Tensor Completion Algorithms in Big Data Analytics

QINGQUAN SONG, HANCHENG GE, JAMES CAVERLEE, and XIA HU,
Texas A&M University

Tensor completion is a problem of filling the missing or unobserved entries of partially observed tensors. Due to the multidimensional character of tensors in describing complex datasets, tensor completion algorithms and their applications have received wide attention and achievement in areas like data mining, computer vision, signal processing, and neuroscience. In this survey, we provide a modern overview of recent advances in tensor completion algorithms from the perspective of big data analytics characterized by diverse variety, large volume, and high velocity. We characterize these advances from the following four perspectives: general tensor completion algorithms, tensor completion with auxiliary information (variety), scalable tensor completion algorithms (volume), and dynamic tensor completion algorithms (velocity). Further, we identify several tensor completion applications on real-world data-driven problems and present some common experimental frameworks popularized in the literature along with several available software repositories. Our goal is to summarize these popular methods and introduce them to researchers and practitioners for promoting future research and applications. We conclude with a discussion of key challenges and promising research directions in this community for future exploration.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Mathematics of computing** → *Dimensionality reduction*; • **Computing methodologies** → *Dimensionality reduction and manifold learning*; *Factorization methods*; *Regularization*;

Additional Key Words and Phrases: Tensor, tensor completion, tensor decomposition, tensor factorization, multilinear data analysis, dynamic data analysis, big data analytics

ACM Reference format:
Qingquan Song, Hancheng Ge, James Caverlee, and Xia Hu. 2019. Tensor Completion Algorithms in Big Data Analytics. *ACM Trans. Knowl. Discov. Data* 13, 1, Article 6 (January 2019), 48 pages.
https://doi.org/10.1145/3278607

1 INTRODUCTION

Tensor analysis has garnered increasing attention in recent years. In one direction, tensor decomposition – which aims to distill tensors into (interpretable) soft representations – has been widely studied from both a theoretical and application-specific perspective [106, 135, 154, 174]. In a related direction and the focus of this survey, *tensor completion* aims to impute missing or unobserved entries of a partially observed tensor. Tensor completion is one of the most actively studied problems in tensor related research and yet diffusely presented across many different research domains. On one hand, multidimensional datasets are often raw and incomplete owing to various unpredictable or unavoidable reasons such as mal-operations, limited permissions, and data missing at

This work is, in part, supported by DARPA (#W911NF-16-1-0565) and NSF (#IIS-1750074, #IIS-1718840).
Authors' addresses: Q. Song, H. Ge, J. Caverlee, and X. Hu, Department of Computer Science and Engineering, Texas A&M University, 400 Bizzell St, College Station, TX 77843; emails: {song_3134, hge, caverlee, xiahu}@tamu.edu.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2019 Association for Computing Machinery.
1556-4681/2019/01-ART6 $15.00
https://doi.org/10.1145/3278607

random [61, 129, 132]. On the other hand, in practice, due to the multiway property of modern datasets, tensor completion natural arises in data-driven applications such as image completion and video compression.

In the past few decades, the matrix completion problem – a special case of tensor completion – has been well-studied. Mature algorithms [22], theoretical foundations [26], and various applications [28] pave the way for solving the completion problem in high-order tensors. Intuitively, the tensor completion problem could be solved with matrix completion algorithms by downgrading the problem into a matrix level, typically by either slicing a tensor into multiple small matrices or unfolding it into one big matrix. However, several problems distinguish tensor completion from being treated as a straightforward extension of the matrix completion problem. First, it has been shown that matrix completion algorithms may break the multi-way structure of a tensor and lose the redundancy among modes to improve the imputation accuracy [178]. While many tensor-based algorithms directly build upon matrix completion algorithms [143, 215], their key focus is out of the context of matrix level, i.e., trying to develop delicate ways of matricization to keep the multi-way properties of a tensor object while using matrix-based completion algorithms. Second, the high-order characteristics introduce even higher space and computational complexity, which may prevent the usage of traditional matrix completion algorithms.

In some early works [5, 20, 180, 199], tensor completion is often considered as a byproduct when dealing with missing data in the tensor decomposition problem. However, due to the distinct objective and challenges, it gradually becomes an independent topic jumping out of the context of tensor decomposition. Concretely, tensor completion is a specific task, whose ultimate goal is to fill the missing or unobserved entries while tensor decomposition is an intermediate step to distill the tensors into (interpretable) soft representations to benefit subsequent tasks. It is feasible to employ tensor decomposition methods to solve the completion problem. However, the solutions of tensor completion are not restricted to decomposition-based approaches [130]. Moreover, the key issue of missing data in the completion problem may prevent some traditional decomposition methods being used such as the vanilla alternating least square (ALS) methods for CANDECOMP/PARAFAC decomposition [30, 74]. The optimization scheme especially for the projection-based methods should be carefully modified towards feasible adoption.

Though many efforts have focused on the tensor completion problem in recent years, to the best of our knowledge, we still lack a comprehensive survey to capture the many advances across domains. Existing surveys rest on the matrix level [90, 109, 118, 188] or focus on related topics, e.g., tensor decomposition [4, 38, 53, 69, 76, 102, 106, 112, 113, 135, 154, 174, 180]. Moreover, with the rapid growth of real-world big data applications demonstrating variety, velocity, and volume, there has been a commensurate rise in tensor completion algorithms that are capable of leveraging heterogeneous data sources, handling real large-scale datasets, as well as tracking dynamical changes over time. The incremental interest in big data analytics [91, 107, 170] motivates us to provide a view from big data perspective in the present survey that differs from most of the existing reviews. Hence, the goal of this survey is to give an overview of general and advanced high-order tensor completion methods and their applications in different fields. We aim at summarizing the state-of-the-art tensor completion methods for promoting the research process in this area as well as providing a handy reference handbook for both researchers and practitioners.

Tightly coupling with the "3V" challenges [17] [1] in big data analytics including *variety*, *volume*, and *velocity*, we organize the structure of this article as shown in Figure 1. Notations and primary multi-linear algebra operations are first introduced to formulate the tensor completion problem and its variations. A concise summary of general tensor completion algorithms is provided along

[1] As described recently in [220], Big data has "5V" challenges including "variety," "volume," "velocity," "veracity," and "value." Veracity refers to the quality or uncertainty (trustworthiness) of the collected data and value represents the worth

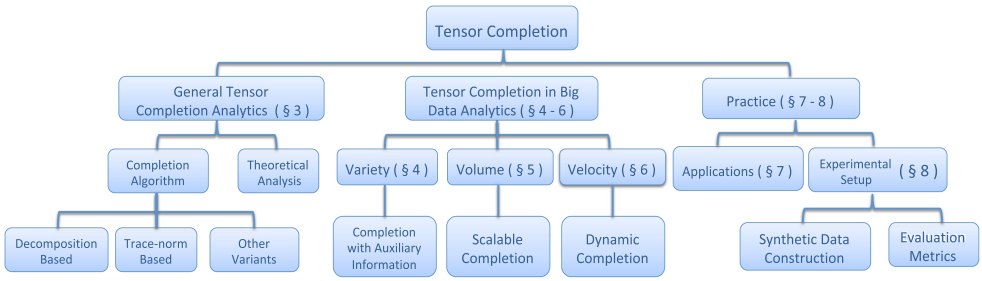

Fig. 1. Outline of this survey.

Table 1. Main Symbols and Operations

Notations	Definitions
$\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$	Nth-order tensor
$\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times (\Pi_{i \neq n}^{N} I_i)}$	Mode-n unfolding matrix of tensor $\mathcal{X}$
$\mathcal{X}_{i_1,\ldots,i_N}$ or $\mathcal{X}(i_1,\ldots,i_N)$	An entry of tensor $\mathcal{X}$ indexed by $[i_1,\ldots,i_N]$
$< \cdot, \cdot >$	Inner product
$\circ$	Outer product
$[\![\cdot]\!]$	Kruskal operator
$\otimes$	Kronecker product
$\odot$	Khatri–Rao product
$*$	Hadamard product
$\|\| \cdot \|\|_F$	Frobenius norm
$\|\| \cdot \|\|_*$	Trace norm

with several statistical assumptions and theoretical analysis. Subsequently, coping with the "3V" challenges in big data analytics, three categories of advanced completion methods are introduced including tensor completion with auxiliary information (*variety*), scalable tensor completion algorithms (*volume*), and dynamic completion methods (*velocity*). Real-world applications in different areas are outlined in Section 7, and several commonly used experimental simulations, metrics as well as software repositories are presented in Section 8. Finally, we describe open challenges and promising directions.

2 TENSOR COMPLETION PROBLEM

In this section, we provide the formal definition of the tensor completion problem. We begin by introducing several tensor operations, basic definitions, and notations following [106]. Table 1 summarizes the notations used throughout this article.

2.1 Preliminaries

Definition 2.1.1 (Tensor). A tensor can be defined in different ways at various levels of abstraction [105, 119]. We follow the most general way and define it as a *multi-dimensional array*. The dimensionality of it is described as its *order*. An Nth-order tensor is an N-way array, also known as N-dimensional or N-mode tensor, denoted by $\mathcal{X}$. We use the term *order* to refer to the dimensionality of a tensor (e.g., Nth-order tensor), and the term *mode* to describe operations on a specific dimension (e.g., mode-n product).

of the data being extracted. We focus on the first three for the convenience of dividing the algorithms from the model perspective.

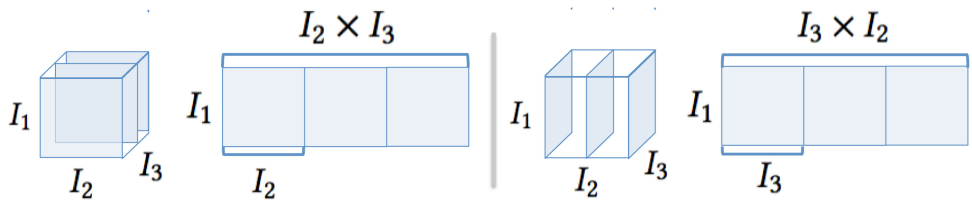

Fig. 2. Forward (left) and backward (right) mode-one matricization ($\mathbf{X}_{(1)}$) of the tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$.

Definition 2.1.2 (Tensor Matricization). Tensor matricization is to unfold a tensor into a matrix format with a predefined ordering of its modes. The most commonly used tensor matricization is mode-n matricization (a.k.a., mode-n unfolding), which is to unfold a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ along its nth mode into a matrix denoted as $\mathbf{X}_{(n)}$ of size $I_n \times (\prod_{k \neq n} I_k)$. The ordering of the other modes except mode n can be arranged randomly to construct the column of $\mathbf{X}_{(n)}$. Two commonly used arrangements are forward [101] (i.e., $n+1, \ldots, N, 1, \ldots, n-1$) and backward [42] (i.e., $n-1, \ldots, 1, N, \ldots, n+1$). We depict such a forward and backward mode-one unfolding of a third-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ in Figure 2. In general, the specific permutation is not important as long as it is consistent across related calculations [106]. More comprehensive comparison and visualizations can be found in [9].

Definition 2.1.3 (Product Operations).

- Outer Product: product of vectors denoted by $\circ$. For N column vectors $\mathbf{a}^{(1)} \in \mathbb{R}^{I_1}, \ldots, \mathbf{a}^{(N)} \in \mathbb{R}^{I_N}$. The outer product among them is defined as

$$\mathcal{T} = \mathbf{a}^{(1)} \circ \ldots \circ \mathbf{a}^{(N)}, \ \mathcal{T}_{i_1, \ldots, i_N} = \mathbf{a}^{(1)}_{i_1} \cdots \mathbf{a}^{(N)}_{i_N}, \tag{1}$$

  where $\mathcal{T} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$. It is the same as matrix multiplication when $N = 2$.
- Kronecker Product: product of two matrices. Given two matrices $\mathbf{A}$ and $\mathbf{B}$ of sizes $m \times n$ and $p \times q$, respectively, their kronecker product is defined as

$$\mathbf{A}_{m \times n} \otimes \mathbf{B}_{p \times q} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}_{mn \times pq}, \text{ where } \mathbf{A}_{m \times n} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}_{m \times n}. \tag{2}$$

- Khatri–Rao Product: block-wise Kronecker product. Here, we treat it as column-wise Kronecker product of two matrices with the same number of columns, which is defined as

$$\mathbf{A}_{m \times N} \odot \mathbf{B}_{p \times N} = [\mathbf{a}_1 \otimes \mathbf{b}_1, \ldots, \mathbf{a}_N \otimes \mathbf{b}_N]_{mp \times N}, \tag{3}$$

  where $\mathbf{A}_{m \times N} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]_{m \times N}$ and $\mathbf{B}_{p \times N} = [\mathbf{b}_1, \ldots, \mathbf{b}_N]_{p \times N}$.
- Hadamard Product: element-wise product of two tensors with the same size, denoted as $\mathcal{A} * \mathcal{B}$. Its inverse division operation is denoted as $\mathcal{A} \oslash \mathcal{B}$.
- Mode-n Product: multiplication of a given tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ on its nth-mode with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$, which is denoted as $\mathcal{Z} = \mathcal{X} \times_n \mathbf{U}$, where $\mathcal{Z} \in \mathbb{R}^{I_1 \ldots I_{n-1} \times J \times I_{n+1} \ldots I_N}$. The elementwise result is described as

$$\mathcal{Z}_{i_1, \ldots, i_{n-1}, j, i_{n+1}, \ldots, i_N} = (\mathcal{X} \times_n \mathbf{U})_{i_1, \ldots, i_{n-1}, j, i_{n+1}, \ldots, i_N} = \sum_{k=1}^{I_n} \mathcal{X}_{i_1, \ldots, i_{n-1}, k, i_{n+1}, \ldots, i_N} \mathbf{U}_{j,k}. \tag{4}$$

- General Tensor Product (Multiplication): the general tensor (outer) product of tensors. Given two tensors $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ and $\mathcal{Y} \in \mathbb{R}^{J_1 \times \ldots \times J_N}$, there tensor product is defined as

$\mathcal{Z} = \mathcal{X} \circ \mathcal{Y}$, where $\mathcal{Z} \in \mathbb{R}^{I_1 \times \ldots \times I_N \times J_1 \times \ldots \times J_N}$ and the elementwise result is described as

$$\mathcal{Z}_{i_1,\ldots,i_N,j_1,\ldots,j_N} = \mathcal{X}_{i_1,\ldots,i_N} \mathcal{Y}_{j_1,\ldots,j_N}. \tag{5}$$

See also Bader and Kolda [9] for a detailed treatment of tensor multiplication.

