left(-\mathcal{H}(\cdot)\right)$. This model is known to be an energy-based model in machine learning [72] and is exploited in tensor decomposition as Legendre decomposition [118]. Technically, it parameterizes a tensor as a discrete probability distribution and reduces the number of parameters by enforcing some of them to be zero in optimization. We explore this energy-based approach further and discover the family of parameter sets that represent interactions between modes in the energy function $\mathcal{H}(\cdot)$. How to choose non-zero parameters in Legendre decomposition has been an open problem, and we firstly address this problem and propose many-body approximation as a special case of Legendre decomposition. Moreover, although Legendre decomposition is not factorization of tensors in general, our proposal always offers factorization, which can reveal patterns in tensors. Since the advantage of Legendre decomposition is inherited to our proposal, many-body approximation can be achieved by convex optimization that globally minimizes the Kullback–Leibler (KL) divergence [70].

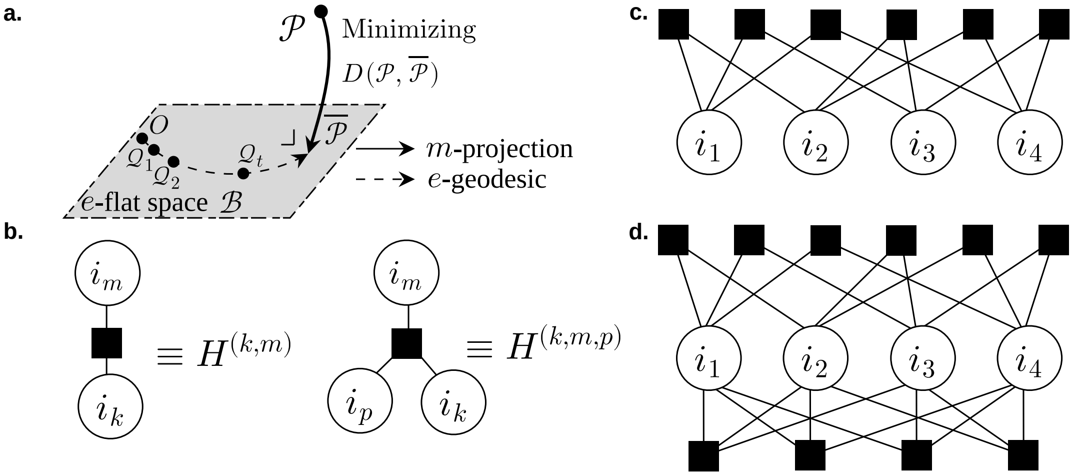


Figure 5.2 (a) An illustration of optimization of Legendre decomposition. Interaction representations corresponding to (c) Equation (5.9) and (d) Equation (5.10). In interaction representations, edges through ■ between modes mean existing interaction. For simplicity, we abbreviate one-body interactions in the diagrams.

Furthermore, we introduce a way of representing tensor interactions, which visualizes the presence or absence of interactions between modes. We discuss the correspondence between our representation and the tensor network and point out that an operation called coarse-grained transformation [76], in which multiple tensors are viewed as a new tensor, reveals unexpected relationship between the proposed method and existing methods such as tensor ring and tensor tree decomposition.

We summarize our contribution as follows:

- By focusing on the interaction between modes of tensors, we introduce an alternative rank-free tensor decomposition, many-body approximation. This decomposition is realized by convex optimization.
- We present a way of describing tensor many-body approximation, interaction representation, a diagram that shows interactions within a tensor. This diagram can be transformed into tensor networks, which tells us the relationship between many-body approximation and existing low-rank approximation.
- We empirically show that many-body approximation is faster than low-rank approximation with competitive reconstruction errors.

Our proposal, tensor many-body approximation, is based on the formulation of Legendre decomposition for tensors. We first review Legendre decomposition and its optimization in Section 5.1. We introduce interactions between modes and its visual representation to prepare for many-body approximation in Section 5.2. Using interactions between modes, we define many-body approximation in Section 5.3. Finally, we transform the interaction representation into a tensor network and point out the connection between many-body approximation and existing low-rank decomposition methods in Section 5.4.

5.1 Legendre Decomposition and Its Optimization

In the following discussion, we consider D-order non-negative tensors whose size is $(I_1, \dots, I_D)$. We assume the sum of all elements in $\mathcal{P}$ is 1 for simplicity, while this assumption can be eliminated using the general property of Kullback–Leibler (KL) divergence, $\lambda D(\mathcal{P}, \mathcal{Q}) = D(\lambda\mathcal{P}, \lambda\mathcal{Q})$, for any real number λ.

5.1.1 Reminder to Legendre Decomposition

Legendre decomposition is a method to decompose a non-negative tensor by regarding the tensor as a discrete distribution and representing it with a limited number of parameters. We describe a non-negative tensor $\mathcal{P}$ using natural parameters $\boldsymbol{\theta} = (\theta_{2,1,\dots,1}, \dots, \theta_{I_1,\dots,I_D})$ and its energy function $\mathcal{H}$ as

$$\mathcal{P}_{i_1,\dots,i_D} = \exp\left(-\mathcal{H}_{i_1,\dots,i_D}\right), \quad \mathcal{H}_{i_1,\dots,i_D} = -\sum_{i'_1=1}^{i_1} \cdots \sum_{i'_D=1}^{i_D} \theta_{i'_1,\dots,i'_D}, \tag{5.1}$$

where $\theta_{1,\dots,1}$ has a role of normalization. Here it is clear that a tensor corresponds to a distribution whose sample space is its index set; that is, the value of each element is regarded as the probability of realizing the corresponding index [117].

As we can see in Equation (5.1), we can uniquely identify tensors from natural parameters $\boldsymbol{\theta}$. We can compute the natural parameter $\boldsymbol{\theta}$ from a given tensor as

$$\theta_{i_1,\dots,i_D} = \sum_{i'_1=1}^{I_1} \cdots \sum_{i'_D=1}^{I_D} \mu_{i_1,\dots,i_D}^{i'_1,\dots i'_D} \log \mathcal{P}_{i'_1,\dots,i'_D} \tag{5.2}$$

using the Möbius function $\mu : \Omega_D \times \Omega_D \to \{-1, 0, +1\}$, where Ω_D is the set of indices, defined inductively as follows:

$$\mu_{i_1,\dots,i_D}^{i'_1,\dots,i'_D} = \begin{cases} 1 & \text{if } i_d = i'_d \text{ for all } d \in [D], \\ -\prod_{d=1}^{D} \sum_{j_d=i_d}^{i'_d-1} \mu_{i_1,\dots,i_D}^{j_1,\dots j_D} & \text{else if } i_d \leq i'_d \text{ for all } d \in [D], \\ 0 & \text{otherwise.} \end{cases}$$

Due to space limitations, we represent one of the two arguments by superscripts and the other by subscripts. We provide the Möbius function as a more general form in Equation (2.3). The above modeling for non-negative tensors is an instance of the log-linear model on posets [117].

Since distribution described by Equation (5.1) belongs to the exponential family, we can also identify each tensor by expectation parameters $\boldsymbol{\eta} = (\eta_{2,1,\ldots,1}, \ldots, \eta_{I_1,\ldots,I_D})$ using the M¨bius inversion formula as

$$\eta_{i_1,\ldots,i_D} = \sum_{i'_1=i_1}^{I_1} \cdots \sum_{i'_D=i_D}^{I_D} \mathcal{P}_{i'_1,\ldots,i'_D}, \quad \mathcal{P}_{i_1,\ldots,i_D} = \sum_{i'_1=1}^{I_1} \cdots \sum_{i'_D=1}^{I_D} \mu_{i_1,\ldots,i_D}^{i'_1,\ldots,i'_D} \eta_{i'_1,\ldots,i'_D}. \tag{5.3}$$

where $\eta_{1,\ldots,1} = 1$ because of normalization. See more general form of the M¨bius inversion formula in Equation (2.4). Since distribution is determined by specifying either θ-parameters or η-parameters, they form two coordinate systems called the θ-coordinate system and the η-coordinate system, respectively. By using the dual flatness, and orthogonality of these coordinate systems, Legendre decomposition achieves convex optimization as shown in the following.

5.1.2 Optimization

Legendre decomposition approximates a tensor by setting some θ values to be zero, which corresponds to dropping some parameters for regularization. Let B be the set of indices of θ parameters that are not imposed to be 0. Then Legendre decomposition coincides with a projection of a given nonnegative tensor $\mathcal{P}$ onto the subspace $\mathcal{B} = \{\boldsymbol{\theta} \mid \theta_{i_1,\ldots,i_D} = 0 \text{ if } (i_1,\ldots,i_D) \notin B\}$.

Let us consider projection of a given tensor $\mathcal{P}$ onto $\mathcal{B}$. The space of probability distributions is not a Euclidean space. Therefore, it is necessary to consider geometry of probability distributions, which is studied in information geometry. It is known that a subspace with linear constraints on natural parameters θ is flat, called e-flat [4, Chapter 2]. The subspace $\mathcal{B}$ is e-flat, meaning that the logarithmic combination, or called e-geodesic, $\mathcal{R} \in \{(1-t)\log\mathcal{Q}_1 + t\log\mathcal{Q}_2 - \phi(t) \mid 0 < t < 1\}$ of any two points $\mathcal{Q}_1, \mathcal{Q}_2 \in \mathcal{B}$ is included in the subspace $\mathcal{B}$, where $\phi(t)$ is a normalizer. There is always a unique point $\overline{\mathcal{P}}$ on the e-flat subspace that minimizes the KL divergence from any point $\mathcal{P}$.

$$\overline{\mathcal{P}} = \operatorname*{argmin}_{\mathcal{Q};\mathcal{Q}\in\mathcal{B}} D(\mathcal{P},\mathcal{Q}) \tag{5.4}$$

This projection is called the m-projection. The m-projection onto a e-flat subspace is a convex optimization. We define two vectors $\boldsymbol{\theta}^B = (\theta_b)_{b\in B}$ and $\boldsymbol{\eta}^B = (\eta_b)_{b\in B}$. We write as $|\mathcal{B}|$ the number of elements in these vectors since it is equal to the cardinality of $\mathcal{B}$. The derivative of the KL divergence and the Hessian matrix $G \in \mathbb{R}^{|\mathcal{B}|\times|\mathcal{B}|}$ are given as

$$\frac{\partial}{\partial\boldsymbol{\theta}^B} D(\mathcal{P},\mathcal{Q}) = \boldsymbol{\eta}^B - \hat{\boldsymbol{\eta}}^B, \quad G_{u,v} = \eta_{\max(i_1,j_1),\ldots,\max(i_D,j_D)} - \eta_{i_1,\ldots,i_D}\eta_{j_1,\ldots,j_D} \tag{5.5}$$

where $\boldsymbol{\eta}^B$ and $\hat{\boldsymbol{\eta}}^B$ are the expectation parameters of $\mathcal{Q}$ and $\mathcal{P}$, respectively, and $u = (i_1,\ldots,i_D), v = (j_1,\ldots,j_D) \in B$. This is a particular case in Equation (2.5). This matrix

G is also known as the negative Fisher information matrix. Using gradient descent with second-order derivative, we can update $\boldsymbol{\theta}^B$ in each iteration t as

$$\boldsymbol{\theta}^B_{t+1} = \boldsymbol{\theta}^B_t - G^{-1}(\boldsymbol{\eta}^B_t - \hat{\boldsymbol{\eta}}^B) \tag{5.6}$$

The distribution $\mathcal{Q}_{t+1}$ is calculated from the updated natural parameters $\boldsymbol{\theta}_{t+1}$. This step finds a point $\mathcal{Q}_{t+1} \in \mathcal{B}$ that is closer to the destination $\overline{\mathcal{P}}$ along with the e-geodesic from $\mathcal{Q}_t$ to $\overline{\mathcal{P}}$. We can also calculate the expected value parameters $\boldsymbol{\eta}_{t+1}$ from the distribution. By repeating this process until convergence, we can always find the globally optimal solution satisfying Equation (5.4). This procedure is illustrated in Figure. 5.2(a).

5.2 Interaction and Its Representation of Tensors

In this subsection, we introduce interactions between modes and its visual representation to prepare for many-body approximation. The following discussion enables us to intuitively describe relationships between modes and formulate our novel rank-free tensor decomposition.

First we introduce n-body parameters, which is a generalized concept of one-body and many-body parameters in Chapter 3 (See Definition 3.1).

Definition 5.1 n-body Parameter

Let n of a n-body parameter be the number of non-one indices.

For example, $\theta_{1,2,1,1}$ is a one-body parameter, $\theta_{4,3,1,1}$ is a two-body parameter and $\theta_{1,2,4,3}$ is a three-body parameter. We regard the normalize factor $\theta_{1,\dots,1}$ as a 0-body parameter. We also use the following notation for n-body parameters:

$$\theta^{(k)}_{i_k} = \theta_{1,\dots,1,i_k,1,\dots,1}, \quad \theta^{(k,m)}_{i_k,i_m} = \theta_{1,\dots,1,i_k,1,\dots,1,i_m,1,\dots,1}, \quad \theta^{(k,m,p)}_{i_k,i_m,i_p} = \theta_{1,\dots,i_k,\dots,i_m,\dots,i_p,\dots,1},$$

for $n = 1, 2,$ and 3, respectively. Also, we also introduce n-th order energy.

Definition 5.2 n-th Order Energy

The n-th order energy for a tensor and its θ-parameters is given as

$$H^{(l_1,\dots,l_n)}_{i_{l_1},\dots,i_{l_n}} = -\sum_{i'_{l_1}=2}^{i_{l_1}} \cdots \sum_{i'_{l_n}=2}^{i_{l_n}} \theta^{(l_1,\dots,l_n)}_{i'_{l_1},\dots,i'_{l_n}}. \tag{5.7}$$

We write the energy function $\mathcal{H}$ with n-body parameters as

$$\mathcal{H}_{i_1,\cdots,i_D} = H_0 + \sum_{m=1}^{D} H^{(m)}_{i_m} + \sum_{m=1}^{D}\sum_{k=1}^{m-1} H^{(k,m)}_{i_k,i_m} + \sum_{m=1}^{D}\sum_{k=1}^{m-1}\sum_{p=1}^{k-1} H^{(p,k,m)}_{i_p,i_k,i_m} + \cdots + H^{(1,\ldots,D)}_{i_1,\ldots,i_D} \tag{5.8}$$

For simplicity, we suppose that $1 \leq l_1 < l_2 < \cdots < l_n \leq D$ holds. We set $H_0 = -\theta_{1,\ldots,1}$. The number of Σ in l-th term in Equation (5.8) is ${}_D\mathrm{C}_l$. We say that an n-body interaction exists between modes $l_1, \ldots, l_n$ if there are indices $i_{l_1}, \ldots, i_{l_n}$ satisfying $H^{(l_1,\ldots,l_n)}_{i_{l_1},\ldots,i_{l_n}} \neq 0$.

The first term H_0 in Equation (5.8) is called the normalized factor or the partition function. The terms $H^{(k)}$ are called bias in machine learning and magnetic field or self-energy in statistical physics. The terms $H^{(k,m)}$ are called the weight of the Boltzmann machine in machine learning and two-body interaction or electron-electron interaction in physics.

To visualize the existence of interactions within a tensor, we newly introduce a diagram called *interaction representation*, which is inspired by factor graphs in graphical modeling [11, Chapter 8]. The graphical representation of the product of tensors is widely known as tensor networks. However, displaying the relations between the modes of a tensor as a factor graph is our novel approach. We represent the n-body interaction as a black square, ■, connected with n modes. We describe examples of the two-body interaction between modes (k, m) and the three-body interaction among modes (k, m, p) in Figure 5.2(b). Combining these interactions, the energy function including all two-body interactions is shown in Figure 5.2(c), and the energy function including all two-body and three-body interactions is shown in Figure 5.2(d) for $D = 4$.

This visualization allows us to intuitively understand the relationship between modes of tensors. For simplicity, we abbreviate one-body interactions in the diagrams, while we always assume them. Once interaction representation is given, we can determine the corresponding decomposition of tensors.

In the following section, we reduce some of n-body interactions, that is, $H^{(l_1,\ldots,l_n)}_{i_{l_1},\ldots,i_{l_n}} = 0$, by fixing each parameter $\theta^{(l_1,\ldots,l_n)}_{i_{l_1},\ldots,i_{l_n}} = 0$ for all indices $(i_{l_1}, \ldots, i_{l_n}) \in \{2, \ldots, I_{l_1}\} \times \cdots \times \{2, \ldots, I_{l_n}\}$.

5.3 Many-body Approximation for Non-negative Tensors

Our proposed method, many-body approximation, approximate a given tensor with assuming the existence of dominant interactions between the modes of the tensor and ignoring other interactions. Since this operation can be understood as setting some natural parameters of the distribution to be zero, it can be achieved by convex optimization through the theory of Legendre decomposition. As we see below, approximated tensors are represented without the summation symbol $\sum$. This property is different from existing low-rank approximations except for rank-1 approximation.

As an example, we consider approximations of a nonnegative tensor $\mathcal{P}$ by tensors represented in Figure 5.2(c) and Figure 5.2(d).

If all energies greater than 2nd-order or those greater than 3rd-order in Equation(5.8) are ignored, that is, $H^{(l_1,\ldots,l_n)}_{i_{l_1},\ldots,i_{l_n}} = 0$ for $n > 2$ or $n > 3$, $\mathcal{P}$ is approximated as follows:

$$\mathcal{P}_{i_1,i_2,i_3,i_4} \simeq \mathcal{P}^{\leq 2}_{i_1,i_2,i_3,i_4} = \mathbf{X}^{(1,2)}_{i_1,i_2}\mathbf{X}^{(1,3)}_{i_1,i_3}\mathbf{X}^{(1,4)}_{i_1,i_4}\mathbf{X}^{(2,3)}_{i_2,i_3}\mathbf{X}^{(2,4)}_{i_2,i_4}\mathbf{X}^{(3,4)}_{i_3,i_4}, \tag{5.9}$$

$$\mathcal{P}_{i_1,i_2,i_3,i_4} \simeq \mathcal{P}^{\leq 3}_{i_1,i_2,i_3,i_4} = \chi^{(1,2,3)}_{i_1,i_2,i_3}\chi^{(1,2,4)}_{i_1,i_2,i_4}\chi^{(1,3,4)}_{i_1,i_3,i_4}\chi^{(2,3,4)}_{i_2,i_3,i_4}, \tag{5.10}$$

where each small matrix and tensor on the right-hand side is represented as

$$\mathbf{X}^{(k,m)}_{i_k,i_m} = \frac{1}{\sqrt[6]{Z}} \exp\left(\frac{1}{3}H^{(k)}_{i_k} + H^{(k,m)}_{i_k,i_m} + \frac{1}{3}H^{(m)}_{i_m}\right),$$

$$\chi^{(k,m,p)}_{i_k,i_m,i_p} = \frac{1}{\sqrt[4]{Z}} \exp\left(\frac{H^{(k)}_{i_k} + H^{(m)}_{i_m} + H^{(p)}_{i_p}}{3} + \frac{1}{2}H^{(k,m)}_{i_k,i_m} + \frac{1}{2}H^{(m,p)}_{i_m,i_p} + \frac{1}{2}H^{(k,p)}_{i_k,i_p} + H^{(k,m,p)}_{i_k,i_m,i_p}\right).$$

The partition function, or the normalization factor, is given as $Z = \exp(-\theta_{1,1,1,1})$, which do not depend on indices (i_1, i_2, i_3, i_4). Each $X^{(k,m)}$ (resp. $\chi^{(k,m,p)}$) is a factorized representation for the relationship between k-th and m-th (resp. k-th, m-th and p-th) modes. Although our model can be transformed into a linear model by taking the logarithm, our convex formulation enables us to find the optimal solution more stable than traditional linear low-rank based nonconvex approaches. Since we do not impose any low-rankness, factorized representations, e.g., $X^{(k,m)}$ and $\chi^{(k,m,p)}$, can be full-rank matrices or tensors.

We provide the definition of m-body approximation as follows:

Definition 5.3 m-body Approximation for Non-negative Tensors

For a given tensor $\mathcal{P} \in \mathbb{R}^{I_1\times\cdots\times I_D}_{\geq 0}$, its m-body approximation $\mathcal{P}^{\leq m} \in \mathbb{R}^{I_1\times\cdots\times I_D}_{\geq 0}$ is the optimal tensor in Equation (5.4) for $\mathcal{B} = \{\, \theta \mid \theta_{i_1,\ldots,i_D} = 0 \text{ if } (i_1,\ldots,i_D) \notin B \,\}$, where B is the set of indices of $n(\leq m)$-body parameters.

We show constraints of m-body approximation in Figure 5.3 for $D = 4$. Interestingly, the two-body approximation for a non-negative tensor with $I_1 = \cdots = I_D = 2$ is equivalent to approximating the empirical distribution with the fully connected Boltzmann machine.

Although we can find the analytical solution for one-body approximation by Theorem 3.1, we need numerical calculation to conduct $n(> 1)$-body approximation. See the optimization procedure in Section 5.1.2.

In Boltzmann machines, we usually consider binary (two-level) variables and their second order energy. In our proposal, we consider multi-level D variables, each of which can take a natural number from 1 to I_d for $d \in [D]$. Moreover, higher-order interactions among them are allowed. Therefore our proposal is a multi-level extension of Boltzmann machines with higher-order interaction, where each node of Boltzmann machines corresponds to the tensor mode.

In the above discussion, we consider many-body approximation with all n-body parameters, while our formulation allows us to use only a part of n-body interactions as shown in the following. We consider the situation where only one-body interaction and two-body interaction between modes $(d, d+1)$ exist for all $d \in [D]$ ($D+1$ implies 1 for simplicity). Figure 5.5(a) shows the interaction representation of the approximated tensor. As we can confirm by substituting 0 for $H^{(k,l)}_{i_k,i_l}$ if $l \neq k+1$, we can describe the approximated tensor as the element-wise cyclic product of matrices,

$$\mathcal{P}_{i_1,\dots,i_D} \simeq \mathcal{P}^{\mathrm{cyc}}_{i_1,\dots,i_D} = \mathbf{X}^{(1)}_{i_1,i_2}\mathbf{X}^{(2)}_{i_2,i_3}\dots\mathbf{X}^{(D)}_{i_D,i_1} \tag{5.11}$$

where

$$\mathbf{X}^{(k)}_{i_k,i_{k+1}} = \frac{1}{\sqrt[D]{Z}} \exp\left(\frac{1}{2}H^{(k)}_{i_k} + H^{(k,k+1)}_{i_k,i_{k+1}} + \frac{1}{2}H^{(k+1)}_{i_{k+1}}\right). \tag{5.12}$$

The partition function is given as $Z = \exp(-\theta_{1,\dots,1})$, which does not depend on indices $(i_1, \dots, i_D)$. When the tensor $\mathcal{P}$ is approximated by $\mathcal{P}^{\mathrm{cyc}}$, the set B contains only all one-body parameters and two-body parameters $\theta^{(d,d+1)}_{i_d,i_{d+1}}$ for $d \in [D]$. We call this approximation *cyclic two-body approximation* since the order of indices in Equation (5.11) is cyclic. Here we provide the formal description:

> **Definition 5.4** Cyclic Two-body Approximation for Non-negative Tensors
>
> For a given tensor $\mathcal{P} \in \mathbb{R}^{I_1\times\dots\times I_D}_{\geq 0}$, its cyclic two-body approximation $\mathcal{P}^{\mathrm{cyc}} \in \mathbb{R}^{I_1\times\dots\times I_D}_{\geq 0}$ is the optimal tensor in Equation (5.4) where the set B contains only all one-body parameters and two-body parameters $\theta^{(d,d+1)}_{i_d,i_{d+1}}$ for $d \in [D]$.

The constraint of the cyclic two-body approximation is shown in Figure 5.4. We show the connection between cyclic two-body approximation and existing tensor ring decomposition in Section 5.4.

Interaction and Conservation Laws Let $\boldsymbol{\eta}$ be the expectation parameter of the original tensor and $\hat{\boldsymbol{\eta}}$ be the expectation parameter after many-body approximation. From Proposition 3.8, $\boldsymbol{\eta}^B = \hat{\boldsymbol{\eta}}^B$ holds. In many-body approximation, defining the existing interactions determines which natural parameters of the projected tensor become zero and, at the same time, which expectation parameters are conserved in the approximation. As described in Section 2.2.1, the conserved expectation parameters lead to the invariance of the summation in each fiber or slice in the approximation.