Definition 2.1.4 (Rank-1 Tensor). Also called simple tensor [77] or decomposable tensor [72]. It is an Nth-order tensor $\mathcal{X}$ ($N \in \mathbb{Z}_+$) that could be written as the outer product of N vectors, i.e., $\mathcal{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \ldots \circ \mathbf{a}^{(N)}$.

Definition 2.1.5 (Tensor (CP) Rank). The tensor rank (i.e., tensor CP rank) of a tensor $\mathcal{X}$ is defined as the minimum number of summations of rank-one tensors that generate $\mathcal{X}$ [79, 115], i.e.,

$$\text{rank}_{\text{CP}}(\mathcal{X}) \triangleq \min \left\{ R \in \mathbb{Z}_+ : \exists \{\mathbf{a}_r^{(n)}\}, s.t. \, \mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \ldots \circ \mathbf{a}_r^{(N)} \right\}. \tag{6}$$

As calculating the tensor rank is an NP-hard problem [75], the tensor n-rank was introduced by Kruskal [116] as a special case of multiplex rank introduced by Hitchcock [78].

Definition 2.1.6 (Tensor n-rank). The tensor n-rank is defined as the rank (usually column rank) of $\mathbf{X}_{(n)}$, i.e., $\text{rank}_n(\mathcal{X}) = \text{rank}(\mathbf{X}_{(n)})$. Another similar definition called Tucker rank or multi-linear rank [96], which is introduced earlier by Tucker [203] is defined as $\text{rank}_{tc}(\mathcal{X}) = (\text{rank}(\mathbf{X}_{(1)}), \ldots, \text{rank}(\mathbf{X}_{(N)}))$.

Definition 2.1.7 (Tensor Inner Product). The inner product of two tensors $\mathcal{X}$ and $\mathcal{Y}$ of same size is defined as $< \mathcal{X}, \mathcal{Y} >$. Unless otherwise specified, we treat it as dot product defined as follows:

$$< \mathcal{X}, \mathcal{Y} > = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{X}_{i_1,i_2,\ldots,i_N} \mathcal{Y}_{i_1,i_2,\ldots,i_N}. \tag{7}$$

Definition 2.1.8 (Tensor Frobenius Norm). Generalized from matrix Frobenius norm, the F-norm of a tensor $\mathcal{X}$ is defined as

$$\|\mathcal{X}\|_{\text{F}} = \sqrt{< \mathcal{X}, \mathcal{X} >} = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{X}_{i_1,i_2,\ldots,i_N}^2}. \tag{8}$$

2.2 Tensor Completion Problem

Tensor completion is defined as the problem of filling the missing elements of partially observed tensors. As its particular matrix case [26], to avoid being an underdetermined and intractable problem, *low rank* is often a necessary hypothesis to restrict the degree of freedoms of the missing entries [59, 110, 130, 178]. Since a tensor has different types of rank definitions, to give a relatively general mathematical formulation of the low-rank tensor completion (LRTC) problem, we first summarize several most popular definitions [59, 130] into a unified optimization problem and then specify the variants derived from it by answering several questions.

Definition 2.2.1 (Low-Rank Tensor Completion Problem). Given a low-rank (either CP rank or other ranks) tensor $\mathcal{T}$ with missing entries, the goal of completing it can be formulated as the following optimization problem:

$$\begin{aligned} \underset{\mathcal{X}}{\text{minimize}} \quad & \text{rank}_*(\mathcal{X}) \\ \text{subject to} \quad & \mathcal{X}_\Omega = \mathcal{T}_\Omega, \end{aligned} \tag{9}$$

where rank_* denotes a specific type of tensor rank based on the rank assumption of the given tensor $\mathcal{T}$, $\mathcal{X}$ represents the completed low-rank tensor of $\mathcal{T}$, and Ω is an index set denoting the indices

of observations. The intuitive explanation of the above equation is that: we expect to find a tensor $\mathcal{X}$ with the minimum rank, which subjects to the equality constraints given by the observations (a.k.a., measurements). This equation is a straightforward generalization of the well-understood matrix completion problem [26] and could be treated as the starting point of almost every existing variant definitions of LRTC problem. To delve into these variants, we start by asking three questions with respect to the above equation and try to explore the answers from these variants. (1) What type of the rank should we use? (2) Are there any other constraints defined based on the observations that we could presume? (3) In what condition, could we expect to achieve a unique and exact completion? We mainly focus on the first two questions here to extend our discussion and leave the last question into Section 3.4 on account of its tight correlation with various statistical assumptions.

2.2.1 Rank Variants. We first discuss the first question (i.e., What type of the rank should we use?) and describe some variants of Equation (9) based on different rank selections. In the preliminary part, we have defined the following two different types of ranks: the tensor (CP) rank and the tensor n-rank. As they have covered most of existing popular articles, we now focus on introducing the main variant optimization problem utilizing these two types of ranks.

(1) Tensor (CP) Rank and Its Corresponding Variants of the LRTC Problem.

As the existing works have demonstrated that calculating the tensor (CP) rank is an NP-hard problem [75, 106], directly considering the minimization problem defined in Equation (9) with CP rank is unpractical. To avoid this problem, some researchers [8, 14, 85, 111] assume the CP rank of the target tensor is fixed to achieve a relative milder problem. By fixing the rank of $\mathcal{T}$, they are able to substitute the original objective function with some polynomial time computable norms of the tensor $\mathcal{X}$ [14] or treat the CP rank as the constraint and inversely minimize the difference between the true observations and their predictions as follows [85]:

$$\begin{aligned} \underset{\mathcal{X}}{\text{minimize}} \quad & \mathcal{D}(\mathcal{T}_\Omega, \mathcal{X}_\Omega) \\ \text{subject to} \quad & \text{rank}_{\text{CP}}(\mathcal{X}) = r, \end{aligned} \tag{10}$$

where $\mathcal{D}$ is the error measure between $\mathcal{X}_\Omega$ and $\mathcal{T}_\Omega$, which is often defined as the Frobenius norm of their difference under the assumption of Gaussian noise, i.e., $\mathcal{D}(\mathcal{T}_\Omega, \mathcal{X}_\Omega) = \|\mathcal{X}_\Omega - \mathcal{T}_\Omega\|_{\text{F}}$. Though fixing CP rank helps solve the problem, this assumption is always not an ideal choice since estimating the tensor rank is often very hard [116], which confines the applicability of corresponding completion algorithms. Thus, some researchers only assume the predefined CP rank is the rank upper bound [132] or propose the algorithms, which could dynamically conduct the rank search during the completion process [225]. Recent research has focused on minimizing the tensor CP-rank directly via controlling it with tensor nuclear norm [218] and analyzing it theoretically [231], which provide more practical and general solutions for the LRTC problem under CP-rank constraints.

(2) Tensor n-Rank and Its Corresponding Variants of the LRTC Problem.

Comparing with other ranks, Tensor n-rank (or Tucker rank) is perhaps the most widely adopted rank assumption in existing tensor completion literature [59, 110, 130, 176]. As it is defined by the matricizations of the tensor, many matrix completion techniques such as trace-norm based methods could be generalized into the high-order level [59, 130, 176]. Most of the existing work targets on solving the following tensor n-rank minimization problem described in [59]:

$$\begin{aligned} \underset{\mathcal{X}}{\text{minimize}} \quad & f(\text{rank}_{\text{tc}}(\mathcal{X})) \\ \text{subject to} \quad & \mathcal{X}_\Omega = \mathcal{T}_\Omega, \end{aligned} \tag{11}$$

where f is a predefined function for the tensor n-rank: $\text{rank}_{tc}(\mathcal{X}) = (\text{rank}(\mathbf{X}_{(1)}), \ldots, \text{rank}(\mathbf{X}_{(N)}))$. For example, the most commonly used function is the rank summation function [59, 169], i.e., $f(\text{rank}_{\text{tc}}(\mathcal{X})) = \sum_{i=1}^{N} \text{rank}(\mathbf{X}_{(i)})$. Standing upon the above optimization problem, there are also several variant ways of using tensor n-rank to formulate the LRTC problem. For example, Kressner et al. [110] assume the tensor n-rank to be fixed, i.e., $\text{rank}_{\text{tc}}(\mathcal{X}) = (r_1, \ldots, r_N)$ and then use the similar way defined in Equation (10) to formulate the completion problem. This definition allows them to explicitly leverage the tensor decomposition power to achieve the completion purpose. A correlated variant is to assume the tensor n-rank to be constrained [143], i.e., $\text{rank}_{\text{tc}}(\mathcal{X}) \preceq (r_1, \ldots, r_N)$, where $\preceq$ is an element-wise notation. Though tensor n-rank has been widely employed, recent works [159] also point out a crucial drawback that it only takes into consideration the ranks of matrices that are constructed based on the unbalanced matricization scheme, i.e., one mode versus the rest.

The drawbacks of utilizing tensor CP rank and tensor n-rank encourage researchers to propose other types of ranks such as tensor-train (TT) rank [68, 151, 159] and tensor tubal rank [233]. However, as the rank of a high-order tensor is still not a well-understood area, which is contrary to the matrix case, there is still no absolute conclusion that applying one rank is always better than another. Readers who are interested in the variant mathematical definition of the LRTC problem using other ranks could explore the corresponding papers for a more detailed introduction.

2.2.2 Constraint Variants. Another question that could introduce different variants for the general optimization problem (9) is what kind of constraints defined based on the observations that we could presume? In Equation (9), the constraints are intuitively defined as $\mathcal{X}_\Omega = \mathcal{T}_\Omega$ based on the observations. This simple constraint usually implies the following two statistical assumptions: (1) The situation we considered is the noise-free case [130]. (2) We assume the observations are uniformly random sampled based on Bernoulli distribution [82]. Although these two assumptions are widely used in the LRTC literature, there are also several popular variants. First, as data in real-world applications are often corrupted with various of noises, different noise assumptions, especially Gaussian noises, are commonly used by researchers [14, 59, 67], leading to the following noise-contained formulation:

$$\begin{aligned} \underset{\mathcal{X}}{\text{minimize}} \quad & \text{rank}_*(\mathcal{X}) \\ \text{subject to} \quad & \mathcal{X}_\Omega = \mathcal{T}_\Omega + \mathcal{E}_\Omega, \end{aligned} \tag{12}$$

where $\mathcal{E}$ denotes the noise term. It could be defined as either Gaussian [223] or non-Gaussian noises (e.g., Laplace noise [224]), and thereby formulated into various different noise inequality constraints. Similar definition is also used in Robust Tensor Completion analysis [67], which will be discussed in Section 3.3.2. Second, the way of sampling observations could also be changed and reflected in the constraints. For example, some researchers [50, 143] use the Gaussian measurement operator $\mathcal{G}$ to denote this constraint as a sampling ensemble $\mathcal{G}[\mathcal{X}] = \mathcal{G}[\mathcal{T}]$. In this way, the constraint entries are no longer based on random sampling approach but based on Gaussian random sampling. Since the sampling strategy is also highly correlated with our last question (i.e., In what condition, we could expect to achieve a unique and exact completion?), we leave it to Section 3.4 for further discussion. Besides the above variants, the constraint in Equation (9) could also be expressed in different ways such as $\mathcal{A}(\mathcal{X}) = \mathcal{B}$ in [59] and $\mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T})$ in [85, 110], where $\mathcal{A}$ and $\mathcal{P}_\Omega$ are linear projection operators indicating the sampling operation along with the index set Ω.

3 GENERAL TENSOR COMPLETION ALGORITHMS

After formally defining the LRTC problem, we now introduce some general tensor completion algorithms serving as a stepping stone for more detailed introductions of advanced algorithms in subsequent sections. We start from summarizing some fundamental tensor completion approaches by categorizing them into decomposition based approaches, trace-norm based approaches, and some other variants. These methods are by no means necessarily more straightforward, and non-scalable compared with the advance tensor completion methods. The reason we put them here is either due to their general and primary effect in laying the foundation of building up advanced tools or because of their original focus is not from the perspective of big data analytics.

3.1 Decomposition Based Approaches

Tensor decompositions or factorizations are powerful tools for extracting meaningful, latent structures in heterogeneous, multi-aspect data [106, 154]. Since real-world datasets are often raw and incomplete, many early researches are conducted on tensor decomposition with missing values [2, 20, 85, 199, 206], which could be regarded as the pioneer problem of LRTC. We focus on two most widely used decomposition methods – CP and Tucker decomposition – to begin the overview. Readers interested in tensor decompositions and their applications could go further into [4, 106, 154, 174] for a more comprehensive introduction. Not surprisingly, other decomposition approaches could also be applied to solving the completion problem such as hierarchical tensor (HT) representations [39, 161, 162], PARAFAC2 models [161], and so on.

3.1.1 CANDECOMP/PARAFAC (CP) Based Methods. CP Decomposition was first proposed by Hitchcock [79] and further discussed by Carroll [30] and Harshman [74]. It is formally defined as: Given an Nth-order tensor $\mathcal{X}$, its CP decomposition is an approximation of N loading matrices $\mathbf{A}_n \in \mathbb{R}^{I_n \times R}, n = 1, \ldots, N$, such that,

$$\mathcal{X} \approx [\![\mathbf{A}_1, \ldots, \mathbf{A}_N]\!] = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \ldots \circ \mathbf{a}_r^{(N)}, \tag{13}$$

where R is a positive integer denoting an upper bound of the rank of $\mathcal{X}$, $\mathbf{a}_r^{(n)}$ is the rth column of matrix $\mathbf{A}_n$. We can further unfold $\mathcal{X}$ along its nth mode as follows:

$$\mathcal{X} \quad \overset{\text{unfold}}{\Longrightarrow} \quad \mathbf{X}_{(n)} \approx \mathbf{A}_n(\mathbf{A}_N \odot \ldots \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \ldots \odot \mathbf{A}_1)^\top = \mathbf{A}_n[(\mathbf{A}_k)^{\odot_{k \neq n}}]^\top. \tag{14}$$

Borrowing the idea from weighted least square method applied in the matrix level [100], Bro [20] demonstrates two ways of handling missing data in PARAFAC model, which might be one of the earliest approaches targeting missing data imputation problem in multi-way data analysis. These two ways were further illustrated in his subsequent paper [199] with Tomasi, and the underlying ideas were explicitly or implicitly contained in almost every existing tensor completion methods as far as we know.