Trivial Examples of Many-body Approximation Note that D-body approximation always provides the same tensor as input, that is, $\mathcal{P}^{\leq D} = \mathcal{P}$. Zero-body approximation is also a trivial example. It always provide a tensor whose all values are the same, that is $\mathcal{P}^{\leq 0}_{i_1,\dots,i_D} = Z^{-1}$ where $Z = \prod_{d=1}^{D} I_d$. It corresponds to a uniform discrete distribution. Since the value of $\mathcal{P}^{\leq 0}_{i_1,\dots,i_D}$ does not depend on the index $(i_1, \dots, i_D)$, we cannot define its interaction representation.

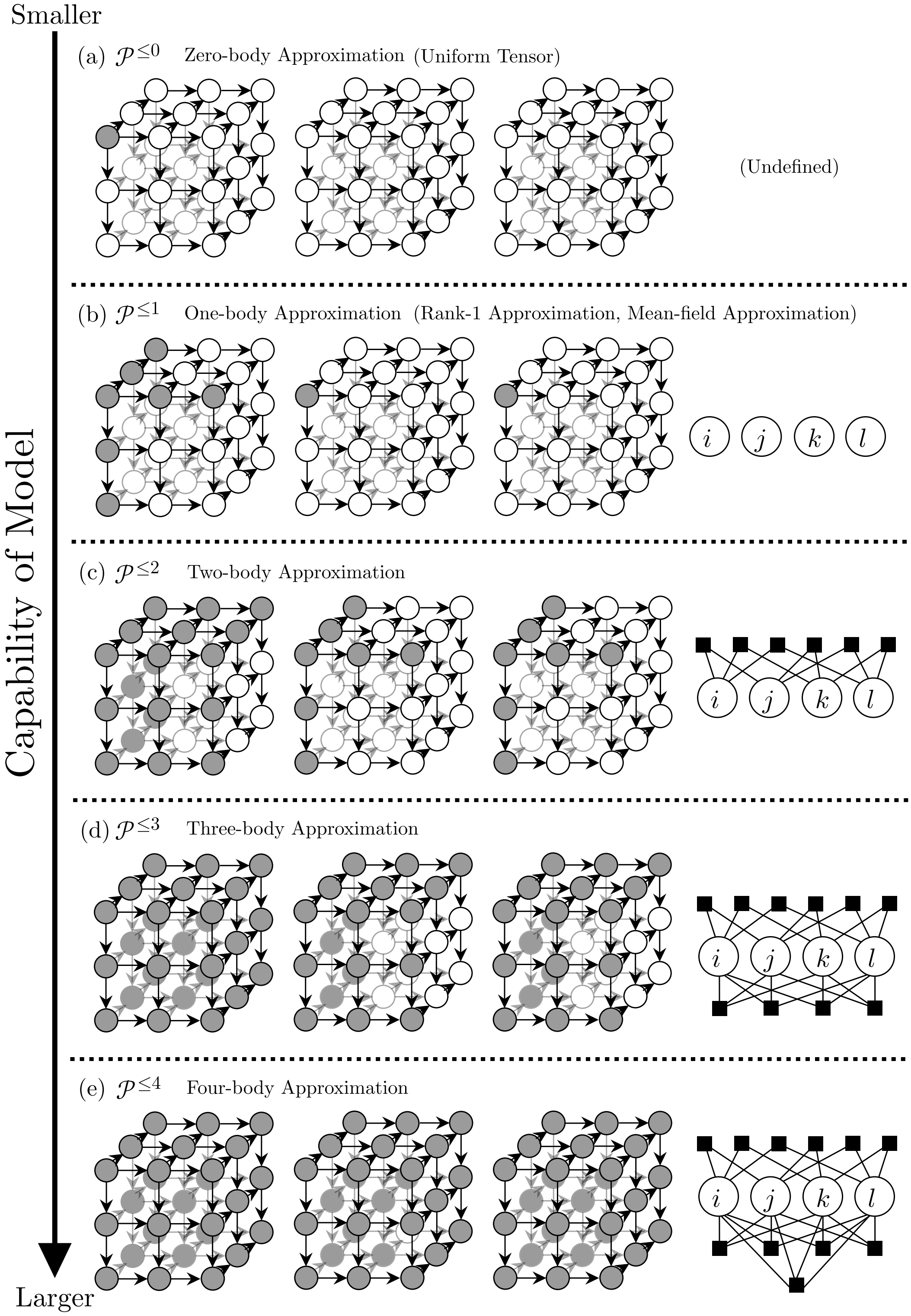

Smaller
Capability of Model
Larger
(a) $\mathcal{P}^{\leq 0}$ Zero-body Approximation (Uniform Tensor)
(Undefined)
(b) $\mathcal{P}^{\leq 1}$ One-body Approximation (Rank-1 Approximation, Mean-field Approximation)
i
j
k
l
(c) $\mathcal{P}^{\leq 2}$ Two-body Approximation
i
j
k
l
(d) $\mathcal{P}^{\leq 3}$ Three-body Approximation
i
j
k
l
(e) $\mathcal{P}^{\leq 4}$ Four-body Approximation
i
j
k
l

Figure 5.3 Constraints and interaction representations for n-body approximation of a 4th-order tensor $\mathcal{P}_{\geq 0}^{3\times3\times3\times3}$ for $n = 0, 1, \ldots, 4$. Only θ-parameters on gray-colord nodes can have non-zero values after each approximation. The four-body approximation for 4th-order tensor will not reduce θ-parameters. Since one-body approximation is equivalent to rank-1 approximation, n-body approximation is a generalization of rank-1 approximation. We abbreviate one-body interactions in the diagrams. The m-projection does not change the values of the expectation parameters on the gray nodes.

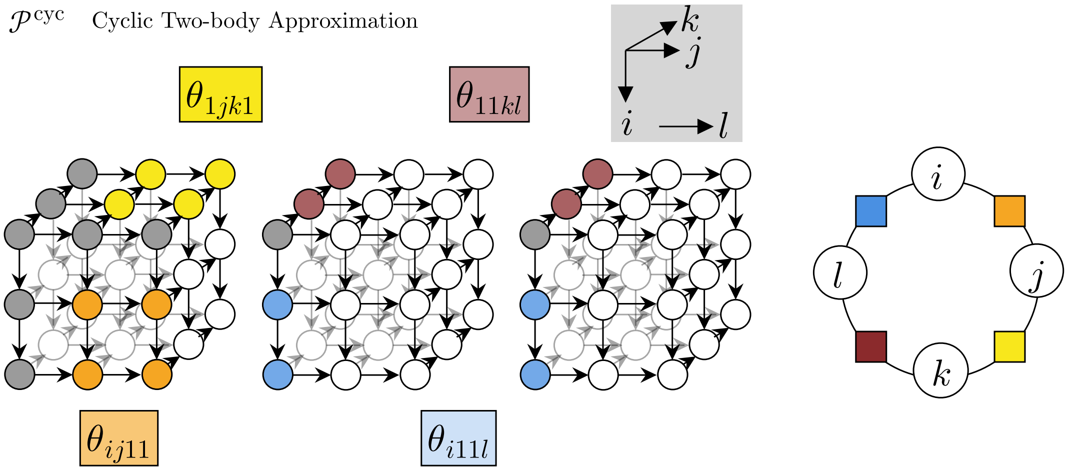


Figure 5.4 The constraint and interaction representation for cyclic two-body approximation of 4th-order tensors $\mathcal{P}_{\geq 0}^{3\times3\times3\times3}$. Only θ-parameters on colored nodes can have non-zero values after the approximation. Interactions, ■, and corresponding two-body parameters are filled in the same color. One-body parameters are on gray-colored nodes.

5.4 Connection to Tensor Network

Our tensor interaction representation is an undirected graph that focuses on the relationship between modes. In contrast, tensor networks, which are well known as diagrams that focus on smaller tensors after decomposition, represent a tensor as an undirected graph, whose nodes correspond to matrices or tensors and edges to summation over a mode in tensor products [23].

We provide interesting examples in our representation that can be converted to tensor networks, which implies our representation has a tight connection to tensor networks. For the conversion, we use a hyper-diagonal tensor Ω defined as $\Omega_{ijk} = \delta_{ij}\delta_{jk}\delta_{ki}$, where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. The tensor Ω is often represented by • in tensor networks. In the community of tensor networks, the tensor Ω appears in the CNOT gate and a special case of the Z spider [92]. The tensor network in Figure 5.5(a) represents the following formula

$$\prod_{d=1}^{D}\left(\sum_{j_d}\sum_{l_d}\mathbf{X}^{(d)}_{l_d,j_{d+1}}\mathcal{I}_{j_{d+1},i_{d+1},l_{d+1}}\right), \tag{5.13}$$

where $j_{D+1} = j_1, i_{D+1} = i_1, l_{D+1} = l_1$. Substituting the definition of $\mathcal{I}$ in Equation (5.13), we realize that the tensor network corresponds to Equation (5.11).

A remarkable finding is that the converted tensor network representation of cyclic two-body approximation and the tensor network of tensor ring decomposition, whose tensor network is shown in Figure 5.5(b), have the similar structure in common, despite their different modeling. If we consider the region enclosed by the dotted line in Figure 5.5(a)

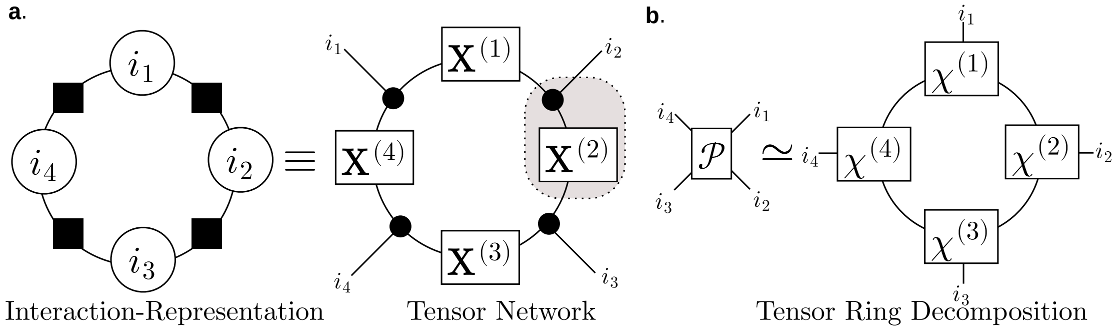


Figure 5.5 (a) Interaction representation of an example of cyclic two-body approximation and its transformed tensor network for $D = 4$. Each tensor is enclosed by a square and each mode is enclosed by a circle. A black circle • is a hyper diagonal tensor. Edges through ■ between modes mean interaction existence. (b) Tensor network of tensor ring decomposition.

as a new tensor, the tensor network of the cyclic two-body approximation coincides with the tensor network of the tensor ring decomposition shown in Figure 5.5(b). This operation, in which multiple tensors are regarded as a new tensor in a tensor network, is called coarse-graining transformation [34].

Formally, cyclic two-body approximation coincides with tensor ring decomposition with a specific constraint as described below. Non-negative tensor ring decomposition approximates a given tensor $\mathcal{P} \in \mathbb{R}_{\geq 0}^{I_1 \times \cdots \times I_D}$ with D core tensors $\chi^{(1)}, \chi^{(2)}, \dots, \chi^{(D)}$ with $\chi^{(d)} \in \mathbb{R}_{\geq 0}^{R_{d-1} \times I_d \times R_d}$ for each $d \in [D]$ as

$$\mathcal{P}_{i_1,\dots,i_D} \simeq \overline{\mathcal{P}}_{i_1,\dots,i_D} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_D=1}^{R_D} \chi^{(1)}_{r_D,i_1,r_1} \chi^{(2)}_{r_1,i_2,r_2} \cdots \chi^{(D)}_{r_{D-1},i_D,r_D} \tag{5.14}$$

where $(R_1, \dots, R_D)$ is called tensor ring rank. The decomposition is described in Figure 5.5(b). The cyclic two-body approximation also approximates the tensor $\mathcal{P}$ in the form of Equation (5.14), imposing an additional constraint that each core tensor $\chi^{(d)}$ is decomposed as

$$\chi^{(d)}_{r_{d-1},i_d,r_d} = \sum_{m_d=1}^{I_d} \mathbf{X}^{(d)}_{r_{d-1},m_d} \mathcal{I}_{m_d,i_d,r_d} \tag{5.15}$$

for each $d \in [D]$, where $\mathcal{I}_{ijk} = \delta_{ij}\delta_{jk}\delta_{ki}$. We assume $r_0 = r_D$ for simplicity. We obtain Equation (5.11) by substituting Equation (5.15) into Equation (5.14).

This constraint enables us to perform convex optimization. This means that we find a subclass that can be solved by convex optimization in tensor ring decomposition, which has suffered from the difficulty of non-convex optimization. In addition, this is simultaneously a subclass of two-body approximation.

From Kronecker's delta δ, $r_d = i_d$ holds in Equation (5.15), thus $\chi^{(d)}$ is a tensor with the size (I_{d-1}, I_d, I_d). Tensor ring rank after the cyclic two-body approximation is $(I_1, \dots, I_D)$ since the size of core tensors coincides with tensor ring rank.

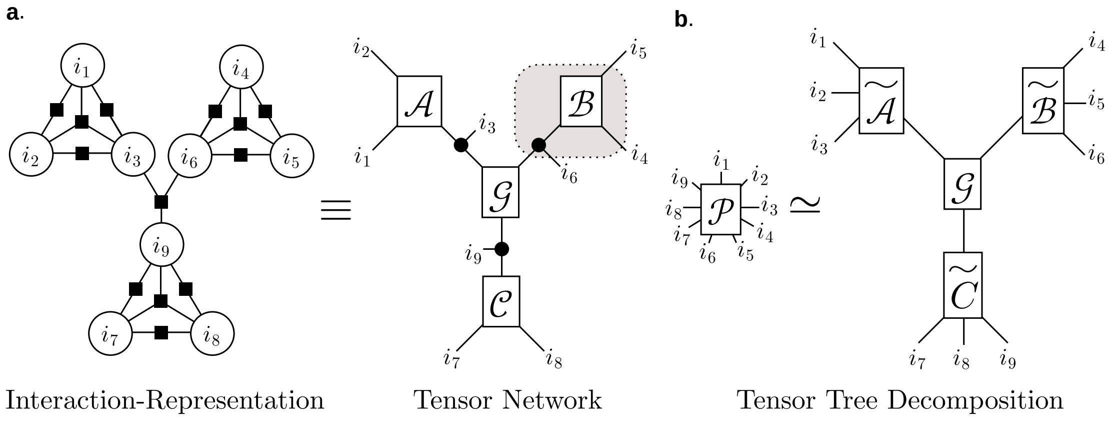


Figure 5.6 (a) Interaction representation corresponding to Equation (5.17) and its transformed tensor network for $D = 9$. We abbreviate one-body interactions in the diagram. (b) Tensor network of a variant of tensor tree decomposition. Each mode is enclosed by a circle. Each tensor is enclosed by a square. A black circle, $\bullet$, is a hyper diagonal tensor. Edges through ■ between modes mean existing interaction.

This result firstly reveals the relationship between Legendre decomposition and low-rank approximation via tensor networks.

Comparing the number of parameters The number of elements of an input tensor is $I_1 \times I_2 \times \cdots \times I_D$. After the cyclic two-body approximation, the number of parameters is given as

$$|\mathcal{B}| = 1 + \sum_{d=1}^{D}(I_d - 1) + \sum_{d=1}^{D}(I_d - 1)(I_{d+1} - 1) \tag{5.16}$$

where we assume $I_{D+1} = I_1$. The first term is for a normalizer, the second is the number of one-body parameters, and the final term is the number of two-body parameters. In contrast, in the tensor ring decomposition with the target rank $(R_1, \ldots, R_D)$, the number of parameters is given as $|\mathcal{R}| = \sum_{d=1}^{D} R_d I_d R_{d+1}$. The ratio of the number of parameters of these two methods $|\mathcal{B}|/|\mathcal{R}|$ is proportional to I/R^2 if we assume $R_d = R$ and $I_d = I$ for all $d \in [D]$ for simplicity. Therefore, when the target rank is small and the size of the input tensor is large, the proposed method has more parameters than the tensor ring decomposition.

5.5 Other Example of Many-body Approximation and Its Tensor Network

In the same way, we can find a correspondence between another example of many-body approximation and the existing low-rank approximation. For $D = 9$, we consider three-body and two-body interactions among (i_1, i_2, i_3), (i_4, i_5, i_6), and (i_7, i_8, i_9) and three-body approximation among (i_3, i_6, i_9). We provide the interaction representation of the target energy function in Figure 5.6(a). In this approximation, the decomposed tensor can be described as

$$\mathcal{P}_{i_1,\dots,i_9} = \mathcal{A}_{i_1,i_2,i_3}\mathcal{B}_{i_4,i_5,i_6}\mathcal{C}_{i_7,i_8,i_9}\mathcal{G}_{i_3,i_6,i_9}. \tag{5.17}$$

In the same way in the case of the cyclic two-body approximation, we can convert the interaction representation to a tensor network, as described in Figure 5.6(a). A tensor network of tensor tree decomposition in Figure 5.6(b) emerges when the region enclosed by the dotted line is replaced with a new tensor (shown with tilde) in Figure 5.6(a). Such tensor tree decomposition is used in generative modeling [19], computational chemistry [90] and quantum many-body physics [109].

As we have seen above, by transforming tensor interaction representation to tensor networks and applying coarse-graining, we can reveal the relationship between tensor many-body approximations and low-rank approximations.

5.6 Many-body Approximation as Generalization of Mean-field Approximation

It has been already pointed out that any tensor $\mathcal{P}$ can be represented by vectors $\boldsymbol{x}^{(d)} \in \mathbb{R}^{I_d}$ for $d \in [D]$ as

$$\mathcal{P}_{i_1,\dots,i_D} = x^{(1)}_{i_1} x^{(2)}_{i_2} \dots x^{(D)}_{i_D} \tag{5.18}$$

if and only if all $n(\geq 2)$-body θ-parameters are 0 [40]. The right-hand side is equal to the Kronecker product of D vectors $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(D)}$, and therefore this approximation is equivalent to the rank-1 approximation since the rank of the tensor that can be represented by the Kronecker product is always 1, which is also known to correspond to mean-field approximation. In this study, we propose many-body approximation by relaxing the condition for the mean-field approximation that ignores $n(\geq 2)$-body interactions. Therefore many-body approximation is generalization of rank-1 approximation and mean-field approximation. We show the relationship between these approximations in Figure 5.3.

5.7 Computational Complexity

We analyze the computational complexity of many-body approximation. In many-body approximation, the overall complexity is dominated by the update of θ, which includes matrix inversion of G.

Algorithm 3: Many-body Approximation

```
ManyBodyApproximation($\mathcal{T}$, $B$)
    $s \leftarrow$ Total sum of $\mathcal{T}$.
    Obtain normalized tensor $\mathcal{P} \leftarrow \mathcal{T}./s$    // "./" denotes element-wise division
    Compute $\hat{\boldsymbol{\eta}}$ of $\mathcal{P}$ using Equation (5.3).
    Initialize $\boldsymbol{\theta}^{B}_{t=1}$                            // e.g. $\theta_b = 0$ for all $b \in B$
    $t \leftarrow 1$
    repeat
        Compute $\mathcal{Q}_t$ using the current parameter $\boldsymbol{\theta}^{B}_{t}$ with Equation (5.1).
        Compute $\boldsymbol{\eta}^{B}_{t}$ from $\mathcal{Q}_t$ using Equation (5.3).
        Compute the Fisher information matrix $G$ using Equation (5.5).
        $\boldsymbol{\theta}^{B}_{t+1} \leftarrow \boldsymbol{\theta}^{B}_{t} - G^{-1}(\boldsymbol{\eta}^{B}_{t} - \hat{\boldsymbol{\eta}}^{B})$
        $t \leftarrow t + 1$
    until $||\boldsymbol{\eta}^{B}_{t} - \hat{\boldsymbol{\eta}}^{B}|| < \epsilon$          // We set $\epsilon = 10^{-5}$ in our implementation;
    $\overline{\mathcal{T}} \leftarrow \mathcal{Q}_t .* s$                        // ".*" denotes element-wise multiplication
    return $\overline{\mathcal{T}}$
```

The complexity of computing the inverse of an $n \times n$ matrix is $O(n^3)$. Therefore, the computational complexity of many-body approximation is $O(\gamma|\mathcal{B}|^3)$, where γ is the number of iterations.

This complexity can be reduced if we reshape tensors so that the size of each mode becomes small. For example, let us consider a 3rd-order tensor whose size is (J^2, J^2, J^2) and its cyclic two-body approximation. In this case, the time complexity is $O(\gamma J^{12})$ since it holds that $|\mathcal{B}| \propto J^4$ (See Equation (5.16)). In contrast, if we reshape the input tensor to a 6-order tensor whose size is (J, J, J, J, J, J), the time complexity becomes $O(\gamma J^6)$ since it holds that $|\mathcal{B}| \propto J^2$.

This technique of reshaping a tensor into a larger-order tensor is used practically not only in the proposed method but also in various methods based on tensor networks, such as tensor ring decomposition [83].

5.8 Experiments for Many-body Approximation

As seen in Section 5.4, many-body approximation has a close connection to low-rank approximation. For example, in a tensor ring decomposition, if we impose that decomposed factors can be represented as products with hyper-diagonal tensors $\mathcal{I}$, this decomposition is equivalent to a cyclic two-body approximation (see Figure 5.5). Therefore, to examine our conjecture that cyclic two-body approximation is as capable of approximating as tensor ring decomposition, we empirically examine the efficiency and effectiveness of cyclic two-body approximation compared with tensor ring decomposition. As baselines, we use five existing methods of non-negative tensor ring decomposition, NTR-APG, NTR-

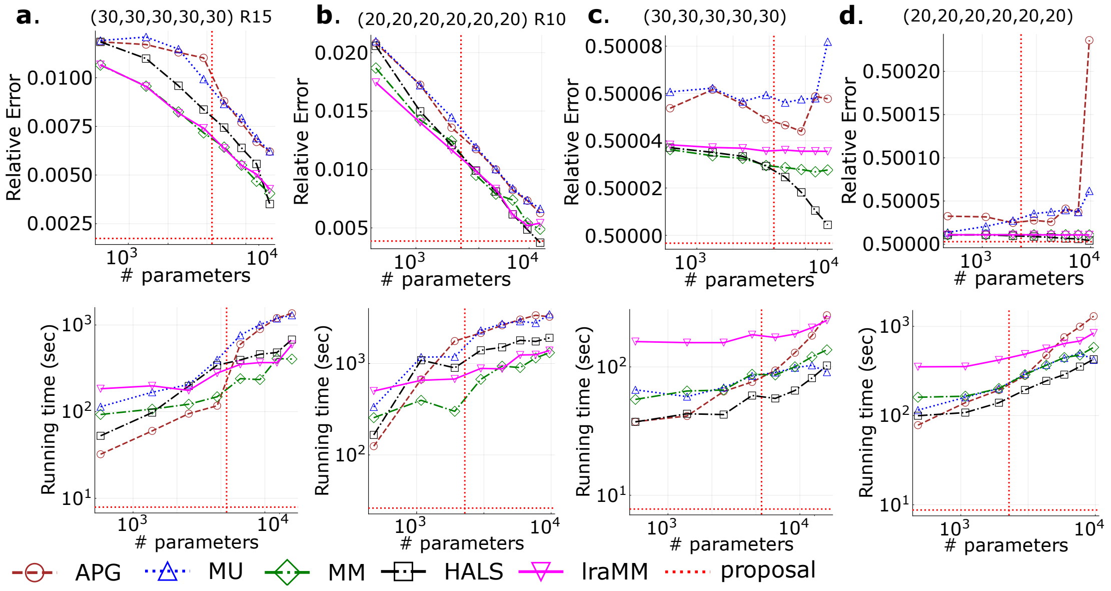


Figure 5.7 (a)(b) Results for low ring rank tensor. (c)(d) Results for tensors sampled from uniform distribution. The vertical red dotted line is $|\mathcal{B}|$ in Equation (5.16).

HALS, NTR-MU, NTR-MM and NTR-lraMM [135, 136]. These methods minimize the reconstruction error defined with the Frobenius norm by the gradient method.