The first approach is to alternatively estimate the model parameters while imputing the missing data, which could be regarded as a special expectation maximization (EM) approach under the assumption of Gaussian residuals. Hence, we call it *EM-like approach*. This approach is usually used in alternating projection optimizations. For example, in the work of Tomasi's and Bro's [199], they apply this idea in the PARAFAC model and propose a model called ALS-IS leveraging the standard ALSs optimization Though not explicitly described, the idea of this approach also appears in some other early works for handling missing entries in PARAFAC model [19, 103, 112]. The main idea of this approach is to conduct imputation based on the following equation iteratively:

$$\mathcal{X} = \mathcal{P}_\Omega(\mathcal{T}) + \mathcal{P}_{\Omega^c}(\hat{\mathcal{X}}) = \mathcal{W} * \mathcal{T} + (\mathbf{1} - \mathcal{W}) * \hat{\mathcal{X}}, \tag{15}$$

where $\hat{\mathfrak{X}} = [\![\mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$ is the interim low-rank approximation based on CP decomposition and $\mathfrak{X}$ is the recovered tensor which is used in next iteration for decomposition, Ω^c denotes the complement set of Ω defined as $\Omega^c = \{(i_1, \ldots, i_N) | 1 \leq i_i \leq I_i\} \setminus \Omega$, $\mathcal{W}$ is the observation index tensor with the same size as $\mathcal{T}$:

$$\mathcal{W}(i_1, i_2, \ldots, i_N) = \begin{cases} 1 & \text{if } \mathcal{T}(i_1, i_2, \ldots, i_N) \text{ is observed.} \\ 0 & \text{if } \mathcal{T}(i_1, i_2, \ldots, i_N) \text{ is unobserved.} \end{cases} \tag{16}$$

This approach is easy to be conducted because it only needs to follow certain alternation projection optimization scheme such as ALS optimization [199] while performing imputation at the end of every iteration based on (15). However, with the increasing percentage of missing entries, the convergence rate of the methods based on this approach may be reduced, and the risk of converging to a local minimum would be increased [199].

The second approach mentioned by Bro [20] is to skip the missing value and build up the model based only on the observed part, which we call *Missing-Skipping (MS) approach*. This approach is often found in gradient-based optimizations or probabilistic methods. It can also be realized by masking the missing entries during optimization scheme. The objective function when using this approach is often defined in the following format:

$$J = \sum_{(i,j,k) \in \Omega} \mathcal{D}(\mathfrak{X}_{i,j,k}, \mathcal{T}_{i,j,k}), \tag{17}$$

where $\mathcal{D}$ is an error measure similar to the one defined in Equation (10). The MS approach for tensor completion is first employed in the INDAFAC (INcomplete DAta paraFAC) model [199] to solve the aforementioned effectiveness-decay problem of the EM-like approach. This model is optimized by the Levenberg–Marquardt version of Gauss–Newton (iterative) algorithm inspired by Pentti [152] and is proved to be computationally efficient when the missing ratio is high. Acar [2] also uses the second idea and develop an algorithm named CP-WOPT (CP Weighted OPTimization). Different from a second-order approach employed by INDAFAC, CP-WOPT is optimized based on a first-order gradient and shown to enjoy both effectiveness and higher scalability with different missing ratios. Besides, some probabilistic CP decomposition methods with Bayesian inference are also proposed for solving missing-value prediction problem [214]. These methods treat $\mathcal{D}$ as the log-likelihood function and delete the terms of missing observations from likelihood functions to handle missing values and conduct imputation. To sum up, CP-based completion models, which leverage MS approach, are usually more robust compared with the models with EM-like approach when the missing ratio is large but could be hard to be optimized with existing projection-based algorithms.

3.1.2 Tucker-Based Methods. Tucker decomposition is proposed by Tucker [203] and further developed by Kroonenberg et al. [114] and De Lathauwer et al. [42]. Given an Nth-order tensor $\mathfrak{X}$, its Tucker decomposition is defined as an approximation of a core tensor $\mathfrak{C} \in \mathbb{R}^{\Pi_{n=1}^{N} R_n}$ multiplied by N (orthogonal) factor matrices $\mathbf{A}_n \in \mathbb{R}^{I_n \times R_n}, n = 1, \ldots, N$ along each mode, such that,

$$\mathfrak{X} \approx \mathfrak{C} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \times_3 \cdots \times_N \mathbf{A}_N = [\![\mathfrak{C}; \mathbf{A}_1, \ldots, \mathbf{A}_N]\!], \tag{18}$$

where R_n is a positive integer denoting an upper bound of the rank of $\mathfrak{X}$. We can further unfold $\mathfrak{X}$ along its nth mode as follows:

$$\mathfrak{X} \quad \overset{\text{unfold}}{\Longrightarrow} \quad \mathbf{X}_{(n)} \approx \mathbf{A}_n \mathbf{C}_{(n)} (\mathbf{A}_{n-1} \otimes \ldots \mathbf{A}_1 \otimes \mathbf{A}_N \ldots \otimes \mathbf{A}_{n+1})^\top. \tag{19}$$

Tucker decomposition is also a widely used tool for tensor completion. Similar to CP-based methods, EM-like and MS approaches are still two traditional ways of handling missing values as well as data imputation. Early works by Walczak et al. [206] and Andersson et al. [5] have mentioned the way of using EM-like approach for handling Tucker decomposition with missing values. This method was combined with the higher-order orthogonal iteration (HOOI) algorithm [43] to deal with missing values in pattern-recognition problems in [64, 65]. Some other researchers utilized MS approach such as Karatzoglou et al. [92] and Chu et al. [36]. The former applies high-order SVD (HOSVD) algorithm to track low-rank Tucker subspace for complete multi-dimensional arrays and use the SGD algorithm for optimization with only observed values. The later one considers a probabilistic approach called "pTucker" in which the missing part is naturally deleted in the likelihood functions.

3.1.3 Tucker-Based Methods versus CP-Based Methods. From the former discussion, we can see that both Tucker and CP-based methods could achieve the completion purpose with similar imputation approaches (EM-like approach or MS approach). However, two aspects may lead to different applications of different approaches. (1) *Effectiveness:* Tucker-based methods are shown to be usually more effective than CP-based methods [92, 154] under the same rank assumptions and ideal hyperparameter selections (i.e., the perfect number of latent factors based on the rank assumptions) because their core tensor is more general in capturing complex interactions among components that are not strictly trilinear. Thus, if one only concerns about the completion accuracy without caring about the uniqueness or interpretability of the decomposed latent factors, Tucker-based methods could always be a good choice. (2) *Uniqueness and Interpretability:* Comparing with Tucker decomposition, a good property of the CP decomposition is that it is often unique with the exception of elementary indeterminacies of permutation and scaling [106]. This property could also facilitate easier interpretation of factorized latent matrices with the help of domain knowledge as imposed constraints [31, 208]. Besides, CP-based methods are usually more computationally flexible to deal with large-scale datasets in a distributed way since CP decomposition do not introduce complex core tensor as Tucker decomposition.

Recently, several methods based on HT representations [39, 161, 162] provide a flexible generalization of classical Tucker model to deal with high order tensors. Most of these approaches are optimized using projected gradient methods such as iterative hard thresholding algorithms [161, 162]. The key idea is to use Riemannian gradient iteration method (RGI) that contains the following two steps in each iteration: (1) to perform a gradient step in the ambient space, and (2) map the result back to the low-rank tensor manifold with hierarchical singular value thresholding procedure. Another variant projected gradient method called Riemannian optimization approach [39, 95, 110] is also considered based on the hierarchical representations for manifold construction. For its particularity, we leave it later in Section 3.3 for a more detailed introduction.

3.2 Trace Norm Based Approaches

Matrix trace norm (or nuclear norm) has been popularized as a convex surrogate of non-convex rank function for solving the matrix completion problem. For matrices with bounded operator norm, it has been shown to be the tightest lower bound of matrix rank function among all possible convex approximation [165]. Generalized from this matrix relaxation, Liu et al. [129] first defined the tensor trace norm as the combination of the trace norms of its unfoldings and reformulated the tensor rank minimization problem as a convex optimization problem. Signoretto et al. [176] further generalized the tensor trace norm to be a particular case of Shatten-p,q norm. In fact, a more direct generalization from matrix trace norm to tensor trace norm is to utilize the definition of tensor n-rank. By substituting the tensor rank with a linear combination of tensor n-rank, it

could be further relaxed with trace norms for defining a low-n-rank tensor pursuit problem [59, 130]:

$$\begin{aligned} \underset{\mathfrak{X}}{\text{minimize}} \quad & \text{rank}_*(\mathfrak{X}) \overset{\text{substitute}}{\Longrightarrow} \sum_{n=1}^{N} \alpha_n rank(\mathbf{X}_{(n)}) \overset{\text{relax}}{\Longrightarrow} \sum_{n=1}^{N} \alpha_n \|\mathbf{X}_{(n)}\|_* \\ \text{subject to} \quad & \mathfrak{X}_{\mathbf{\Omega}} = \mathfrak{T}_{\mathbf{\Omega}} + \mathfrak{E}_{\mathbf{\Omega}}, \end{aligned} \tag{20}$$

where $\sum_{n=1}^{N} \alpha_n$ is usually defined as 1 to maintain the consistency with the matrix trace norm [130]. We call this model SNN (sum of the nuclear norm) model. Though these unfolding matrices cannot be optimized independently because of their multi-linear correlations, this convex relaxation allows us to solve the completion problem without predefining the tensor rank, which is more tractable in practice. An equivalent problem in the case of Gaussian noise could be formulated as

$$\underset{\mathfrak{X}}{\text{minimize}} \quad \frac{\lambda}{2}\|\mathcal{P}_{\mathbf{\Omega}}(\mathfrak{X} - \mathfrak{T})\|_{\text{F}}^2 + \sum_{n=1}^{N} \alpha_n \|\mathbf{X}_{(n)}\|_*, \tag{21}$$

where $\lambda > 0$ is a tradeoff constant. To solve problem (21), Liu et al. [129] propose a simple algorithm optimized by the block coordinate decent.[2] This algorithm is the first trace-norm based tensor completion algorithm, and to our best knowledge, it is the first tensor completion algorithm without fixing the rank of the tensor in advance as decomposition-based approaches do. A simplified version of it and two advanced algorithms were proposed in their subsequent journal paper [130]. A more popular approach for solving the above problem is to apply splitting method [59, 130, 176, 200], e.g., Alternating Direction Method of Multipliers (ADMM) which is a special case of Douglas–Rachford splitting method.[3] As there are only slight changes in these methods, we first consider the noise exist case and introduce the integrated approach of Grandy et al. [59] and Tomioka et al. [200], and then come back to the noiseless case discussed in [59, 130, 176].

Under the paradigm of ADMM, the completion problem defined in Equation (21) could be reformulated by introducing auxiliary matrices $\{\mathbf{Z}_{(n)}\}$, $n = 1, \ldots, N$, as

$$\begin{aligned} \underset{\mathfrak{X}, \mathbf{Z}_1, \ldots, \mathbf{Z}_N}{\text{minimize}} \quad & \frac{\lambda}{2}\|\mathcal{P}_{\mathbf{\Omega}}(\mathfrak{X} - \mathfrak{T})\|_{\text{F}}^2 + \sum_{n=1}^{N} \alpha_n \|\mathbf{Z}_n\|_* \\ \text{subject to} \quad & \mathbf{Z}_{(n)} = \mathbf{X}_n, n = 1, \ldots, N. \end{aligned} \tag{22}$$

A corresponding augmented Lagrangian objective function could be derived as

$$\mathcal{L}_\eta = \frac{\lambda}{2}\|\mathcal{P}_{\mathbf{\Omega}}(\mathfrak{X} - \mathfrak{T})\|_{\text{F}}^2 + \sum_{n=1}^{N} \left(\alpha_n \|\mathbf{Z}_n\|_* + < \mathbf{Y}_n, \mathbf{Z}_n - \mathbf{X}_{(n)} > + \frac{\eta}{2}\|\mathbf{Z}_n - \mathbf{X}_{(n)}\|_{\text{F}}^2 \right). \tag{23}$$

where $\{\mathbf{Y}_n\}_n$ are the Lagrange multipliers, $\eta > 0$ is a penalty parameter. Following the standard updating scheme of ADMM, $\{\mathbf{Z}_n\}_n$ and $\{\mathbf{Y}_n\}_n$ could be iteratively updated as follows

$$\begin{aligned} \mathbf{Z}_n &\leftarrow SVT_{\frac{\alpha_n}{\eta}} \left(\mathbf{A}_n - \frac{\mathbf{Y}_n}{\eta} \right), \\ \mathbf{Y}_n &\leftarrow \mathbf{Y}_n + \eta(\mathbf{Z}_n - \mathbf{A}_n), \quad n = 1, 2, \ldots, N, \end{aligned} \tag{24}$$

where $SVT_{\frac{\alpha_n}{\eta}}$ is the singular value thresholding operator [22] that is a shrinkage operator defined as $SVT_\delta(\mathbf{A}) = \mathbf{U}(\text{diag}\{\sigma_i - \delta\})_+\mathbf{V}^\top$, where $\mathbf{U}(\text{diag}\{\sigma_i\}_{1 \le i \le r})\mathbf{V}^\top$ represents the singular value decomposition of the matrix $\mathbf{A}$. For any matrix $\mathbf{X}$, $\mathbf{X}_+ = \max\{\mathbf{X}, 0\}$, where $\max\{\cdot, \cdot\}$ denotes an

[2]Though the formulation solved in [129] is kind of different from (32), but they are the same in essence.
[3]Douglas-Rachford splitting method has also been shown to be a special case of point proximal methods, which is useful for accommodating various non-smooth constraints. Details are discussed in [48].

ALGORITHM 1: ADMM for noise exist trace-norm based tensor completion

Input: $\mathcal{T}$, Ω, $\{\alpha_n\}_{n=1}^N$, λ, η, ρ, *tol*
Output: $\mathcal{X}$
Initialize $\mathbf{Z}_n = \mathbf{Y}_n = \mathbf{0}$;
repeat
  for $n = 1 : N$ **do**
    Update $\mathbf{Z}_n$ using Equation (24);
  end
  Update $\mathcal{Y}$ using Equation (24);
  Update $\mathcal{X}$ with EM-like Approach;
until *Certain Stop Criterion is Satisfied*;

element-wise operator. The update of $\mathcal{X}$ could have several choices, one can use the EM-like approach as we describe above [130] or use the exact [59, 200] or inexact approaches [59]. The exact updating approach is describe in Algorithm 1. To deal with the noiseless case, Signoretto et al. [176] and Gandy et al. [59] propose to decrease the value λ in Equation (32) while Liu et al. [130] directly delete the first term and prove their algorithm to be more efficient.