We evaluate the approximation performance by the relative error

$$\frac{\|\mathcal{T} - \overline{\mathcal{T}}\|_{\mathrm{F}}}{\|\mathcal{T}\|_{\mathrm{F}}} \tag{5.19}$$

for an input tensor $\mathcal{T}$ and a reconstructed tensor $\overline{\mathcal{T}}$ with the Frobenius norms $\|\cdot\|_{\mathrm{F}}$. Since all the existing methods are based on nonconvex optimization, we plot the best score (minimum relative error) among 5 restarts with random initialization. In contrast, the score of our method is obtained by a single run as it is convex optimization and such restarts are fundamentally unnecessary. We compare the total running time of them.

5.8.1 Results on Synthetic Data

We performed experiments on four synthetic datasets. The first two are synthetic data with low tensor ring rank. This setting is often used in evaluation of tensor ring decomposition. We create D core tensors of size $R \times I \times R$ by sampling from uniform distribution. Then a tensor with the size I^D and the tensor ring rank $(R, \dots, R)$ is obtained by the product of these D tensors. Results for $R = 15, D = 5, I = 30$ are shown in Figure 5.7(a), and those for $R = 10, D = 6, I = 20$ in Figure 5.7(b). Relative error and computation time are plotted with gradually increasing the target rank of the tensor ring decomposition, which is compared to the score of our method, plotted as the cross point of horizontal and vertical red dotted lines. Please note that our method does not have the concept of the rank, thus the score of our method is invariant to changes of the target rank unlike other

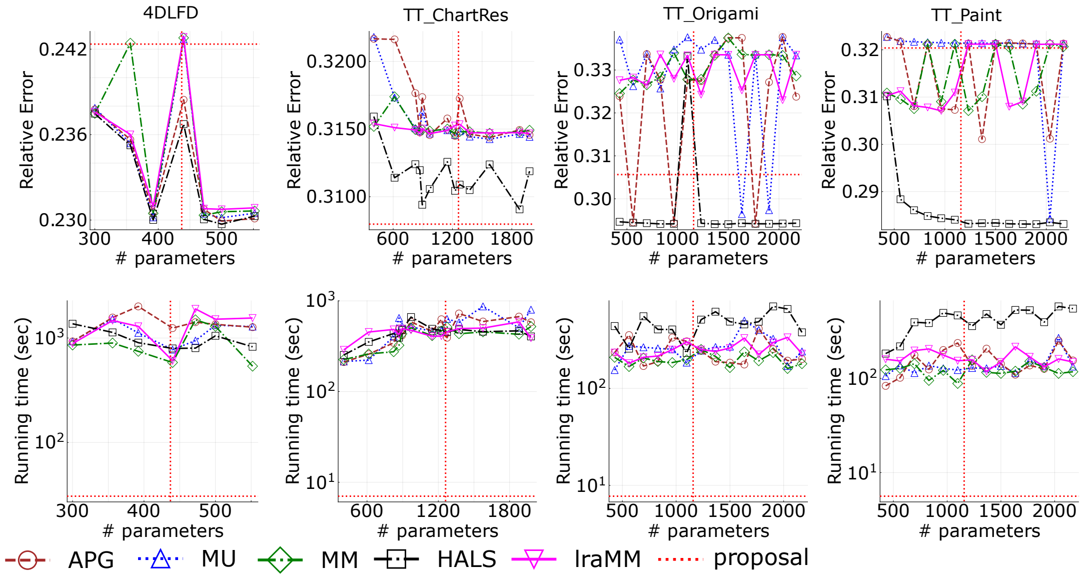


Figure 5.8 Experimental results for real datasets. The vertical red dotted line is $|\mathcal{B}|$ in Equation (5.16).

methods. If the cross point of red dotted lines is lower than other lines, the proposed method is better than other methods.

In addition to the above case in which we assumed the low-rankness, we also generated synthetic datasets without such an assumption. We created a tensor of size 30^5 and a tensor of size 20^5 by sampling from uniform distribution and performed the same experiment. Results are shown in Figure 5.7(c) and Figure 5.7(d). In all experiments, the proposed method is superior to comparison partners in both efficiency and effectiveness. It should be noted that the relative error of the proposed method is smaller even when the target rank of the tensor ring decomposition is large and the number of parameters is several times larger than the proposed method.

For all experiments on synthetic datasets, we change the target ring-rank as $(R, \dots, R)$ for $R = 2, 3, \dots, 9$ for baseline methods.

5.8.2 Results on Real Data

Next, we evaluate our method on real data. `4DLFD` is a 9-order tensor, which is produced from 4D Light Field Dataset [57, 47, 77]. `TT_ChartRes`, `TT_Origami` and `TT_Paint` are 7-order tensors, which is produced from TokyoTech Hyperspectral Image Dataset [88, 87]. Each tensor has been reshaped to reduce the computational complexity. See the following dataset details. The proposed method is always faster than baselines with keeping the competitive relative errors. In baseline methods, a slight change of the target rank can induce a significant increase of the reconstruction error due to the nonconvex nature of them. We eliminate the instability of non-negative tensor ring decomposition by our convex formulation.

Dataset detail for real dataset `4DLFD` is originally a $(9, 9, 512, 512, 3)$ tensor, which is produced from 4D Light Field Dataset described in [57]. Its license is Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. We use `dino` images and their depth and disparity map in training scenes. We concatenate them to produce a tensor. We reshaped the tensor as $(6, 8, 6, 8, 6, 8, 6, 8, 12)$. For baseline methods, we chose the target ring-rank as $(2, 3, 2, 2, 2, 2, 2, 2, 2)$, $(2, 3, 2, 2, 3, 2, 2, 3, 2)$, $(2, 2, 2, 2, 2, 2, 2, 2, 5)$, $(2, 5, 2, 2, 5, 2, 2, 2, 2)$, $(2, 2, 2, 2, 2, 2, 2, 2, 7)$, $(2, 2, 2, 2, 3, 2, 2, 2, 7)$, $(2, 2, 2, 2, 2, 2, 2, 2, 9)$.
`TT_ChartRes` is originally a $(736, 736, 31)$ tensor, which is produced from TokyoTech 31-band Hyperspectral Image Dataset. We use `ChartRes.mat`. We reshaped the tensor as $(23, 8, 4, 23, 8, 4, 31)$. For baseline methods, we chose the target ring-rank as $(2, 2, 2, 2, 2, 2, 2)$ $(2, 2, 2, 2, 2, 2, 5)$, $(2, 2, 2, 2, 2, 2, 8)$, $(3, 2, 2, 3\ \ , 2, 2, 5)$, $(2, 2, 2, 2, 2, 2, 9)$, $(3, 2, 2, 3,, 2, 2, 6)$, $(4, 2, 2, 2, 2, 2, 6)$, $(3, 2, 2, 4, 2, 2, 8)$, $(3, 2, 2, 3,\ 2, 2, 9)$, $(3, 2, 2, 3, 2, 2, 10)$, $(3, 2, 2, 3, 2, 2, 12)$, $(3, 2, 2, 3, 2, 2, 15)$, $(3, 2, 2, 3, 2, 2, 16)$.
`TT_Origami` and `TT_Paint` are originally $(512, 512, 59)$ tensors, which are produced from TokyoTech 59-band Hyperspectral Image Dataset. We use `Origami.mat` and `Paint.mat`. In `TT_Origami`, 0.0016% of elements were negative, hence we preprocessed all elements of `TT_Origami` by subtracting -0.000764, the smallest value in `TT_Origami`, to make all elements non-negative. We reshaped the tensor as $(8, 8, 8, 8, 8, 8, 59)$. For baseline methods, we chose the target ring-rank as $(2, 2, 2, 2, 2, 2, R)$ for $R = 2, 3, \ldots, 15$.

5.8.3 Implementation detail

We describe the implementation details of methods in the following.

Proposed method Our method is implemented in Julia 1.8. We use a natural gradient method for cyclic two-body approximation. The natural gradient method uses the inverse of the Fisher information matrix to perform second-order optimization in a non-euclidean space. For non-normalized tensors, we conduct the following procedure. First, we compute the total sum of elements of an input tensor. Then, we normalize the tensor. After that, we conduct Legendre decomposition for the normalized tensor. Finally, we get the product of the result of the previous step and the total sum we compute initially. The termination criterion is the same as the original implementation of Legendre Decomposition by [118], that is, it terminates if $||\boldsymbol{\eta}_t^B - \hat{\boldsymbol{\eta}}^B|| < 10^{-5}$, where $\boldsymbol{\eta}_t^B$ is the expectation parameters on t-th step and $\hat{\boldsymbol{\eta}}^B$ is the expectation parameters of an input tensor, which are defined in Section 5.1. The overall procedure is described in Algorithm 3. Note that this algorithm is based on Legendre decomposition by [118].

Baseline methods We implemented baseline methods by translating MATLAB code provided by the authors into Julia code for fair comparison. As we can see from their original papers, NTR-APG, NTR-HALS, NTR-MU, NTR-MM and NTR-lraMM have an inner and outer loop to find a local solution. We repeat the inner loop 100 times. We stop the outer loop when the difference between the relative error of the previous and the current iteration is less than 10e-4. NTR-MM and NTR-lraMM require diagonal parameters matrix Ξ. We define $\Xi = \omega\mathbf{I}$ where $\mathbf{I}$ is an identical matrix and $\omega = 0.1$. The NTR-lraMM method performs low-rank approximation to the matrix obtained by mode expansion of an input tensor. The target rank is set to be 20. This setting is the default setting in the provided code. The initial positions of baseline methods were sampled from uniform distribution on $(0, 1)$.

Environment Experiments were conducted on Ubuntu 20.04.1 with a single core of 2.1GHz Intel Xeon CPU Gold 5218 and 128GB of memory.

5.9 Conclusion

We propose *many-body approximation* for tensors, which decomposes tensors with focusing on the relationship between modes represented by an energy-based model. It approximates tensors by ignoring the energy corresponding to some interactions, which can be viewed as generalization of mean-field approximation that considers only one-body interactions. Our novel formulation enables us to achieve convex optimization of the model, while the existing approaches based on the low-rank structure are non-convex. Furthermore, we introduce a way of visualize interactions between modes, called interaction representation, to see activated interactions between modes. We have established transformation between our representation and tensor networks, which reveals the non-trivial connection between many-body approximation and the classical tensor low-rank tensor decomposition.

Conclusion 6

This chapter summarizes this dissertation, focusing on the relationship among proposed methods. This study proposes three dimensionality reduction methods, LTR, A1GM, and many-body approximation. LTR reduces the Tucker rank of tensors rapidly, A1GM obtains an approximate solution of rank-1 NMF with missing values based on the closed formula of rank-1 NMMF, and many-body approximation decomposes tensors into element-wise product formats. While LTR and A1GM are faster methods for traditional low-rank approximation, many-body approximation is a novel convex dimensionality reduction naturally derived from the information geometric analysis of rank-1 approximation. This study works across three fields: linear algebra, which deals with tensors and matrices; information geometry, which is the geometry of probability distributions; and energy-based models, which is inspired by statistical mechanics.

In Section 6.1, we review the unique characteristics and strengths of our study. Next, in Section 6.2, we provide some interesting remaining questions that were not clarified in this study. We also discuss the limitations of this study in Section 6.3. Finally, we describe future directions of our study in Section 6.4.

6.1 Characteristics and Strengths of This Study

Here we summarize the characteristics and strengths of our study, focusing on the following three points.

Model flatness guarantees the uniqueness and convexity of learning.

Formulating these dimensionality reduction approaches as projections onto e-flat model manifolds provided convex reconstruction errors to be optimized. We designed e-flat model manifolds by mapping the input data to a probability distribution and describing the set of dimensionality-reduced arrays with a linear condition of natural parameters of the distribution. For example, in LTR, the subspace of low-Tucker-rank is not e-flat, but by introducing the bingo rule, the e-flat subspace is extracted from the space of low-Tucker-rank tensors as described in Chapter 3. In addition, we can perform LTR and A1GM without gradient-based methods. This is because the trick of describing the low-rank condition in a dual-flat coordinate system and using a conservation law for the parameters allows us to know the projection destination in a closed form.

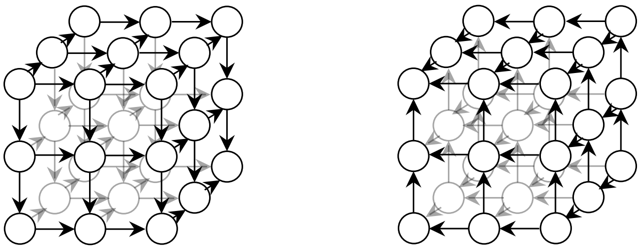


Figure 6.1 We could construct an equivalent theory by introducing either structure for tensors. Which ordered structure should we have introduced?

Unified view for dimensionality reductions with various data structure.

To discuss the dimensionality reduction of matrices, tensors, and multiple matrices in a unified manner, we used a log-linear model on posets that allow for flexible modelling. We can discuss dimensionality reduction to various data structures by designing posets to correspond to data structures. After designing proper posets, these algorithms perform m-projection in common, where some natural parameters become zero. In summary, our approach formulates tasks as convex optimizations by taking the input data structure as proper posets and describing the constraints of the task in a dual-flat coordinate system.

Information geometric analysis derives a novel convex dimensionality reduction.

In this study, we regard non-negative tensors as joint probability distributions and analyze their rank-1 decompositions from an information geometric viewpoint. As a result, we interpreted rank-1 decompositions of tensors as mean-field approximations as described in Chapter 3. The mean-field approximation is an approximation that makes all modes independent. As a natural extension of this approximation, we defined many-body approximation in which several modes are independent in Chapter 5. Since we can describe the constraints of tensors after many-body approximation with linear conditions of natural parameters, we succeeded in proposing a fast convex dimensionality reduction based on the natural gradient method. As seen above, this study not only analyzed traditional low-rank approximation by information geometry and developed efficient algorithms, but also led to a novel dimensionality reduction task that overcomes the problems of non-convexity and rank tuning that traditional low-rank approximation faced.

6.2 Questions to Be Answered

Here we describe two challenges that could not be resolved in this study yet can be interesting research topics in the future.

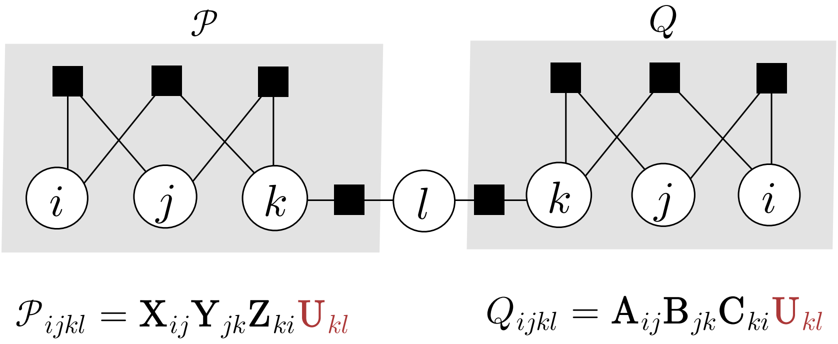

$\mathcal{P}$
$\mathcal{Q}$
i
j
k
l
k
j
i
$\mathcal{P}_{ijkl} = \mathbf{X}_{ij}\mathbf{Y}_{jk}\mathbf{Z}_{ki}\mathbf{U}_{kl}$
$\mathcal{Q}_{ijkl} = \mathbf{A}_{ij}\mathbf{B}_{jk}\mathbf{C}_{ki}\mathbf{U}_{kl}$

Figure 6.2 A sketch of a concept for interactions between tensors. Two tensors, $\mathcal{P} \in \mathbb{R}^{I\times J\times K\times L}$ and $\mathcal{Q} \in \mathbb{R}^{I\times J\times K\times L}$, have a two-body interaction through mode-l. It implies that they have the following structure $\mathcal{P}_{ijkl} = \mathbf{X}_{ij}\mathbf{Y}_{jk}\mathbf{Z}_{ki}\mathbf{U}_{kl}$ and $\mathcal{Q}_{ijkl} = \mathbf{A}_{ij}\mathbf{B}_{jk}\mathbf{C}_{ki}\mathbf{U}_{kl}$ where $\mathbf{U}$ is a shared factor.

How to choose model manifold?

In Chapter 3, the concept of bingo is introduced to solve the low-Tucker-rank approximation as a convex optimization problem. Determining positions of bingos is a model selection problem. Therefore, a criterion for selecting a bingo is required. In addition, it remains to be discussed which interactions should be activated in the many-body approximation. Both of the above challenges are about how the constraints on the model manifold should be determined. Note that each model manifold is always e-flat, regardless of the choice of interactions and bingos.

How should we introduce ordered structures to data?

We use a log-linear model on posets to map data to a discrete probability distribution. For this purpose, in Section 3.4 and 4.3, we introduced partially ordered structures that did not originally exist in data. The correspondence between data and partially ordered structures is not unique. This study did not answer the question, “Why do we map data to this partially ordered structure?”

Since it does not exist in the original data, the closed solution formulas for the best rank-1 approximation in Theorems 3.1 and 4.1 and numerical solutions for many-body approximation should be independent from the choice of partially ordered structures. For example, with either partially ordered structure shown in Figure 6.1, we could discuss rank-1 approximations and interactions between the modes. However, we have not yet answered the interesting question of which structure is more suitable for our discussions.

6.3 Limitations of the Proposed Methods

We discuss limitations of our proposed methods in this section.

We assume input data are non-negative and dense.

We developed efficient algorithms by regarding data as joint distributions to apply the theory of information geometry, which is the key idea throughout our research. Due to the correspondence between data and joint probabilities, we naturally impose non-negativity on input data. Therefore, our proposed methods cannot treat tensor data that include negative values.

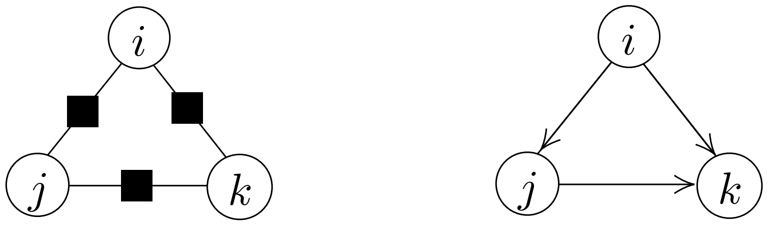
It has been pointed out that larger-order tensors often become sparse. Although various strategies to treat sparse data have been developed [94, 97, 110], we do not consider them in our research. Our proposed methods can treat only non-negative dense tensors. We also need more discussion to develop decompositions with popular constraints such as symmetry [13, 14] and definiteness [2, 80], which could be our future works.

We optimize the KL divergence instead of the least-squares error.

Our proposed methods, LTR, A1GM, and many-body approximation, optimize the reconstruction error defined with the KL divergence. It is because we use the optimization theory in information geometry. However, the LS error is the most popular cost function in tensor and matrix decompositions. Although we empirically evaluated proposed methods with the LS error in Sections 3.11 and 5.8, we did not provide any theoretical bound about the LS error.

DAG is not learnable.

We assume that the data structure is static. This means that, once we introduce a handcrafted poset to map data to probabilities, we suppose that the partial order structure does not change. This strategy does not accommodate applications where the data structure is dynamic, such as dynamic analysis of source-code [10]. Although some applications directly obtain DAGs, e.g., causal inference [45], we cannot directly apply the proposed framework to these tasks.



$$\mathcal{P}_{ijk} = \mathbf{X}_{ij}\mathbf{Y}_{jk}\mathbf{Z}_{ki} \qquad \mathcal{P}_{ijk} = p(k|i)p(j|i)p(k|j)$$

Figure 6.3 Undirected interaction (left) and directed interaction (right) for 3rd-order tensors.

6.4 Future Directions

The flexibility of a log-linear model on posets, our proposed manifold learning with physics strategies, and interaction representation for tensors lead us to new research directions. In this section, we show some examples of them.

6.4.1 From Array to More General Data Structure

This dissertation handled matrices, tensors, multiple matrices, and matrices with missing values. We think we can additionally formulate manifold learning for more general data structures, i.e., attributed graphs, based on the flexibility of the log-linear model on posets. Attributed graphs are frequently used in bioinformatics, medical science [132] and chemoinformatics [125]. By describing graph data with nonnegative attributes at nodes by the natural parameter θ and then reducing some of them, a convex manifold learning for these data can be formulated. That is, our proposed low-rank and many-body approximations for multidimensional arrays defined by natural parameter reduction can potentially be extended to graph data. Of course, it is not trivial how to define ranks or interactions naturally for such a general data structure, but we believe discussions based on information geometry may provide some suggestions.

6.4.2 From Classical to Quantum Many-body Approximation

Many-body approximation for tensors is based on the strategy of statistical mechanics. The probability of the appearance of a state, i.e., the value of the indices of a tensor, is defined by an energy function. The model in Equation (5.1) that includes the energy function in the exponential function is called canonical distribution [106, Chapter 5].

In quantum mechanics, physical quantities are described by Hermitian matrices whose eigenvalues are the possible measured values [20, Appendix A2]. In particular, the Hermitian matrix corresponding to energy is called a Hamiltonian. The many-body approximation may be extended into the world of quantum mechanics by replacing the energy function in the canonical distribution with a Hamiltonian. Similar extensions have already been demonstrated in quantum Boltzmann machines [5].

The quantity obtained by putting a Hamiltonian into an exponential function is not a distribution but a density matrix. We obtain a probability distribution by computing the trace of the product of the density matrix and the projection matrix corresponding to the physical quantity to be observed, e.g., the z-direction of spins. By mapping the obtained probability distribution to a non-negative tensor and parameterizing the Hamiltonian with n-body parameters, we can formulate a quantum many-body approximation for tensors.

Quantum Boltzmann machines are more expressive than classical Boltzmann machines because they can handle quantum spin states that classical Boltzmann machines cannot describe. We believe that a similar advantage can be seen in the quantum many-body approximation for tensors. We are fascinated by this idea of increasing the expressive power of tensor decompositions through quantum mechanics.

6.4.3 From Intra-tensor to Inter-tensor

Many-body approximation assumes the existence of dominant interactions between modes in a tensor. Using the tensor interaction diagram introduced in Chapter 5, we intuitively discuss approximations based on these interaction in a tensor. The natural question arises: can this diagram be applied to multiple tensors? For example, as shown in Figure 6.2, we can use the interactive representation to show how multiple tensors are related to each other through certain modes. In this way, we might be able to introduce interactions between tensors. Couple tensor analysis [121], which discovers relationships between tensors, often assumes shared bases between the tensors and performs a decomposition based on nonconvex optimization. The decomposition based on interactions between tensors possibly reveals tensors' relationships that cannot be discovered by traditional low-rank approximation.

6.4.4 From Undirected to Directed Interactions

We introduced an interaction between the modes of the tensor and visualized it in a factor graph. A factor graph is a network representation using an undirected graph [11, Chapter 8]. On the other hand, Bayesian networks, which represent distributions by directed graphs, are widely used in the field of graphical modeling [91, Chapter 10].

Bayesian networks represent the product of conditional probabilities. Interestingly, when we map non-negative tensors to joint probability distributions, the n-body expectation parameter naturally appears in the computation of the conditional probabilities.

$$p(i \mid j) = \frac{p(i,j)}{\sum_k p(i,k)} = \frac{p(i,j)}{\eta_{i1} - \eta_{i+1,1}}.$$

In the many-body approximation, we control interactions between modes with the natural parameter θ. However, we did not discuss the role of the expectation parameter η. Can we introduce directed interactions between modes by using a many-body η parameter? What do we mean by directed interactions between modes? We feel that many possibilities lie in this exploration.

6.5 Final Remarks

This dissertation proposed a novel framework that enables convex and rank-free approximation for various data structures by designing manifold learning with information geometry, the geometry of probability distributions, and energy-based models inspired by statistical mechanics. As we have described in this chapter, we expect many landscapes beyond the dissertation. It is just the beginning of our framework, and we believe that this dissertation opens a new world that will free us from the dimensionality reduction difficulties we have suffered from for a long time, e.g., tuning of low-rank structure, non-convexity of the objective function, initial value dependence, and ill-posed problems of low-rank tensor reconstruction.