Beyond these traditional approaches focusing on minimizing the sum of nuclear-norm, several variants are proposed, which could be categorized into two types. One is to introduce variant trace norm structures, and another is to combine with decomposition based approaches by transferring the trace-norm constraints on factorized matrices.

SNN model is neither the tightest relaxation of tensor rank nor an optimal solution considering the sample size requirement. To reduce the sample size required for completion, the first type of methods is mainly focusing on proposing better convex relaxation based on new trace norm structure. Mu et al. [143] suggest to use a single powerful regularizer rather than a combination structure as SNN model, i.e., apply trace norm on a more balanced unfolding matrix rather than using the summation of trace norm. Besides numerical experiments, the theoretical analysis of sampling bound is also conducted by exploiting the sharp properties of the Gaussian measurement. Different from matrix trace-norm structure, some other trace-norm based structures are also proposed such as tensor-nuclear-norm [104, 234] and incoherent trace norm [230]. They either target new rank structure (tensor tubal-rank) or consider the incoherence condition. As most of them are focusing on the analyzing the sampling size requirement based on different sampling approaches, we will stress their theoretical findings in the statistical analysis section.

The second type of approaches is to add trace norm constraints on factorized matrices rather than on unfolding matrices, which could be treated as mixed approaches of trace-norm based methods and decomposition based methods. These methods are mainly focusing on reducing the computation complexity since trace norm minimization methods are often costly and require SVD decomposition of potentially large matrices [110]. Also, they are more flexible in incorporating different structures [221] on decomposition matrices and alleviate the requirement of the predefined ranks for decomposition based methods [132]. Ying et al. [221] focus on the exponential signal completion problem. As a signal model only has $\mathcal{O}(R)$ degree of freedoms,[4] they explore to exploit an exponential structure of the factor vectors to reduce the number of samples for exact recovery. Each factor is first transformed into a Hankel matrix to promote exponential structure, and then nuclear norms are applied to force these Hankel matrices to be low-rank.

[4]A signal model expresses each entry as $x_{i_1,\ldots,i_N} = \sum_{r=1}^{R}(d_r \prod_{n=1}^{N} z_{n,r}^{i_n-1})$. R is the CP rank.

To address the computational complexity problem, some researchers utilize QR decomposition [131] while some researchers add trace-norm constraints on CP or Tucker factor matrices [132]. Their approaches both apply the trace norm onto the factorized matrices rather than the original unfolding matrices for acceleration while the latter one also alleviates the rank predefinition problem faced by decomposition-based approaches to some extent. Another interesting and straightforward variation of trace norm approach is to transform matrix trace norm into Frobenius norm. In [187], for a matrix $\mathbf{X}$ with low-rank decomposition $\mathbf{X} = \mathbf{U}\mathbf{V}^\top$, an equivalent expression of its trace norm could be defined as

$$\|\mathbf{X}\|_* := \min_{\mathbf{U},\mathbf{V}} \frac{1}{2}\{\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2\}. \tag{25}$$

In a similar spirit to the matrix case, Mardani et al. [138] further extends it into the tensor level and apply it in the online situation which we will introduce in Section 6. Other approaches, which focus on relaxing nuclear norm to new convex relaxations in order to reduce both the computational complexity and the number of sample sizes could also be found [163].

3.3 Other Variants

To provide a more comprehensive and informative introduction, we include several important variants in this section as a supplementary for general tensor completion methods.

3.3.1 Non-Negative Constrained Approaches. Real-world applications sometimes require non-negative constraints such as image completion [216], medical data analysis [80, 208] and so on. To impute missing entries of non-negative tensors, a popular way is to add non-negative constraints on latent factor matrices based on decomposition models [37]. Different optimization method could be applied such as Block Coordinate Descent [216] and ADMM framework [208] with an iteratively non-negative threshold. For non-negative tensors with integer entries, Takeuchi and Ueda [193] use generalized Kullback–Leibler (gKL) divergence by analogy with several matrix-based approaches [21, 44, 217].

3.3.2 Robust Tensor Completion. Robust data analysis plays an instrumental role in dealing with outliers, gross corruptions (non-Gaussian noises) in completion problem [67]. Drawing upon the advances in robust PCA analysis [27], robust tensor completion is to complete a tensor $\mathcal{T}$ by separating it into a low-rank part $\mathcal{X}$ plus a sparse part $\mathcal{E}$ to capture the different noise patterns. The objective function could be formulated as follows:

$$\begin{aligned} \underset{\mathcal{X},\mathcal{E}}{\text{minimize}} \quad & \text{rank}_*(\mathcal{X}) + \|\mathcal{E}\|_0 \\ \text{subject to} \quad & \mathcal{X}_\Omega + \mathcal{E}_\Omega = \mathcal{T}_\Omega. \end{aligned} \tag{26}$$

Similar trace-norm strategy could be added to the low-rank part for a relaxation of the rank minimization and l_1 norm is often used as a convex surrogate to relax the l_0 norm. The completion problem could be well addressed by the joint minimization of trace norm and l_1 norm. Li et al. [126] utilize a block coordinate decent method similar with [129] and shows promising results in separating corrupted noise and image restoration. Goldfarb and Qin [67] study the problem of robust LRTC in a convex ADMM optimization framework and provide both theoretical and practical analysis of convex and non-convex models. Recent work of Jain [86] also adopts ADMM framework to complete noisy tensors with one of the CP factors being sparse. Besides ADMM framework, Zhao et al. [238] consider the probabilistic framework with a fully Bayesian treatment optimized by variational Bayesian inference approach. Javed et al. [87] provide an online stochastic optimization method for robust low-rank and sparse error separation in addressing background subtraction problem.

3.3.3 Riemannian Optimization. Riemannian optimization technique has gained increasing popularity in recent years [39]. The idea of it lies in an alternative treatment focusing on the fixed-rank tensor completion problem. By embedding the rank constraint into the search space, an unconstrained problem on a smooth manifold $\mathcal{M}_{\mathbf{r}} = \{\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N} | \text{rank}_{\text{tc}}(\mathcal{X}) = \mathbf{r} = (r_1, \ldots, r_N)\}$ is defined as follows:

$$\underset{\mathcal{X} \in \mathcal{M}_{\mathbf{r}}}{\text{minimize}} \quad \frac{1}{2} \|\mathcal{P}_{\mathbf{\Omega}}(\mathcal{X} - \mathcal{T})\|_{\text{F}}^2. \tag{27}$$

Riemannian optimization is an iteratively gradient projection approach based on a two-step procedure, i.e., the projected gradient step and a retraction step. A smooth manifold and its tangent space, a proper definition of Riemannian metric for gradient projection and the retraction map are three essential settings for Riemannian optimization. As the Tucker decomposition gives an efficient representation of tensors in $\mathcal{M}_{\mathbf{r}}$, the factorized orthogonal matrices belonging to Stiefel manifold of matrices[5] are usually used for the parametrization of manifold $\mathcal{M}_{\mathbf{r}}$ and its tangent space [95, 110]. Besides the Tucker manifold $\mathcal{M}_{\mathbf{r}}$, other choices of the smooth manifold are also used such as hierarchical Tucker space for handling high-dimensional applications [39]. After defining the manifold and its tangent space, the most important ingredients in Riemannian approach is to set the Riemannian metric and gradient. Kressner [110] focused on the search space and exploited the differential geometry of the rank constraint based on Tucker decomposition. Kasai et al. [95] laid emphasis on the cost function $\|\mathcal{P}_{\mathbf{\Omega}}(\mathcal{X} - \mathcal{T})\|_{\text{F}}^2$ by introducing the block diagonal approximation of its Hessian and defined a novel metric on the tangent space of $\mathcal{M}_{\mathbf{r}}$ and its quotient manifold. They further generalized a preconditioned conjugate gradient algorithm proposed in [141] with a total computational cost $O(|\Omega| r_1 r_2 r_3)$ for a three-order tensor. Finally, the retraction map is used map the updated tensor back to the low-rank tensor manifold. Maps such as HOSVD method [110] and hierarchical singular value thresholding [161], which satisfy the necessary properties of a retraction mentioned in [110] could be used.

There are still various of interesting methods for solving the completion problem such as alternative convex relaxation approach for Tucker ranks [169], adaptive sampling methods [18, 111] or various matrix-based methods [89]. As it is unpractical to cover all the bases, readers interested in these approaches are encouraged to explore them in the original paper. Finally, we list several representative algorithms we mentioned in Table 2 for summing up and comparison.

3.4 Statistical Assumption and Theoretical Analysis

In this section, we introduce some primary statistical assumptions and theoretical analysis of the completion feasibility. The goal is to elaborate more theoretical foundations of previously mentioned methods before delving into some advanced tensor completion techniques in big data analytics. These assumptions also provide the answers to the question we leave in Section 2.2, i.e., In what condition, we could expect to achieve a unique and exact completion?[6] It should be noted that, in general, one cannot expect to fully recover every low-rank tensor from a sample of its entries. For example, similar to the matrix situation described in [26], if a tensor has only one nonzero entry, it is not guaranteed to recover it even we are able to randomly select 90% of its entries, since it is with 10% probability to obtain only zero samples. Thus, a more appropriate question for the completion feasibility should be the following: In what condition, could we expect to achieve a unique and exact completion with high probability?

[5]Stiefel manifold of matrices is defined as $\{\mathbf{W} \in \mathbb{R}^{n \times p} | \mathbf{W}^\top \mathbf{W} = \mathbf{I}_p\}, \ n \geq p$. [149].

[6]In the matrix completion problem described in [26], this question is answered as "If the number of measurements is sufficiently large, and if the entries are sufficiently uniformly distributed as above, one might hope that there is only one low-rank matrix with these entries."

Table 2. Comparison between General Tensor Completion Algorithms

Rank assumption	Name	Core objective	Optimization method	Imputation
CP	ALS-SI [20, 199]	CPD	ALS	EM-like
	INDAFAC [199]	CPD	Levenberg–Marquadt	MS
	CP-WOPT [2]	CPD	Nonlinear conjugate gradient	MS
	BPTF [214]	Probablistic CPD	Bayesian (MAP)	MS
	STC [111]	CPD+Adaptive Sampling	RLS	MS
	Tensor L.S. [18]	CPD+Adaptive Sampling	ALS	MS
	[85]	CPD	ALS	MS
Tucker	TNCP [134]	SNN+CPD	ADMM	EM-like
	ALS/IA (Tucker3/IA) [5, 206]	TD (HOOI)	ALS	EM-like
	pTucker [36]	Probablistic TD	EM/Bayesian (MAP)	EM/MS
	MRTF [92]	TD (HOSVD)	SGD	MS
	[56]	TD	NCGM [2]	MS
	geomCG [110]	Riemannian Opt.	Riemannian NCG	MS
	Riemannian Preconditioning [95]	Riemannian Opt.	Riemannian Preconditioned NCG	MS
	SiLRTC [129, 130]	SNN	Block CD	EM-like
	FaLRTC [130]	SNN	Smoothing Scheme	EM-like
	HaLRTC [130]	SNN	ADMM	EM-like
	[200]	SNN	ADMM	EM-like
	(A/E)-CMLE [176]	SNN	ADMM	EM-like
	ADM-TR (E) [59]	SNN	ADMM	EM-like
	Square Deal [143]	Single NN	Matrix ALM [128]	EM-like
	[169]	Alternative Relaxation	ADMM	EM-like
	[230]	Incoherent Tensor Norms	Gradient based	MS
Hierarchical Tucker	HTTC [39]	Hierarchical TD	Steepest Descent/CG	MS
	TIHT [161, 162]	Hierarchical TD	Iterative Hard Thresholding	MS
Tubal-rank	t-SVD [234]	Tensor-nuclear-norm	ADMM	EM-like

CPD: CP decomposition. TD: Tucker Decomposition. SNN: Sum of nuclear norms. MAP: Maximum a posteriori estimation.

3.4.1 Sampling Assumption. Tensor completion is usually based on random sampling assumption in which we assume the partially observed entries are uniformly random sampled from the original tensor. In matrix case, Bernoulli sampling [26] or independent sampling with replacement [164] are always assumed to simply the random sampling approach. Generalizing these assumptions such as Bernoulli sampling to the tensor case is straightforward [82]. Several other sampling assumptions are also used for either the convenience of theoretical demonstration or the practicability in real-world applications, e.g., Gaussian measurements [143], Fourier measurements [162], and adaptive sampling [18, 111]. Gaussian random measurement is a Gaussian linear map $\mathcal{G} : \mathbb{R}^{I_1 \times \ldots \times I_N} \to \mathbb{R}^m$ defined by m tensors $\mathcal{G}_i \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ via $[\mathcal{G}(\mathcal{X})](i) =< \mathcal{G}_i, \mathcal{X} >$, where each tensor $\mathcal{G}_i$ has i.i.d. standard normal entries. It enjoys sharp theoretical tools [143] and has been well-studied in signal processing field [202] especially in the compressed sensing area [45]. Another widely studied sampling measurement is random Fourier measurement constructed by Fourier transformation:

$$\mathcal{G} : \mathbb{R}^{I_1 \times \ldots \times I_N} \to \mathbb{R}^m, \quad \mathcal{G}(\mathcal{X}) = \frac{1}{\sqrt{m}} \mathcal{P}_\Omega \mathcal{F}_N(\mathcal{X}), \tag{28}$$

where $\mathcal{P}_\Omega$ is the random subsampling operator, $\mathcal{F}_N$ is the N-dimensional Fourier transformation, which is formally defined as:

$$\mathcal{F}_N(\mathcal{X})(j_1,\ldots,j_N) = \sum_{k_1=1}^{I_1}\cdots\sum_{k_N=1}^{I_N}\mathcal{X}(k_1,\ldots,k_N)e^{-2\pi i\sum_{n=1}^{N}\frac{k_n j_n}{I_n}}. \tag{29}$$

Comparing with sub-Gaussian measurements which usually serve as the benchmark guarantees of required observations in exact low-rank recovery, Fourier measurement offers advantages on its more structured sampling property and available fast multiplication routines [162].

In some real-world applications, sampling populations are sparse but clustered. Under these circumstances, non-adaptive sampling methods may not be able to achieve exact recovery using a similar size of samples as usual. Instead, varies adaptive sampling approaches [18, 111] are derived to relax the incoherence assumptions which is another important assumption introduced next.

3.4.2 Incoherence Assumption. Tensor incoherence assumption is a generalized conception of matrix incoherence arisen in compressed sensing. It is closely correlated with passive uniformly samplings where each entry should have a comparable amount of information to guarantee the recovery tractable. The opposite situation is "coherent," which means most of the mass of a tensor concentrate in a small number of elements and these informative elements has to be completely sampled. It may lead to a missing in mass when sampling methods are uniformly at random and independent of the underlying distribution of the tensor. Mathematically, the coherence of an r-dimensional linear subspace $\mathbf{U}$ of $\mathbb{R}^N$ is defined as

$$\mu(\mathbf{U}) = \frac{N}{r}\max_{1\le i\le N}\|\mathbf{P}_\mathbf{U}\mathbf{e}_i\|_2^2 = \frac{\max_{1\le i\le N}\|\mathbf{P}_\mathbf{U}\mathbf{e}_i\|_2^2}{N^{-1}\sum_{i=1}^{N}\|\mathbf{P}_\mathbf{U}\mathbf{e}_i\|_2^2}, \tag{30}$$

where $\mathbf{P}_\mathbf{U}$ is the orthogonal projection onto subspace $\mathbf{U}$ and $\mathbf{e}_i$'s are the canonical basis for $\mathbb{R}^N$. Based on this definition, the tensor μ-incoherent is defined as

$$\mu_i(\mathcal{X}) := \mu(\mathcal{L}_i(\mathcal{X})) = \mu(\mathbf{X}_{(i)}) \le \mu, \quad (i = 1,\ldots,N), \tag{31}$$

where $\mathcal{L}_i(\mathcal{X})$ is the linear subspace spanned by the mode-i fibers. In [82], besides this N-mode incoherence, the tensor incoherence conditions are further extended to a new "mutual incoherence" based on the singular value decompositions of the matricizations $\mathbf{X}_{(n)}$'s. Based on different sampling strategies and the incoherence assumptions, now we discuss the theoretical bounds of the observed entries we need to recover the tensors using different algorithms.

3.4.3 Number of Observed Entries. To guarantee a unique reconstruction of the original tensor, a theoretic lower bound of the number of observed entries is often required. In a matrix case, a given incoherent $m \times n$ matrix could be recovered with high probability if the sample size is larger than $Cnr\dot{p}olylog(n)$ for some constant C, where r is the matrix rank [70]. Generalizing this result to the high-order situation is not straightforward [143]. Tomioka et al. [201] first conducted the statistical analysis of low-rank tensor recovery. They provided the first reliable recovery bound for the SNN (sum of nuclear norms) model (20) under the assumption of Gaussian measurement. In their analysis, for a given Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I\times\ldots\times I}$ of Tucker rank $(r, r, \ldots, r)$, $\mathcal{X}$ could be completed with high probability if the number of observed entries is larger than CrI^{N-1}, where C is a constant. This corollary is further proved by Mu et al. [143] to be a necessary condition for SNN model and the probability is shown to be $1 - 4\exp(-\frac{(\kappa-m-2)^2}{16(\kappa-2)})$, where $\kappa = \min_n\{\|\mathbf{X}_{(n)}\|_*/\|\mathcal{X}\|_\mathrm{F}\} \times I^{N-1}$.

However, this necessary bound for SNN model is far from satisfactory because one only needs no more than $r^N + rIN$ parameters to describe the tensor $\mathcal{X}$ defined above based on the Tucker decomposition in Equation (18). Furthermore, Mu et al. prove that $(2r)^N + 2rIN + 1$ measurements are sufficient to complete $\mathcal{X}$ almost surely based on the non-convex model formulated in Equation (11). In fact, the freedom of a tensor $\mathcal{X} \in \mathbf{R}^{I_1 \times \ldots \times I_N}$ with Tucker rank $(r_1, \ldots, r_N)$ is only $\prod_{n=1}^{N} r_n + \sum_{n=1}^{N}(r_n I_n - r_n^2)$ [110]. This result can be calculated based on an easy two-step construction: (1) The number of entries of the core tensor is randomly valued which demonstrate its freedom is $\prod_{n=1}^{N} r_n$. (2) To construct the rest part, we can iteratively unfold the core tensor along each mode to get the mode-n unfolding matrix of size $r_n \times C$, where $C = I_1 \times \ldots \times I_{n-1} \times r_{n+1} \times \ldots \times r_N$ (n=1,...,N). Each of this matrix represents r_n uncorrelated vectors which could be used to construct the rest $I_n - r_n$ vectors with freedom $r_n(I_n - r_n)$. Thus, the total freedom is $\prod_{n=1}^{N} r_n + \sum_{n=1}^{N}(r_n I_n - r_n^2)$. This analysis motivates researchers to find lower bounds as well as advanced methods for tensor completion. In [143], the authors improve the SNN model into a better convexification. Rather than using a combination of all mode-n trace norms, they suggest using an individual trace norm of a more balanced matrix defined by $\mathcal{X}_{[j]} = \text{reshape}(\mathbf{X}_{(1)}, \prod_{n=1}^{j} I_n, \prod_{n=j+1}^{N} I_n)$, where $j \in [N] := \{1, \ldots, N\}$. Under this square deal setting, $CrI^{[\frac{N}{2}]}$ entires are sufficient for recovery when $\text{rank}_{CP}\mathcal{X} = r$, where $\mathcal{X} \in \mathbf{R}^{I_1 \times \ldots \times I_N}$. Only $Cr^{[\frac{N}{2}]}I^{[\frac{N}{2}]}$ observations are needed to recover a tensor $\mathcal{X}$ of Tucker rank $(r, \ldots, r)$ with hight probability.

Jain et al. [85] stress the setting of random sampling (rather than Gaussian random sampling in [143]) and consider that the square deal approach requires each observation to be a "dense random projection" rather than a single observed entry. They generate a matrix alternating minimization technique to the tensor level with a CP decomposition approach and prove that a symmetric third order tensor $\mathcal{X} \in \mathcal{R}^{I \times I \times I}$ with CP rank-r could be correctly reconstructed from $O(I^{\frac{3}{2}} r^5 \log^4 I)$ randomly sampled entries. In a similar spirit, Barak and Moitra [13] give an algorithm based on sum-of-squares relaxation for prediction of incoherent tensors. Instead of exact recovery, their main result shows that $O(I^{\frac{3}{2}} r^2 \log^4 I)$ observations could guarantee an approximation with an explicit upper bound on error. However, this sum-of-squares approach is point out not to be scale well to large tensors and is substituted by a spectral approach proposed by Montanari et al. [142] with matching statistical guarantee (i.e., the required sample size). Recently, Yuan and Zhang [230] explore the correlations between tensor norms and different coherence conditions. They prove that an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I \times \ldots \times I}$ with CP rank-r could be completed entirely from $O((r^{\frac{N-1}{2}} I^{\frac{3}{2}} + r^{N-1} I)(\log(I))^2)$ uniformly sampled entries through a proper incoherent nuclear norm minimization.

Recently, the theme of adaptive sensing emerges as an efficient alternative to random sampling approach in large-scale data obtaining and processing. By exploiting adaptivity to identify highly informative entries, the required number of observed entries could be substantially reduced. Krishnamurthy et al. [111] propose an adaptive sampling method with an estimation algorithm which could provably complete an Nth-order rank-r tensor with $O(Ir^{N-\frac{1}{2}} N^2 \log(r))$ entries. Subsequently, Bhojanapalli and Sanghavi [18] give a new sampling approach based on a parallel weighted ALS algorithm. They show that a symmetric rank-r third order tensor $\mathcal{X} = \sum_{i=1}^{r} \sigma_i^* \mathbf{U}_i^* \otimes \mathbf{U}_i^* \otimes \mathbf{U}_i^*$, where $\mathbf{U}^* \in \mathbb{R}^{I \times r}$ is an orthonormal matrix, could be proved to be exactly recovered from $O((\sum_{i=1}^{I} \|(\mathbf{U}^*)^i\|^{\frac{3}{2}})^2 I r^3 \kappa^4 log^2(I))$, where $(\mathbf{U}^*)^i$ are n rows of $\mathbf{U}^*$, κ is a restricted condition number. Although both algorithm crucially relies on the adaptive sampling technique that does not generalize to random samples, they achieve the completion purpose with little or no incoherence assumptions on the underlying tensor. In the end, we summarize the sufficient sampling lower bounds of several existing approaches for exact recovery as well as their sampling methods and incoherent assumptions in Table 3.

Table 3. Statistical Assumptions and Bounds (Third-Order Tensor of Size $I \times I \times I$)

Rank assumption	Method	Sampling method	Incoherent and other conditions	Requirement for exact recovery (w.h.p)
$\text{Rank}_{\text{cp}} = r$	Provable TF [85]	Random Sampling	Incoherent, orth. CPD, Symmetric	$\mathcal{O}(r^5 I^{\frac{3}{2}} \log^4(I))$
	ITN [230]	Uniformly Random Sample	Incoherent	$\mathcal{O}((rI^{\frac{3}{2}} + r^2 I)\log^2(I))$
	NTC [14]	Uniformly Random Sample	Incoherent	$\mathcal{O}((rI^{\frac{3}{2}})\log^4(I))$
	STC [111]	Adaptive Sampling	Partial Incoherent	$\mathcal{O}(Ir^{\frac{5}{2}}\log(r))$
	Parallel WALS [18]	Adaptive Sampling	orthogonal CPD, Symmetric	$\mathcal{O}((\sum_{i=1}^{I} \|(\mathbf{U}^*)^i\|^{\frac{3}{2}})^2 Ir^3\kappa^4\log^2(I))$
	M-norm [66]	Uniformly Random Sample	Incoherent	$O(r^6 I)$
$\text{Rank}_{\text{tc}} = (r, r, r)$	SNN [143, 201]	Gaussian measurements	N.A.	$\mathcal{O}(rI^2)$
	Square Deal [143]	Gaussian measurements	N.A.	$\mathcal{O}(r^{\lceil \frac{3}{2} \rceil} I^{\lceil \frac{3}{2} \rceil})$
	GoG [211]	Uniformly Random Sample	Incoherent	$\mathcal{O}(r^{\frac{7}{2}} I^{\frac{3}{2}} log^{\frac{7}{2}}(I) + r^7 I\log^6(I))$
Tubal-rank r	t-SVD [233]	Uniformly Random Sample	Incoherent	$\mathcal{O}(rI^2\log(I))$

Rank_{tc} means Tucker rank.

4 TENSOR COMPLETION WITH AUXILIARY INFORMATION

In the previous section, tensor completion methods are introduced along with some key statistical assumptions according to the completion feasibility. Though the techniques varied, both experimental results and theoretical analysis reflect a natural and intuitive phenomenon that: with the increasing ratio of missing entries, the prediction accuracy tends to be significantly decreased. In real-world data-driven applications, besides the target tensor object, additional side information such as spatial and temporal similarities among objects or auxiliary coupled matrices/tensors may also exist [60, 61, 144, 207, 213]. These heterogeneous data sources are usually bonded and compensate with each other and could serve as potential supplements to improve the completion quality especially when the missing ratio of the target tensor is high. In this section, we provide an overview of related approaches and mainly introduce two ways of incorporating variety auxiliary information in existing literatures of tensor completion – similarity based approaches and coupled matrices/tensors factorization.

4.1 Similarity Based Approaches

Inspired by relation regularized matrix factorization proposed by Li et al. [124], Narita et al. [144] use two regularization methods called "within-mode regularization" and "cross-mode regularization" respectively to incorporate auxiliary similarity among modes for tensor factorization. These methods are all based on EM-like approach combined with Tucker or CP decomposition. The key idea is to construct within-mode or cross-mode similarity matrices and incorporate them as following regularization terms:

$$\begin{aligned} R_{\text{within}}(\mathcal{X}; \mathbf{A}_1, \ldots, \mathbf{A}_N) &= \sum_{n=1}^{N} \sum_{i,j=1}^{I_n} \mathbf{S}_n(i,j) \|\mathbf{A}_n(i,:) - \mathbf{A}_n(j,:)\|_{\text{F}}^2 = \sum_{n=1}^{N} \text{tr}(\mathbf{A}_n^\top \mathbf{L}_n \mathbf{A}_n) \\ R_{\text{cross}}(\mathcal{X}; \mathbf{A}_1, \ldots, \mathbf{A}_N) &= \text{tr}((\mathbf{A}_1 \otimes \ldots \otimes \mathbf{A}_N)^\top \mathbf{L}(\mathbf{A}_1 \otimes \ldots \otimes \mathbf{A}_N)), \end{aligned} \tag{32}$$

where $\mathbf{A}_n$ is the mode-n latent matrix of CP or Tucker decomposition, $\mathbf{S}_n$ is the mode-n auxiliary similarity matrix of size $I_n \times I_n$. $\mathbf{L}_n$ is the Laplacian matrix of $\mathbf{S}_n$ and $\mathbf{L}$ is defined as the Laplacian matrix of $\mathbf{S}_1 \otimes \ldots \otimes \mathbf{S}_N$, which is also called Kronecker product similarity in the early work of Kashima et al. [97]. By combining them with standard decomposition approaches and applying to tensor completion problem, the auxiliary information was shown to improve completion accuracy especially when observations are sparse. In fact, an early work [239] has taken advantage of the "within-mode regularization" and combine it with the coupled matrix method for the mobile

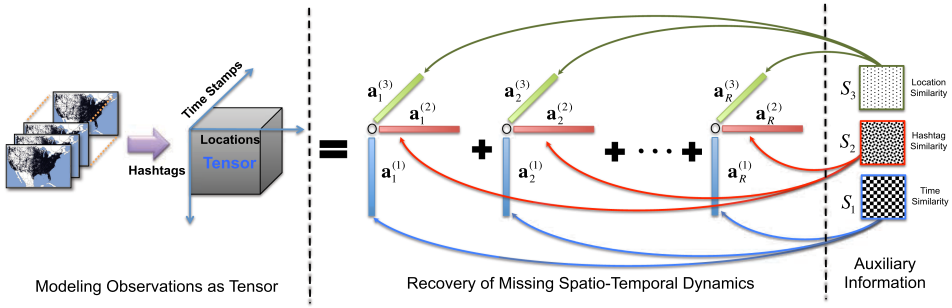

Fig. 3. Illustration of the AirCP framework [61].

recommendation problem. Similar idea of these two approaches have also been used in [11, 61] and [33] respectively.