Symbols

Tensors and Matrices

$\mathcal{P}, \mathcal{Q}, \mathcal{R}, \dots$	Tensors
$\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \dots$	Matrices
$\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \dots$	Vectors
$\overline{\mathcal{P}}$	The rank-reduced or body-reduced tensor
D	The order of tensor
I_k	The length of the k-th mode of a tensor
Ω_D	The index set of the D-th order tensor
$\mathcal{I}$	The hyper diagonal tensor
$\mathbf{P}^{(k)}$	The mode-k expansion of the tensor $\mathcal{P}$
$\mathcal{P}^{\leq m}$	The m-body approximation for the non-negative tensor $\mathcal{P}$
$\mathcal{P}^{\mathrm{cyc}}$	The cyclic two-body approximation for the non-negative tensor $\mathcal{P}$
$\mathcal{Q}_{\boldsymbol{c}}$	The set of slice-balanced tensors
$\mathcal{Q}_{\mathcal{C}}$	The set of fiber-balanced tensors
$\mathcal{P}_{(k)}$	The resulting tensor of m-projection of $\mathcal{P}$ onto $\mathcal{B}^{(k)}$
$\mathcal{P}_{a^{(k)}:b^{(k)}}$	The subtensor obtained by fixing the range of kth index to only from a to b of $\mathcal{P}$
$(r_1, \dots, r_D)$	The Tucker rank
$(R_1, \dots, R_D)$	The tensor ring rank
$\mathbf{R}^{k\leftrightarrow l}$	The permutation matrix, which switches the k-th row and the l-th row
$\mathbf{I}$	The identity matrix
$\mathbf{1}_{IJ}$	The $I \times J$ all-one matrix
$\mathbf{0}_{IJ}$	The $I \times J$ all-zero matrix
$\boldsymbol{\Phi}$	The weight matrix
$\otimes$	Kronecker product
$\circ$	The element-wise product
$S(\cdot)$	The total sum of the matrix or the vector
$\mathrm{Rank}(\cdot)$	The matrix rank

Log-linear Model on Posets and Information Geometry

$(\Omega, \leq)$	The poset (partially ordered set)
$\perp$	The least element in the poset
θ	The natural parameters
η	The expectation parameters
$\theta_{i_k}^{(k)}, \eta_{i_k}^{(k)}$	The one-body θ and η-parameter
G	Fisher information matrix
$\psi(\theta)$	Helmholtz free energy
$\mu(\cdot, \cdot)$	Möbius function
$\zeta(\cdot, \cdot)$	Zeta function
$D(\cdot, \cdot)$	Kullback–Leibler (KL) divergence
$D_{\mathbf{\Phi}}(\cdot, \cdot)$	Weighted KL divergence
$H^{(l_1, \dots, l_n)}$	The n-th order energy
$\mathcal{B}^{(k)}$	The bingo space on the mode-k
$\mathcal{B}_1$	The rank-1 space

Sets

$\varnothing$	The empty set
$\mathbb{N}$	The set of natural numbers
$\mathbb{R}$	The set of real numbers
$\mathbb{R}_{\geq 0}$	The set of non-negative real numbers
$\mathbb{R}_{> 0}$	The set of positive real numbers
$B \setminus A$	The set difference of B and A
$[n, m]$	$\{n, n+1, \dots, m-1, m\}$
$[m]$	$\{1, 2, \dots, m\}$

Others

O	Landau's symbol
Z	The partition function
$\|\cdot\|_{\mathrm{F}}$	Frobenius norm
$\|\cdot\|$	Euclidean norm of a vector
$\mathbb{E}[\,\cdot\,]$	The expected value
δ_{ij}	Kronecker delta, i.e., $\delta_{ij} = 1$ if $i = j$ and 0 otherwise

List of Definitions

List of Propositions

List of Theorems

List of Tasks

List of Tables

List of Figures

References

[1] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

[2] Emmanuel Agullo, Eric Darve, Luc Giraud, and Yuval Harness. Low-rank factorizations in data sparse hierarchical algorithms for preconditioning symmetric positive definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 39(4):1701–1725, 2018.

[3] Alfred V. Aho, Michael R Garey, and Jeffrey D. Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1(2):131–137, 1972.

[4] S. Amari. *Information Geometry and Its Applications*. Springer, 2016.

[5] Mohammad H Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum boltzmann machine. *Physical Review X*, 8(2):021050, 2018.

[6] J. R. Anderson and C. Peterson. A mean field theory learning algorithm for neural networks. *Complex Systems*, 1:995–1019, 1987.

[7] Titu Andreescu and Zuming Feng. *A Path to Combinatorics for Undergraduates: Counting Strategies*. Springer Science & Business Media, 2003.

[8] C. Bhattacharyya and S. S. Keerthi. Information geometry and Plefka's mean-field theory. *Journal of Physics A: Mathematical and General*, 33(7):1307, 2000.

[9] Michael Biggs, Ali Ghodsi, and Stephen Vavasis. Nonnegative matrix factorization via rank-one downdate. In *Proceedings of the 25th International Conference on Machine Learning*, pages 64–71, 2008.

[10] David Binkley. Source code analysis: A road map. *Future of Software Engineering (FOSE'07)*, pages 104–119, 2007.

[11] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[12] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[13] Jerome Brachat, Pierre Comon, Bernard Mourrain, and Elias Tsigaridas. Symmetric tensor decomposition. *Linear Algebra and its Applications*, 433(11-12):1851–1872, 2010.

[14] James R Bunch, Linda Kaufman, and Beresford N Parlett. Decomposition of a symmetric matrix. *Numerische Mathematik*, 27(1):95–109, 1976.

[15] P. E. Caines, M. Huang, and R. Malhamé. Large population stochastic dynamic games: Closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. *Communications in Information and Systems*, 6(3):221–252, 2006.

[16] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

[17] Jiahao Chen and Jarrett Revels. Robust benchmarking in noisy environments. *arXiv:1608.04295*, Aug 2016.

[18] Xing-Yu Chen, Jie Zhang, and Li-Rong Dai. Reference microphone selection and low-rank approximation based multichannel wiener filter with application to speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4963–4967. IEEE, 2022.

[19] Song Cheng, Lei Wang, Tao Xiang, and Pan Zhang. Tree tensor networks for generative modeling. *Physical Review B*, 99(15):155131, 2019.

[20] Mahn-Soo Choi. *A Quantum Computation Workbook*. Springer, 2022.

[21] Andrzej Cichocki, Sergio Cruces, and Shun-ichi Amari. Generalized alpha-beta divergences and their application to robust nonnegative matrix factorization. *Entropy*, 13(1):134–170, 2011.

[22] Andrzej Cichocki, Hyekyoung Lee, Yong-Deok Kim, and Seungjin Choi. Nonnegative matrix factorization with α-divergence. *Pattern Recognition Letters*, 29(9):1433–1440, 2008.

[23] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, Danilo P Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016.

[24] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE signal processing magazine*, 32(2):145–163, 2015.

[25] Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.

[26] Alex P da Silva, Pierre Comon, and André LF de Almeida. A finite algorithm to compute rank-1 tensor approximations. *IEEE Signal Processing Letters*, 23(7):959–963, 2016.

[27] Alex Pereira da Silva, Pierre Comon, and Andre Lima Ferrer de Almeida. Rank-1 tensor approximation methods and application to deflation. *arXiv preprint arXiv:1508.05273*, 2015.

[28] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[29] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.

[30] Vin De Silva and Lek-Heng Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.

[31] Weisheng Dong, Guangyu Li, Guangming Shi, Xin Li, and Yi Ma. Low-rank tensor approximation with laplacian scale mixture modeling for multiframe image denoising. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 442–449, 2015.

[32] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[33] Aybüke Erol and Borbála Hunyadi. Tensors for neuroimaging: A review on applications of tensors to unravel the mysteries of the brain. *Tensors for Data Processing*, pages 427–482, 2022.

[34] Glen Evenbly and Guifre Vidal. Tensor network renormalization. *Physical Review Letters*, 115(18):180405, 2015.

[35] Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830, 2009.

[36] Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Computation*, 23(9):2421–2456, 2011.

[37] Ronald Aylmer Fisher and Frank Yates. *Statistical Tables for Biological, Agricultural and Medical Research*. Hafner Publishing Company, 1953.

[38] Shmuel Friedland, Volker Mehrmann, Agnieszka Miedlar, and M Nkengla. Fast low rank approximations of matrices and tensors. *The Electronic Journal of Linear Algebra*, 22:1031–1048, 2011.

[39] K. Ghalamkari and M. Sugiyama. Towards geometric understanding of low-rank approximation. In *NeurIPS 2020 Workshop: Differential Geometry meets Deep Learning*, Virtual Event, December 2020.

[40] K. Ghalamkari and M. Sugiyama. Fast tucker rank reduction for non-negative tensors using mean-field approximation. In *Advances in Neural Information Processing Systems*, volume 34, pages 443–454, Virtual Event, December 2021.

[41] K. Ghalamkari and M. Sugiyama. Fast rank-1 NMF for missing data with KL divergence. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, pages 2927–2940, Virtual Event, March 2022.

[42] Kazu Ghalamkari and Mahito Sugiyama. Many-body approximation for tensors. *arXiv preprint arXiv:2209.15338*, 2022.

[43] Kazu Ghalamkari and Mahito Sugiyama. Non-negative low-rank approximations for multi-dimensional arrays on statistical manifold. *Information Geometry*, pages 1–36, 2023.

[44] Nicolas Gillis and François Glineur. Low-rank matrix approximation with weights or missing data is NP-hard. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1149–1165, 2011.

[45] Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[46] Edward F. Gonzalez. *Efficient Alternating Gradient-type Algorithms for the Approximate Non-negative Matrix Factorization Problem*. PhD thesis, Rice University, 1 2007.

[47] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 43–54, 1996.

[48] Lars Grasedyck, Daniel Kressner, and Christine Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013.

[49] Christian Grussler and Anders Rantzer. On optimal low-rank approximation of non-negative matrices. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 5278–5283. IEEE, 2015.

[50] Wolfgang Hackbusch. *Tensor spaces and numerical tensor calculus*, volume 42. Springer, 2012.

[51] Davood Hajinezhad, Tsung-Hui Chang, Xiangfeng Wang, Qingjiang Shi, and Mingyi Hong. Nonnegative matrix factorization using ADMM: Algorithm and convergence analysis. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4742–4746. IEEE, 2016.

[52] Richard A. Harshman. Foundations of the parafac procedure : Models and conditions for an "explanatory" multi-mode factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

[53] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):1–39, 2013.

[54] Frank L Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927.

[55] Frank L Hitchcock. Multiple invariants and generalized rank of a p-way matrix or tensor. *Journal of Mathematics and Physics*, 7(1-4):39–79, 1928.

[56] Ngoc-Diep Ho and Paul Van Dooren. Non-negative matrix factorization with fixed row and column sums. *Linear Algebra and its Applications*, 429(5–6):1020–1025, 2008.

[57] Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *Asian Conference on Computer Vision*, pages 19–34. Springer, 2016.

[58] Junhui Hou, Lap-Pui Chau, Nadia Magnenat-Thalmann, and Ying He. Sparse low-rank matrix approximation for data compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(5):1043–1054, 2015.

[59] Kejun Huang and Nicholas D Sidiropoulos. Kullback-leibler principal component for tensors is not np-hard. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pages 693–697. IEEE, 2017.

[60] Yuwang Ji, Qiang Wang, Xuan Li, and Jie Liu. A survey on tensor techniques and applications in machine learning. *IEEE Access*, 7:162950–162990, 2019.

[61] Xiaoran Jiang, Mikaël Le Pendu, Reuben A Farrugia, and Christine Guillemot. Light field compression with homography-based low-rank approximation. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):1132–1145, 2017.

[62] Jingu Kim, Yunlong He, and Haesun Park. Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. *Journal of Global Optimization*, 58(2):285–319, 2014.

[63] Yong-Deok Kim and Seungjin Choi. Nonnegative Tucker decomposition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[64] Yong-Deok Kim and Seungjin Choi. Weighted nonnegative matrix factorization. In *2009 IEEE international conference on acoustics, speech and signal processing*, pages 1541–1544. IEEE, 2009.

[65] Yong-Deok Kim, Andrzej Cichocki, and Seungjin Choi. Nonnegative Tucker decomposition with alpha-divergence. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1829–1832. IEEE, 2008.