To effectively incorporate auxiliary information, the main concentration of these similarity-based methods is to define the within-mode or cross-mode similarity matrices. As an example, we briefly introduce one state-of-the-art model leveraging the within-mode auxiliary similarity for tensor completion. Figure 3 illustrates the general framework of the AirCP (Auxiliary Information Regularized CANDECOMP/PARAFAC) model proposed by Ge et al. [61] to impute the missing online memes by incorporating their auxiliary spatio-temporal information. By modeling the observed data as a third-order tensor (left), the authors seek to recover the missing entries with a CP-based method (center) by integrating auxiliary information like the relationships between locations, memes and times (right). Three similarity matrices $\{\mathbf{S}_i\}_{i=1,2,3}$ are defined for three modes adopting the geographical distance, temporal relationships or the fusions of multiple similarity measures. For example, the spatial relationships ($\mathbf{S}_2$) and the temporal relationships ($\mathbf{S}_3$) are modeled as the fusion of two similarity measures and the tri-diagonal matrix respectively as follows:

$$\mathbf{S}_2(i,j) = \tau \mathbf{S}_{GD}(i,j), +(1-\tau)\mathbf{S}_{AS}(i,j) \quad (1 \le i,j \le I_2), \tag{33}$$

$$\mathbf{S}_3 = \begin{bmatrix} 0 & 1 & 0 & \dots \\ 1 & 0 & 1 & \dots \\ 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{34}$$

where $\mathbf{S}_{GD}$ is the geographical distance matrix defined as $\mathbf{S}_{GD}(l_i, l_j) = \exp(-\frac{\mathrm{Dist}(l_i,l_j)^2}{2\alpha^2})$ in which α is a predefined dispersion constant and $\mathrm{Dist}(l_i, l_j)$ is the distance between location l_i and location l_j based on their GPS coordinates; $\mathbf{S}_{AS}$ is a heuristics similarity called adoption similarity [61], which we do not specify it here for the ease of presentation; τ is a balancing hyperparameter. Similar to these similarity matrices defined in AirCP, there are also other ways to define the similarity matrices, which are usually guided by different domain knowledge varying from applications to applications [11, 33]. From the perspective of probabilistic approaches [15, 117, 236, 237], these similarity matrices among each mode could all be treated as the inverse of kernel matrices in Gaussian prior distributions defined for each factor matrices.

4.2 Coupled Matrices/Tensors Factorization

Another popular way of incorporating auxiliary information is to couple tensors or matrices together for jointly factorization and imputation. Deriving from coupled matrix factorization [179, 181], the way of coupled side information has also been widely developed in multi-way data analysis [3, 12, 16, 52, 127, 209, 239]. One of the most popular methods applied to tensor completion is coupled matrix and tensor factorization (CMTF) proposed by Acar et al. [3]. Based on the idea of

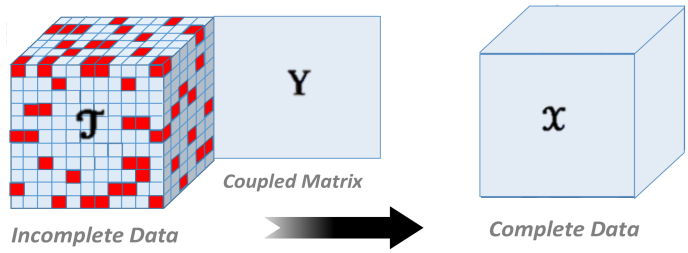

Fig. 4. Illustration of the coupled matrices/tensors factorization for tensor completion [3].

sharing the factorization matrices among matrices and tensors constructed from heterogeneous datasets, they propose an all-at-once algorithm to impute an incomplete tensor coupled with one or more matrices. Mathematically, it is translated to

$$\underset{\mathbf{A}_1,\ldots,\mathbf{A}_N,\mathbf{V}}{\text{minimize}} \ \frac{1}{2}\|\mathcal{P}_{\Omega}(\mathcal{X} - [\![\mathbf{A}_1,\ldots,\mathbf{A}_N]\!])\|_{\mathrm{F}}^2 + \frac{1}{2}\|\mathbf{Y} - \mathbf{A}_n\mathbf{V}^{\top}\|_{\mathrm{F}}^2, \tag{35}$$

where $\mathbf{Y}$ is the coupled auxiliary matrix for the nth mode. An intuitive illustration is depicted in Figure 4. Both the target tensor $\mathcal{X}$ and auxiliary matrix $\mathbf{Y}$ have low-rank structure, and they are assumed to share common latent factors in their coupled mode. Acar et al. [3] also show that though CP decomposition achieves comparable recovery accuracy when the missing ratio is lower than 80%, its recovery error sharply increased when the error rate is further increased. By contrast, CMTF could keep acquiring low recovery accuracy until the error rate is over 90%. Similar to the CP-based methods [16, 239], factor matrices of Tucker decomposition could also be coupled with auxiliary matrices or tensors to benefit the completion tasks [52, 209, 219]. Moreover, recent work by Zhou et al. [241] provides a Riemannian manifold approach to incorporate auxiliary coupled matrices for tensor completion. By adding coupled-matrix regularizers[7] into the vanilla Riemannian tensor completion model described in Equation (27), the model could be optimized with Riemannian Conjugate Gradient descent method. An interesting property of this model is that the coupled-matrix regularizers could be equivalently transformed as the same format as the with-in mode regularizer defined in Equation (32), with the difference being that the Laplacian matrices are substituted by the projection matrices constructed by the coupled auxiliary matrices.

4.3 Other Approaches

Besides the above two approaches, an increasing number of research are conducted recently seeking for new ways to exploit various types of auxiliary information. We list some of them here and give a brief introduction.

4.3.1 Hybrid Models. The hybrid model is one of the most intuitive approaches based on the aforementioned two approaches. For example, Ge et al. [60] propose a tensor-based recommendation framework TAPER to tackle the personalized expert recommendation problem. In addition to the similarity matrices utilized in its groundwork model AirCP, TAPER also incorporates the cross-mode features as coupled matrices and examines the effectiveness of different combinations of the heterogeneous contextual information.

4.3.2 Coupled Trace Norm. This method is a recent model by Wimalawarne et al. [210]. Different from couple factorization models, the authors defined three different coupled trace norms

[7]Similar to the one defined in Equation (35).

by calculating the trace norm of coupled matrices concatenated with the latent factorized matrices or the matricization of the targeted tensor. For example, for a third-order tensor, the coupled overlapped trace norm for its first mode with the corresponding coupled matrix is defined as

$$\|\mathfrak{X}, \mathbf{Y}\|^1 := \|[\mathbf{X}_{(1)}; \mathbf{Y}]\|_* + \sum_{i=2}^{3} \|\mathbf{X}_{(i)}\|_*, \tag{36}$$

where $[\mathbf{X}_{(1)}; \mathbf{Y}]$ denotes the concatenation of the mode-1 unfolding of the target tensor and the corresponding coupled matrix.

4.3.3 Other Constraints. Since all of the methods above can be treated as adding different types of regularization constraints based on the auxiliary information, in the end, we describe several other constraints leveraging different side information here for better coverage of related advances. In [225], the authors proposed a smooth constrained completion method, which imposes linear constraints on individual components in the CP factorized matrices. It could be treated as an extension to the symmetric similarity matrix in Equation (32) into non-symmetric equations using prior knowledge. The method also considers other two types of smoothness constraints, i.e., total variation and quadratic variation, and supports automatic rank search by gradually increasing the number of components during the optimization until the optimal rank is achieved. Rather than putting the concerns on single components, Vervliet et al. consider another linear constraint in [205]. They assume each factorized matrix ($\mathbf{A}_n$) could be further decomposed into a known matrix $\mathbf{B}_n$ and an unknown coefficient matrix $\mathbf{C}_n$, i.e., $\mathbf{A}_n = \mathbf{B}_n\mathbf{C}_n$. It could also be treated as a "reversed" version of the coupled matrices/tensors factorization (since we are not decomposing the auxiliary matrices $\mathbf{B}_n$ but decomposing the latent matrices $\mathbf{A}_n$) and could be optimized with a novel nonlinear least squares algorithm faster and more accurately comparing with the traditional CP decomposition methods when few entries are observed or when missing entries are structured.

5 SCALABLE TENSOR COMPLETION

With the rapid growth in the volume of datasets with the high-order tensor structure, it is very challenging for tensor completion approaches to handle large-scale datasets with billions of elements in each mode due to the high computational costs and space. In this section, we introduce existing efforts in tackling the scalable tensor completion problem and elaborate these methods with the key challenge they focus on. Since almost all of the current advances are concentrating on CP- and Tucker-based approaches, we split them into a separate section and emphasize them first, and then summarize other related methods with a unified view. It should be noted that although some methods (especially decomposition based methods) neither explicitly conduct the completion tasks nor claim that they are focusing on the completion problem, as they leverage the sparsity property to tackle the scalable challenges, we still include some popular ones here for the sake of completeness.

5.1 Scalable CP and Tucker-Based Methods

As described in Section 3, decomposition based methods are one of the most widely used types of methods in tensor completion problem. Due to increasingly amount of data volume, they often face with several challenges such as the intermediate data explosion problem [91] where the amount of intermediate data of an operation exceeds the capacity of a single machine or even a cluster, and the data decentralization problem, where the multi-linear property, as well as various types of the regularization terms, affect the scalability and parallelism of the optimization. Notably, these challenges also appear in the scalable tensor decomposition problem, but we mainly target on the

methods, which are proposed for the tensor completion problem or sparse tensor decomposition[8] for the following three reasons: (1) Our main focus is to impute the missing entries rather than distilling the tensors into possibly interpretable soft clusterings. (2) Different from the dense tensors, which are usually assumed in the tensor decomposition problem, the missing ratios of the incomplete tensor or the sparsity property may highly affect the time and space complexities of the tensor completion algorithms besides the size of the tensors. (3) Although decomposing sparse tensors does not require the process of data imputation, the sparsity property it assumes causes it to share high similarities and challenges with the tensor completion problem. Since the methods in both problems focus on the optimization schemes to address the scalability issues, most of these optimization algorithms could be interchangeably used thus benefit both tasks.

5.1.1 Intermediate Data Explosion Problem. Prior research studies have put their main focus on the intermediate data explosion problem occurring in the operations for updating factor matrices in ALS algorithm [9, 16, 35, 88, 172, 227]. With the increase of the size of input data, the size of the intermediate data in the operations could become huge (with hundreds of millions of non-zero elements [91]).

ALS [180] is one of the most popular algorithms to decompose either the fully observed tensor data or the partially observed tensor data into its CP format by conditionally updating each factor matrix given others. However, previous ALS approaches usually have scalability issues. It is inevitable to materialize matrices, calculated by Khatri–Rao Product, that are very large in sizes and cannot fit in memory. For instance, we review the definition of Khatri–Rao product of $\mathbf{A} \in \mathbb{R}^{I\times R}$ and $\mathbf{B} \in \mathbb{R}^{J\times R}$ with the same number of columns denoted as R:

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{A}(:,1) \otimes \mathbf{B}(:,1), \ldots, \mathbf{A}(:,R) \otimes \mathbf{B}(:,R)], \tag{37}$$

where the size of $\mathbf{A} \odot \mathbf{B}$ is $IJ \times R$. Apparently, if $I = 1$ million, $J = 1$ million and $R = 10$, the Khatri–Rao Product $\mathbf{A} \odot \mathbf{B}$ is of size 1 trillion $\times$ 10, which is normally very large and dense, and obviously cannot fit in memory. Therefore, how to solve this intermediate data explosion problem plays an important role in scaling tensor decomposition algorithms. Since the sparsity is a common property of tensors in the completion task, meaning that most elements in the tensor are zeros (or more generally, unobserved), a naive solution to the intermediate data explosion problem is to adopt the coordinate format to only store non-zero elements for tensor completion and further avoid the materialization of huge, unnecessary intermediate Khatri–Rao products [9].

A recent ALS approach for partially observed matrices [242] is extensible and parallelizable for the CP-based tensor completion, which updates each latent matrix $\mathbf{A}_n$ row by row as follows:

$$\mathbf{A}_n(i,:) = (\mathbf{B}_i^{(n)} + \lambda \boldsymbol{I}_R)^{-1}\mathbf{C}_i^{(n)}, \tag{38}$$

where $\mathbf{B}_i^{(n)} \in \mathbb{R}^{R\times R}$ is a matrix whose entries are as follows:

$$\mathbf{B}_i^{(n)}(j_1,j_2) = \sum_{(i_1,i_2,\ldots,i_N)\in\Omega_i^{(n)}} \left(\prod_{k\neq n} \mathbf{A}_k(i,j_1) \prod_{k\neq n} \mathbf{A}_k(i,j_2) \right). \tag{39}$$

$\mathbf{C}_i^{(n)} \in \mathbb{R}^{R\times 1}$ is a vector whose entries are as follows:

$$\mathbf{C}_i^{(n)}(j,1) = \sum_{(i_1,i_2,\ldots,i_N)\in\Omega_i^{(n)}} \left(x_{i_1,i_2,\ldots,i_N} \prod_{k\neq n} \mathbf{A}_k(i,j) \right). \tag{40}$$

[8]The sparse tensor decomposition here is different from decomposing a tensor into sparse factors. While former one assumes the original tensor to be sparse, the latter one usually requires the decomposed latent factors to be sparse.

$\boldsymbol{I}_R$ denotes an identity matrix with the size of $R \times R$, $\Omega_{\mathbf{i}}^{(\mathbf{n})}$ is the set of indices of observed entries in $\mathfrak{X}$ whose nth mode's index is i. Therefore, updating each latent matrix can be parallelized by distributing the rows of the latent matrix across machines and performing the update simultaneously without affecting the correctness of ALS. However, since this ALS approach requires that each machine should exchange its latent matrix with others, its communication and memory costs are relatively high with the complexity of $O(R \sum_{n=1}^{N} I_n)$ per iteration. Hence, each machine has to load all the fixed matrices into its memory as a scalability bottleneck noted in [63, 227].

Similarly, Zinkevich et al. [243] extend stochastic gradient descent (SGD) to a distributed version as parallelized stochastic gradient descent (PSGD). Concretely, PSGD randomly splits the observed tensor data into M machines and runs SGD on each machine independently. The updated parameters (latent matrices) are averaged after each iteration. Each element $A_n(i, j)$ in the latent matrix is updated by the following rule:

$$\mathbf{A}_n(i,j) \leftarrow \mathbf{A}_n(i,j) - 2\eta \left(\frac{\lambda \mathbf{A}_n(i,j)}{\|\Omega_i^{(n)}\|} - r_{i_1, i_2, \ldots, i_N} \left(\prod_{k \neq n} \mathbf{A}_k(i,j) \right) \right), \tag{41}$$

where $r_{i_1, i_2, \ldots, i_N} = x_{i_1, i_2, \ldots, i_N} - \sum_{s=1}^{R} \prod_{k \neq n} \mathbf{A}_k(i, s)$. Apparently, in PSGD, the memory requirements cannot be distributed since all latent parameters $O(R \sum_{n=1}^{N} I_n)$ need to be exchanged by each machine after each iteration. Also, PSGD has a slow convergence rate.

Another optional way to avoid intermediate data explosion in tensor completion is to reduce the operations that may be too large to fit in memory like the tensor-matrix multiplication. For instance, Kolda and Sun [107] work on the Tucker decomposition method for sparse data and solve the intermediate data explosion problem on the target:

$$\mathcal{Y}_n \rightarrow \mathfrak{X} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \times_{n-1} \mathbf{A}_{n-1} \times_{n+1} \mathbf{A}_{n+1} \cdots \times_N \mathbf{A}_N, \tag{42}$$

where $\mathcal{Y}_n$ is the multiplication of the tensor-matrix multiplication for the nth dimension, i.e., computing the product of a sparse tensor times a series of dense matrices, which may be too large to fit in memory in the process. In order to maximize the computational speed while optimally utilizing the limited memory, the computation is handled in a *piecemeal fashion* by adaptively selecting the order of operations. Concretely, the tensor data is stored with the coordinate format as proposed in [9]. Such tensor–matrix multiplication can be calculated one slice or fiber at a time instead of directly multiplying a matrix since the tensor–vector multiplication will produce the results with much smaller size that can easily fit in the limited memory.

Though methods stated above are capable of efficiently decompose and impute an incomplete tensor when the tensor data is very large and sparse, these methods still operate in the limited memory of a single machine. Hence, many researchers utilize the parallel computing system like MPI (Message Passing Interface) [35] and the distributed computing system such as MAPREDUCE [16, 88, 172] to effectively and efficiently perform tensor decomposition and completion for large-scale datasets. In order to address the intermediate data explosion problem in a distributed/parallel fashion, the main ideas include carefully selecting the order of computations in the update of latent matrices for minimizing flops (floating point operations), exploiting the sparsity of the tensor data to avoid the intermediate data explosion, and utilizing the distributed cache multiplication to further minimize the intermediate data by exploiting the skewness in matrix multiplications in a distributed/parallel computing system.

One of the most popular scalable distributed algorithm for PARAFAC sparse tensor decomposition is GigaTensor proposed by Kang et al. [91], which solves the intermediate data explosion problem by reordering computation and exploiting the sparsity of the tensor data in MAPREDUCE. Specifically, GigaTensor targets the intermediate data explosion problem occurred on updating

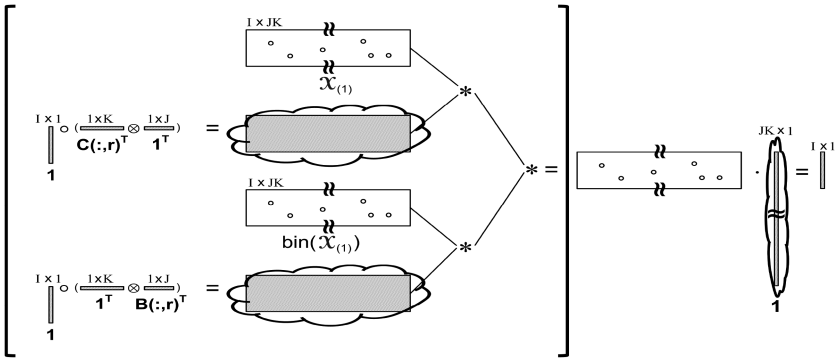

Fig. 5. Illustration of the key idea of GigaTensor to avoid the intermediate data explosion problem [91].

latent matrices based upon ALS as follows (taking a third-order tensor as an example):

$$\mathbf{A} \leftarrow \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})(\mathbf{C}^\top \mathbf{C} * \mathbf{B}^\top \mathbf{B})^\dagger, \tag{43}$$

where $\mathbf{X}_{(1)}$ is the unfolded matrix of the tensor $\mathcal{X}$ on the first mode, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are latent factorized matrices for three dimensions , respectively, and † denotes the pseudo inverse. Inspired by the concept of divide-conquer, the main idea (Figure 5) is to take advantage of the sparsity in tensor data to decouple the two terms in the Khatri–Rao product $\mathbf{C} \odot \mathbf{B}$ and sequentially performing algebraic operations with the unfolded tensor $\mathbf{X}_{(1)}$ instead of directly computing the Khatri–Rao product. Concretely, computing $\mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$ can be equally treated as calculating $(\mathbf{N}_1 * \mathbf{N}_2) \cdot \mathbf{1}_{JK}$ where $\mathbf{N}_1 = \mathbf{X}_{(1)} * (\mathbf{1}_I \circ (\mathbf{C}(:,r)^\top \otimes \mathbf{1}_J^\top))$, $\mathbf{N}_2 = \text{bin}(\mathbf{X}_{(1)}) * (\mathbf{1}_K^\top \otimes \mathbf{1}_I \circ (\mathbf{B}(:,r)^\top))$, bin() function converts any non-zero value into one, preserving sparsity, and $\mathbf{1}_{JK}$ is an all-1 vector of size JK. By this way, the size of intermediate data will reduce from $O(JKR + m)$ to $O(\max(J + m, K + m))$, where m is the number of non-zero elements. All these operations are implemented in MAPREDUCE by designing mapper and reducer. As a similar fashion, Jeon et al. [88] propose HaTen2 that improves on GigaTensor by unifying Tucker and PARAFAC decompositions for sparse tensors into a general framework. Both GigaTensor and HaTen2 can be easily extended to solve the completion problem by adding an indication tensor into the objective function.

Recently, DFacTo, an efficient, scalable and distributed algorithm proposed by Choi et al. [35], also addresses the intermediate data explosion problem by particularly focusing on scaling $\mathcal{F} = \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$. Concretely, DFacTo computes $\mathcal{F}$ in a column-wise manner as follows:

$$\begin{aligned} \mathcal{H} &= \mathbf{X}_{(2)}^\top \mathbf{B}(:,r) \\ \mathcal{F}(:,r) &= unvec_{(I,K)}(\mathcal{H})\mathbf{C}(:,r), \end{aligned} \tag{44}$$

where $\text{unvec}_{(I,K)}(\cdot)$ is the operator of reshaping a IK dimensional vector into a $I \times K$ matrix. All these operations in DFacTo are very efficient assuming that $\mathcal{X}$ is a sparse tensor represented by the Compressed Sparse Row (CSR) format. DFacTo is implemented by applying a master-slave architecture of MPI (Message Passing Interface). The communication cost of DFacTo is relatively high since the master has to transmits all latent matrices to the slaves at every iteration.

5.1.2 Decentralization of the Tensor Data Problem. From a different perspective, some researchers focus on the way of decentralizing the original observed tensor data based on divide-and-conquer rather than the intermediate products. Though it could implicitly solve the intermediate data explosion problem, the decentralization itself could become a new headache since the multi-linear properties may prevent the decentralization of the tensor data to be easily done.

FLEXIFACT [16] is one of recent approaches targeting on this problem. It splits the observed tensor data into M^N blocks in which each M dis-joint blocks are treated as a stratum without sharing common fibers. *FLEXIFACT* processes a stratum at a time by decentralizing M dis-joint blocks into M machines and processes them independently and simultaneously. In each machine, *FLEXIFACT* applies SGD to solve the corresponding block. By this way, the updated parameters in this machine are disjoint with the ones updated by others. Hence, the memory requirements of *FLEXIFACT* are distributed among M machines. Nevertheless, its communication cost is high since after processing a stratum, *FLEXIFACT* needs to aggregate all parameters updated by each machine into the master machine that updates them and decentralizes them for the next iteration.

To overcome the problem of high communication cost in algorithms like *FLEXIFACT*, a scalable tensor factorization CDTF (Coordinate Descent for Tensor Factorization) is developed by Shin et al. [172], which extends CCD++ [227] to higher orders based on coordinate descent. Concretely, each parameter can be updated by the following rule:

$$\mathbf{A}_n(i,j) = \frac{\sum_{(i_1,i_2,\ldots,i_N)\in\Omega_i^{(n)}} (\hat{r}_{i_1,i_2,\ldots,i_N} \prod_{k\neq n} \mathbf{A}_k(i,j))}{\lambda + \sum_{(i_1,i_2,\ldots,i_N)\in\Omega_i^{(n)}} \prod_{k\neq n} \mathbf{A}_k(i,j)}, \tag{45}$$

where $\hat{r}_{i_1,i_2,\ldots,i_N}$ is the rank-one factorization of tensor $\hat{\mathfrak{R}}$ and can be updated with the complexity of $O(\|\Omega_i^{(n)}\|N)$ by the following rule:

$$\hat{r}_{i_1,i_2,\ldots,i_N} \leftarrow r_{i_1,i_2,\ldots,i_N} + \prod_{n=1}^{N} \mathbf{A}_n(i_n,k), \tag{46}$$

where $r_{i_1,i_2,\ldots,i_N} = x_{i_1,i_2,\ldots,i_N} - \sum_{s=1}^{R} \prod_{k\neq n} \mathbf{A}_k(i,s)$ that can be updated with the complexity of $O(\|\Omega_i^{(n)}\|N)$ by the following rule:

$$r_{i_1,i_2,\ldots,i_N} \leftarrow r_{i_1,i_2,\ldots,i_N} + (\mathbf{A}_n^{old}(i_n,k) - \mathbf{A}_n(i_n,k)) \prod_{l\neq n} \mathbf{A}_l(i_l,k). \tag{47}$$

The update sequence of parameters in CDTF adopts the column-wise order by updating the kth column of a latent matrix and then moving on to the kth column of the next latent matrix. After updating the kth columns of all latent matrices, the $(k+1)$th columns of latent matrices get started to be updated. Extensively, Shin et al. propose another scalable tensor factorization algorithm based on subset alternating least square (SALS) that updates each $C(1 \leq C \leq R)$ columns of latent matrices row by row. On the contrary, CDTF updates each column entry by entry; ALS updates all R columns row by row. Both CDTF and SALS are implemented in MapReduce [172]. The differences between the decentralization strategies used in *FLEXIFACT* and CDTF/SALS are compared in Figure 6.

Besides the aforementioned approaches, there are also other scalable tensor completion or sparse tensor decomposition approaches focusing on either CP format [62, 93, 98, 122, 155, 160, 182] or Tucker format [133, 150, 183]. As the end of this subsection, we compare several the state-of-the-art distributed/parallel scalable tensor completion/sparse tensor decomposition algorithms in Table 4, from the aspects of scalability, platform, and complexity.

5.2 Other Approaches

Although CP and Tucker-based methods are the main focus in existing literatures, more and more researches are trying to explore other methods to tackle the scalability issue in tensor completion problem. In the context of CP rank, Cheng et al. [34] put their focus on approximating the tensor trace norms to achieve computational efficiency. By constructing a polytopal approximation of the tensor dual spectral norm, the completion problem regularized with tensor spectral norm could be

Table 4. Comparison between Distributed/Parallel Scalable Tensor Completion/Decomposition Algorithms

	Parallel ALS (generalized from [242])	PGSD [243]	DFacTo [35]	FLEXIFACT [16]	GigaTensor [91]	HaTen2 [88]	CDTF [172]
Scalability							
Dimension	✓	✓	✓		✓	✓	✓
No. of Parameters			✓	✓			✓
No. of Machines	✓		✓		✓	✓	✓
Rank			✓	✓	✓	✓	✓
Platform							
MPI			✓				
MapReduce				✓	✓	✓	✓
Parallel	✓	✓					
Complexity							
Time	$O((\lvert\Omega\rvert(N+K)+IK^2)NK/M)$	$O(\lvert\Omega\rvert NK/M)$	$O((\lvert\Omega\rvert+nnz(\mathbf{X}_{(2)}))K/M)$	$O(\lvert\Omega\rvert NK/M)$	$O(IK^2)$	$O(\lvert\Omega\rvert NK/M)$	$O(\lvert\Omega\rvert N^2K/M)$
Space	$O(NIK)$	$O(NIK)$	$O(nnz(\mathbf{X}_{(2)}))$	$O(NIK/M)$	$O(MK^2)$	$O(\lvert\Omega\rvert K)$	$O(NIK)$
Communication	$O(NIK)$	$O(NIK)$	$O(nnz(\mathbf{X}_{(2)}))$	$O(M^{N-2}NIK)$	$O(MK^2)$	$O(\lvert\Omega\rvert K)$	$O(NI)$

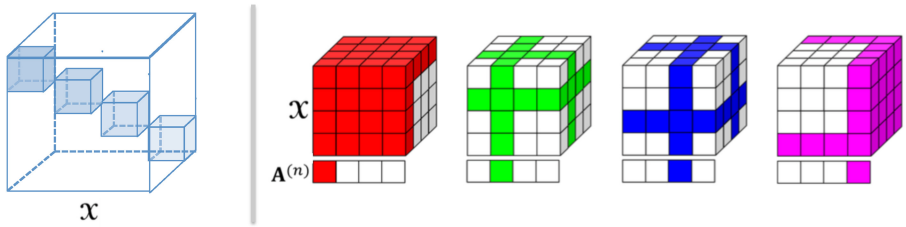

Fig. 6. Comparison of the decentralization manner between *FLEXIFACT* (left) [16] and CDTF/SALS (right) [172] on a third-order tensor with four machines. For *FLEXIFACT*, we only show four of the 16 blocks, which are separated in four machines in one stratum. For CDTF/SALS, different colored blocks, which are decentralized in different machines, represent the data for optimizing different latent factors. One block could be delivered to different machines.

solved by the generalized conditional gradient algorithm [73]. Another recent approach focusing on scalability issue of the trace norm regularizer in the LRTC problem is described in [71]. The authors utilize the Frank–Wolfe (FW) algorithm, which has been successfully applied for matrix completion problem with nuclear norm regularization and propose a time and space-efficient algorithm for the sparse LRTC problem. Since the scalability issue appeared in multi-way datasets is not only caused by the size of each dimension but also caused by the number dimensions, some researchers focus on the developing scalable algorithm to address the high-order challenge in the scalability tensor completion problem. A recent model by Imaizumi et al. [84] leverages the TT decomposition to address the tensor completion problem for higher-order tensors. To achieve the theoretical veracity from the statistical perspective, they first propose a convex relaxation of the TT decomposition and derive the completion error bound. By developing a randomized ALS optimization approach, a scalable completion algorithm is then proposed to achieve both time efficiency and space efficiency. Equipped with the view of leveraging TT decomposition to overcome the problem of "the curse of dimensionality," a recent work by Yuan et al. [235] introduces two gradient-based algorithms to address the high-dimension tensor completion problem. They also proposed a new paradigm called tensor ring decomposition for scalable tensor decomposition and completion [229].