[66] Stefan Klus and Patrick Gelß. Tensor-based algorithms for image classification. *Algorithms*, 12(11):240, 2019.

[67] Masahiro Kohjima, Tatsushi Matsubayashi, and Hiroshi Sawada. Non-negative multiple matrix factorization with Euclidean and Kullback-Leibler mixed divergences. In *2016 23rd International Conference on Pattern Recognition*, pages 2515–2520. IEEE, 2016.

[68] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[69] Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, and Maja Pantic. Tensorly: Tensor learning in python. *Journal of Machine Learning Research*, 20(26):1–6, 2019.

[70] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[71] Lieven De Lathauwer. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[72] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fujie Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[73] Daniel Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 556–562. MIT Press, 2001.

[74] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[75] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562, 2001.

[76] Michael Levin and Cody P Nave. Tensor renormalization group approach to two-dimensional classical lattice models. *Physical Review Letters*, 99(12):120601, 2007.

[77] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 31–42, 1996.

[78] Qing Liao and Qian Zhang. Efficient rank-one residue approximation method for graph regularized non-negative matrix factorization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 242–255. Springer, 2013.

[79] P.-L. Lions and J.-M. Lasry. Large investor trading impacts on volatility. *Annales de l'Institut Henri Poincare (C) Non Linear Analysis*, 24(2):311–323, 2007.

[80] Xinyue Liu, Chara Aggarwal, Yu-Feng Li, Xiaugnan Kong, Xinyuan Sun, and Saket Sathe. Kernelized matrix factorization for collaborative filtering. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 378–386. SIAM, 2016.

[81] Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):925–938, 2019.

[82] Yuan Luo, Fei Wang, and Peter Szolovits. Tensor factorization toward precision medicine. *Briefings in bioinformatics*, 18(3):511–514, 2017.

[83] Osman Asif Malik and Stephen Becker. A sampling-based method for tensor ring decomposition. In *International Conference on Machine Learning*, pages 7400–7411. PMLR, 2021.

[84] Ivan Markovsky. *Low rank approximation: algorithms, implementation, applications*, volume 906. Springer, 2012.

[85] Takeru Matsuda and Tasuku Soma. Information geometry of operator scaling. *Linear Algebra and its Applications*, 649:240–267, 2022.

[86] Leon Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 1960.

[87] Yusuke Monno, Daisuke Kiku, Masayuki Tanaka, and Masatoshi Okutomi. Adaptive residual interpolation for color and multispectral image demosaicking. *Sensors*, 17(12):2787, 2017.

[88] Yusukex Monno, Sunao Kikuchi, Masayuki Tanaka, and Masatoshi Okutomi. A practical one-shot multispectral imaging system using a single image sensor. *IEEE Transactions on Image Processing*, 24(10):3048–3059, 2015.

[89] Valentin Murg, Frank Verstraete, Örs Legeza, and Reinhard M Noack. Simulating strongly correlated quantum systems with tree tensor networks. *Physical Review B*, 82(20):205105, 2010.

[90] Valentin Murg, Frank Verstraete, Reinhold Schneider, Peter R Nagy, and O Legeza. Tree tensor network state with variable tensor order: An efficient multireference method for strongly correlated systems. *Journal of Chemical Theory and Computation*, 11(3):1027–1036, 2015.

[91] Kevin P Murphy. *Machine learning: A probabilistic perspective*. MIT press, 2012.

[92] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[93] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[94] Ricardo Otazo, Emmanuel Candes, and Daniel K Sodickson. Low-rank plus sparse matrix decomposition for accelerated dynamic mri with separation of background and dynamic components. *Magnetic resonance in medicine*, 73(3):1125–1136, 2015.

[95] Alexey Ozerov, Ngoc Duong, and Louis Chevallier. Weighted nonnegative tensor factorization: on monotonicity of multiplicative update rules and application to user-guided audio source separation. *Technical Report*, 2013.

[96] Yannis Panagakis, Jean Kossaifi, Grigorios G Chrysos, James Oldfield, Mihalis A Nicolaou, Anima Anandkumar, and Stefanos Zafeiriou. Tensor methods in computer vision and deep learning. *Proceedings of the IEEE*, 109(5):863–890, 2021.

[97] Ankit Parekh and Ivan W Selesnick. Improved sparse low-rank matrix estimation. *Signal Processing*, 139:62–69, 2017.

[98] Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8, 2007.

[99] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[100] Lana Periša and Alex Arslan. lanaperisa/tensortoolbox.jl v1.0.2. Nov 2019.

[101] C. Peterson. A mean field theory learning algorithm for neural networks. *Complex Systems*, pages 995–1019, 1987.

[102] Menaka Rajapakse, Jeffrey Tan, and Jagath Rajapakse. Color channel encoding with NMF for face recognition. In *2004 International Conference on Image Processing, 2004. ICIP'04.*, volume 3, pages 2007–2010. IEEE, 2004.

[103] G.-C. Rota. On the foundations of combinatorial theory I: Theory of Möbius functions. *Z. Wahrseheinlichkeitstheorie*, 2:340–368, 1964.

[104] Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *2013 IEEE International Conference on Acoustics, speech and signal processing*, pages 6655–6659. IEEE, 2013.

[105] Yaser Esmaeili Salehani and Saeed Gazor. Smooth and sparse regularization for nmf hyperspectral unmixing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(8):3677–3692, 2017.

[106] Silvio R.A. Salinas. *Introduction to Statistical Physics*. Springer, 2001.

[107] Ferdinando S Samaria and Andy C Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pages 138–142. IEEE, 1994.

[108] Amnon Shashua and Tamir Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 792–799, 2005.

[109] Y-Y Shi, L-M Duan, and Guifre Vidal. Classical simulation of quantum many-body systems with a tree tensor network. *Physical Review A*, 74(2):022320, 2006.

[110] Nicholas D Sidiropoulos and Anastasios Kyrillidis. Multi-way compressed sensing for sparse low-rank tensors. *IEEE Signal Processing Letters*, 19(11):757–760, 2012.

[111] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35(2):876–879, 1964.

[112] David Skillicorn. *Understanding complex datasets: data mining with matrix decompositions*. Chapman and Hall/CRC, 2007.

[113] Dongjin Song, David A Meyer, and Martin Renqiang Min. Fast nonnegative matrix factorization with rank-one ADMM. In *NIPS 2014 Workshop on Optimization for Machine Learning (OPT2014)*, 2014.

[114] Guang-Jing Song and Michael K Ng. Nonnegative low rank matrix approximation for nonnegative matrices. *Applied Mathematics Letters*, 105:106300, 2020.

[115] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning*, pages 720–727, 2003.

[116] M. Sugiyama, H. Nakahara, and K. Tsuda. Information decomposition on structured space. In *2016 IEEE International Symposium on Information Theory*, pages 575–579, 2016.

[117] M. Sugiyama, H. Nakahara, and K. Tsuda. Tensor balancing on statistical manifold. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3270–3279, 2017.

[118] Mahito Sugiyama, Hiroyuki Nakahara, and Koji Tsuda. Legendre decomposition for tensors. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124017, 2019.

[119] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, pages 43–50, 2008.

[120] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Investigation of various matrix factorization methods for large recommender systems. In *2008 IEEE International Conference on Data Mining Workshops*, pages 553–562. IEEE, 2008.

[121] Koh Takeuchi, Katsuhiko Ishiguro, Akisato Kimura, and Hiroshi Sawada. Non-negative multiple matrix factorization. In *Twenty-Third International Joint Conference on Artificial Intelligence*, pages 1713–1720, 2013.

[122] Koh Takeuchi, Ryota Tomioka, Katsuhiko Ishiguro, Akisato Kimura, and Hiroshi Sawada. Non-negative multiple tensor factorization. In *2013 IEEE 13th International Conference on Data Mining*, pages 1199–1204. IEEE, 2013.

[123] Toshiyuki Tanaka. A theory of mean field approximation. In *Advances in Neural Information Processing Systems*, pages 351–360, 1999.

[124] Bruce Thompson. *Canonical correlation analysis: Uses and interpretation*. Sage, 1984.

[125] Roberto Todeschini and Viviana Consonni. *Handbook of molecular descriptors*. John Wiley & Sons, 2008.

[126] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[127] Laurens Van Der Maaten, Eric Postma, Jaap Van den Herik, et al. Dimensionality reduction: a comparative review. *J Mach Learn Res*, 10(66-71):13, 2009.

[128] Stephen A Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2010.

[129] P. Weiss. L'hypothèse du champ moléculaire et la propriété ferromagnétique. *Journal de Physique Théorique et Appliquée*, 6(1):661–690, 1907.

[130] Max Welling and Markus Weber. Positive tensor factorization. *Pattern Recognition Letters*, 22(12):1255–1261, 2001.

[131] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1–3):37–52, 1987.

[132] Tong Wu, Yunlong Wang, Yue Wang, Emily Zhao, and Yilian Yuan. Leveraging graph-based hierarchical medical entity embedding for healthcare applications. *Scientific Reports*, 11(1):1–13, 2021.

[133] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–273, 2003.

[134] Yuan You, Hongmin Cai, and Jiazhou Chen. Low rank representation and its application in bioinformatics. *Current Bioinformatics*, 13(5):508–517, 2018.

[135] YuYuan Yu, Kan Xie, Jinshi Yu, Qi Jiang, and Shengli Xie. Fast nonnegative tensor ring decomposition based on the modulus method and low-rank approximation. *Science China Technological Sciences*, 64(9):1843–1853, 2021.

[136] Yuyuan Yu, Guoxu Zhou, Ning Zheng, Yuning Qiu, Shengli Xie, and Qibin Zhao. Graph-regularized non-negative tensor-ring decomposition for multiway representation learning. *IEEE Transactions on Cybernetics*, 2022.

[137] Guoying Zhang, Min He, Hao Wu, Guanghui Cai, and Jianhong Ge. Non-negative multiple matrix factorization with social similarity for recommender systems. In *Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*, pages 280–286, 2016.

[138] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 549–553. SIAM, 2006.

[139] Tong Zhang and Gene H Golub. Rank-one approximation to high order tensors. *SIAM Journal on Matrix Analysis and Applications*, 23(2):534–550, 2001.

[140] Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.

[141] Guoxu Zhou, Andrzej Cichocki, and Shengli Xie. Fast nonnegative matrix/tensor factorization based on low-rank approximation. *IEEE Transactions on Signal Processing*, 60(6):2928–2940, 2012.

[142] Xiaowei Zhou, Can Yang, Hongyu Zhao, and Weichuan Yu. Low-rank modeling and its applications in image analysis. *ACM Computing Surveys (CSUR)*, 47(2):1–33, 2014.

Publications by the Author

Journal Papers

[J1] K. Ghalamkari and M. Sugiyama. Non-negative low-rank approximations for multi-dimensional arrays on statistical manifold. Information Geometry, 2023.(accepted)

Peer-reviewed Conference Papers

[P1] K. Ghalamkari and M. Sugiyama. Towards geometric understanding of low-rank approximation. In NeurIPS 2020 Workshop: Differential Geometry meets Deep Learning, Virtual Event, December 2020.

[P2] K. Ghalamkari and M. Sugiyama. Fast tucker rank reduction for non-negative tensors using mean-field approximation. In Advances in Neural Information Processing Systems, volume 34, pages 443–454, Virtual Event, December 2021.

[P3] K. Ghalamkari and M. Sugiyama. Fast rank-1 NMF for missing data with KL divergence. In Proceedings of the 25th International Conference on Artificial Intelligence and Statistics, pages 2927–2940, Virtual Event, March 2022.

Review papers

[R1] ガラムカリ 和, 杉山 麿人, Leslie O'Bray, Bastian Rieck, Karsten Borgwardt. グラフカーネルの進展, 人工知能 36 (4), 421-429

Conference Proceedings

[C1] ガラムカリ 和, 杉山 麿人. 欠損を含む非負行列の高速なランク 1 分解, 人工知能学会研究会資料 第120回人工知能基本問題研究会, 1-5, 2022.3

[C2] ガラムカリ 和, 杉山 麿人. 平均場近似に基づく正テンソルの最良ランク 1 近似, 人工知能学会全国大会論文集, 1H3GS1b02-1H3GS1b02, 2021.3

Index

Colophon

This thesis was typeset with LaTeX 2$_{\varepsilon}$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc. Soham Chatteerjee developed some theorem environments.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.