6 DYNAMIC TENSOR COMPLETION

Beyond traditional static or batch setting, with the increasing amount of high-velocity streaming data, more concerns have been addressed on dynamic tensor analysis [83, 148, 189, 190]. As most of the existing surveys are lack of detailed introduction of dynamic tensor analysis, we first give a brief investigation of some existing works on dynamic tensor analysis especially focusing on dynamic tensor factorization, and then introduce several dynamic tensor completion algorithms.

As for CP decomposition, Nion and Sidiropoulos [148] introduce two adaptive PARAFAC algorithms adopting the recursive least square (RLS) and simultaneous diagonalization tracking approaches to address the online third-order tensor factorization problem. Recent work by Zhou et al. [240] develops an accelerated online CP algorithm that can incrementally track the decompositions of one-mode-change Nth tensors. Phan et al. [158] partition a large-scale tensor into some small grids and develop a grid-based scalable tensor factorization method (gTF), which could also be easily applied to address the streaming tensor factorization problem.

Besides CP decomposition, some Tucker decomposition methods were also proposed. One of the earliest work is conducted by Sun et al. [189, 190]. They propose two dynamic Tucker factorization methods based on the incremental update of covariance matrices for dynamic tensor analysis.

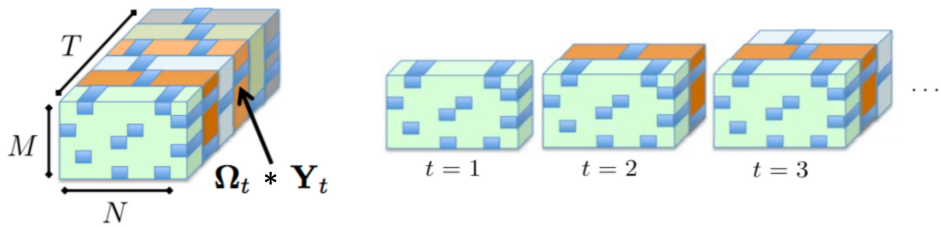

Fig. 7. Batch tensor completion (left) versus steaming tensor completion (right). [138]

Subsequently, Yu et al. [228] raised an accelerated online tensor learning algorithm (ALTO) to conduct streaming Tucker factorization. Several other online Tucker decomposition methods are conducted recently not only targeting on the one-mode-increase pattern [81, 184] but also deriving possible solutions for multi-aspect streaming patterns drawing upon matrix-based online-learning methods such as incremental SVD [136]. Furthermore, a histogram-based approach [54] conducted on Multi-Aspect Streaming Tensor (MAST) analysis could be regarded as the pioneering research on the multi-aspect streaming analysis.

6.1 Streaming Tensor Completion

Although various researches have been conducted on dynamic tensor analysis, there has not been much work focusing on the tensor completion problem with dynamic patterns. Early work of Meng et al. [140] use PARAFAC models to in-filling missing future data and have shown to overcome the challenge of signal monitoring. Matsubara [139] use topic modeling with Gibbs sampling method to get the posterior latent factors of each mode to solve the future traffic forecasting problem, e.g., how many clicks a user may generate tomorrow. However, this approach is more close to being a pure future prediction problem rather than a completion problem. A more proper dynamic completion problem is addressed in a recent approach proposed by Mardani et al. [137, 138]. They focus on streaming tensor completion problem and propose an online Stochastic Gradient Decent algorithm for tensor decomposition and imputation. Figure 7 [137] illustrates the difference between batch tensor completion and streaming tensor completion problem. For a third-order tensor of size $M \times N \times T$, where its third mode is the temporal mode, all of the incomplete slices $\{\Omega_t * \mathbf{Y}_t\}_{t=1,\ldots,T}$ are assumed to be collected in advance in the batch setting while in streaming tensor completion, each slice $\Omega_t * \mathbf{Y}_t$ becomes available until the corresponding time t.

To address this problem, Mardani et al. [138] propose a streaming version of CP decomposition and achieve the completion purpose by adding the separable nuclear-norm regularizations described in Equation (25). Using a three-order tensor as an example, the main objective function is a time-wise exponentially-weighted least-squares loss defined as follows:

$$\mathcal{L} = \frac{1}{2}\sum_{t=1}^{T}\theta^{T-t}\left[\mathbf{\Omega} * (\mathbf{Y}_t - \mathbf{A}\mathrm{diag}(\boldsymbol{\gamma}_t)\mathbf{B}^\top) + \frac{\lambda_t}{\sum_{t=1}^{T}\theta^{T-t}}(\|\mathbf{A}\|_\mathrm{F}^2 + \|\mathbf{B}\|_\mathrm{F}^2) + \lambda_t\|\boldsymbol{\gamma}_t\|^2\right], \tag{48}$$

where $\boldsymbol{\gamma}_t^\top$ represents the tth row of the CP factorized matrix $\mathbf{C} \in \mathbb{R}^{T\times R}$, $0 < \theta \leq 1$ is called the forgetting factor to decay the influence of the past data so that facilitating the subspace tracking in non-stationary environments [138, 185]. $\{\lambda_t\}_{t=1,\ldots,T}$ are the regularization hyperparameters controlling the regularization terms.

The optimization is performed with an online updating scheme, which is a generalization of matrix online PCA algorithm [55], to factorize and impute the tensor "on the fly." The main idea is that when a new data slice $\Omega_t * \mathbf{Y}_t$ arrives at time t, we first fix the factors of the non-incremental modes (i.e., $\mathbf{A}$ and $\mathbf{B}$) and calculate the new temporal factor γ_t with an close-formed solution, which

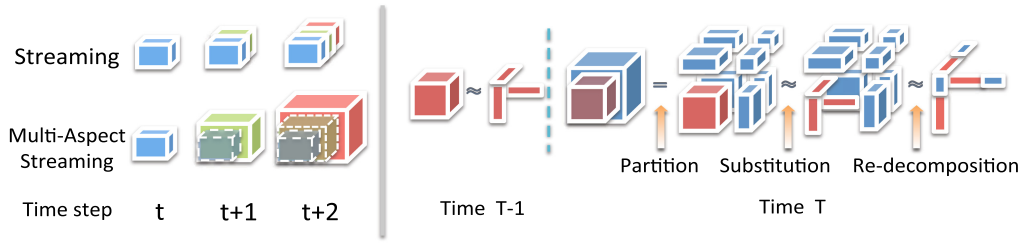

Fig. 8. Streaming tensor completion versus multi-aspect steaming tensor completion(left), multi-aspect streaming tensor decomposition (right). [186]

is derived based on the ridge-regression problem of $\boldsymbol{\gamma}_t$ using Equation (48). And then update the non-incremental factors $\mathbf{A}$ and $\mathbf{B}$ with the SGD. Similar with online PCA, the latent assumption of these online SGD methods should be a rarely and smoothly changed streaming tensor subspace, such as surveillance video streaming, which allows them to pursue the real subspace iteratively. In some real-world situation, the processing speed could be faster than data acquiring speed. Kasai [94] proposes another online CP decomposition algorithm OLSTEC while conducting data imputation based on RLSs algorithm. They consider the situations where the data processing speed is much faster than data acquiring speed and propose a more rapid convergence algorithm for tracking dramatically changed subspace. By storing auxiliary matrices of non-increasing modes, these size fixed modes could be efficiently updated by considering a parallel set of smaller least square problems. The incremental subspace of the time mode is obtained with a one-time pursuit during each timestamp. Although sacrificing time complexity to some extent (increased from $O(|\Omega_t|R^2)$ to $O(|\Omega_t|R^2 + (I_1 + I_2)R^3)$ for a third-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$), OLSTEC is able to converge faster for tracking dramatically changed low-rank subspace comparing to SGD based method.

6.2 Multi-Aspect Streaming Tensor Completion

Recently, several researchers focus on a more general situation in tensor related problems, which is called multi-aspect streaming [54, 147, 186]. As shown in the left figure in Figure 8, different from the streaming tensor setting, in the multi-aspect streaming setting, a tensor could increase on any of its modes, which imposes greater challenge in coordinating multi-dimensional dynamics, handling the higher time and space complexity as well as imputing the incremental missing entries, and so on [186].

Song et al. [186] propose the first MAST completion method, which could handle the all-mode-change tensor dynamics while imputing the missing entries. The core model is a CP-based dynamic tensor decomposition model consisting of three operations, i.e., partition, substitution and re-decomposition. The key idea of its updating scheme at each pair of consecutive timestamps is shown in Figure 8. Suppose we obtain a small tensor along with its CP decomposition at time $T-1$, and it increases into a larger tensor at time T. Since the old tensor is a sub-tensor of the new one, we could partition the large tensor into smaller blocks based on the size of the old tensor and approximately substitute the old tensor with its CP decomposition. This substitution would highly improve the optimization speed especially when the ratio $\mathbf{t} = \frac{\text{size (old tensor)}}{\text{size (new tensor)}}$ is large. Similar with Equation (48), a forgetting factor is also used during the optimization to alleviate the error induced by the substitution step. The authors also describe a special case, which is called temporal multi-aspect streaming, and provide a corresponding algorithm to further reduce the completion error based on this special pattern.

In addition to CP-based models, Nimishakavi et al. [147] propose a Tucker-based model called SIITA that could handle the MAST completion while incorporating auxiliary information. It is an online inductive framework extended from the matrix completion framework proposed by [145, 173] and achieves superior performance in both batch and streaming settings. As the dynamic tensor completion are still in its early stage, there are still vast opportunities and challenges that are valuable and attracted to delve and pursue.

7 APPLICATIONS

The tensor completion problem arises across many application scenarios. In this section, we address several domains including social sciences, computer vision and signal processing, healthcare and bioinformatics, and others. For each application, the following three questions are mainly discussed: (1) what the concrete problem is; (2) how it is formulated as a tensor completion problem; and (3) what kind of methods are applied.

7.1 Social Computing

Social computing concerns with applying computation methods to explore individual activities and relationships within societies. It contains various interdisciplinary branches and includes multiple problems that could be modeled as tensor completion problems.

7.1.1 Link Prediction. Link prediction aims at predicting missing edges in a graph or the probabilities of future node connections. Traditionally, this problem is treated as a matrix completion problem aiming at imputing the missing or observed links based on the existing node connections. However, dynamic interactions over time [1], as well as multiple types of node interactions [132], introduce additional dimensions, leading the tensor completion algorithms to be naturally used to handle the multi-dimensional dataset.

There are two commonly used approaches, which extend the traditional graph topology matrix into tensors based on different data structures. (1) By adding the temporal mode into a 2-D bipartite graph, it natural derives a binary completion problem of a three-order tensor. Acar et al. [1] exploit CP decomposition with a heuristic definition of CP similarity score for future link prediction. Their subsequent work in [47] gives another definition of CP score using the Holt-Winters forecasting method [32] and proves the effectiveness of tensor-based methods in forecasting the varying periodic patterns among links. (2) Different type of interactions among nodes could also be defined as the third mode [97, 132]. Based on this construction, Kashima et al. [97] first utilize auxiliary similarity information of nodes for tensor link prediction. Ermiş et al. [52] address the link prediction problem by jointly analyzing multiple sources via coupled factorization. They apply the proposed generalized framework on two different scale real-world datasets UCLAF (user-location-activity) and Digg (user-story-comment) and compare the affect of different loss function and factorization models in practice. See also Ermiş et al. [51].

7.1.2 Recommender Systems. Recommender systems aim at predicting the users' preferences from partial knowledge for personalized item recommendation. It naturally derives a completion problem if we treat unobserved "ratings" of users to items as missing entries of data matrices or tensors. Due to the ternary relational nature of data in various situations (e.g., Social Tagging Systems (STSs)), many recommendation algorithms, which are originally designed to operate on matrices, cannot be easily applied on these multi-dimensional datasets [191]. Early works by Rendle et al. [166–168] illustrate the power of tensor-based methods for personalized tag recommendation. They proposed three different methods to recommend new items a user would like to tag. Their first approach focuses on learning the best ranking by minimizing Area Under Curve (AUC) score rather than an element-wise error. This method is shown to outperform both

tensor-based method HOSVD and ranking methods such as PageRank in quality and runtime. Their second approach is a special case of CP decomposition. This approach separates the trinity interaction among user-item-tag into the summation of pairwise interactions (user-tag, user-item, item-tag) and is proved to achieve higher prediction accuracy faster. Finally, they propose a mixture approach called factorization machines (FMs) by combining factorization models with Support Vector Machines (SVM). Benefited from factorization models, the model could nest interactions among variables to overcome the sparsity scenarios where SVMs fail. Furthermore, FMs are general predictors, which could mimic both matrix and tensor-based factorization models such as SVD++ [108] and their second approach PITF. Other attempts such as HOSVD-based dimensionality reduction [156, 192] have also been considered for personalized tag recommendations or tag-involved recommender systems.

Another way of transforming a user-item matrix into a tensor object is to incorporate temporal information. One of the early approaches is a Bayesian Probabilistic Tensor Factorization model proposed by Xiong et al. [214]. They extend the matrix-based probabilistic approach as well as the MCMC optimization method proposed in [171] and demonstrates the advantage of a high-order temporal model over matrix-based static models. A more general case of this model is considered by Karatzoglou et al. [92]. They take advantage of different types of context information by considering them as additional dimensions besides traditional 2-D user-item matrix. They exploit Tucker factorization optimized by the SGD method in which missing entries are skipped during the decomposition process and reconstructed in the end. This multiverse recommendation approach outperforms traditional matrix-based systems by a wider margin that demonstrates the superiority of multi-dimensional analysis when more contextual information is available.

In some situations, side information of users, items as well as features are available [11, 60, 179, 239], which motivates us to combine auxiliary information for collective learning. Borrowing the idea from matrix case [179], Zheng et al. [239] target the problem of location-based activity recommendation system. They couple four kinds of side information including location features, GPS visiting information (user-location) as well as two similarity relationships of users and activities, respectively. These auxiliary regularizers effectively alleviate the sparsity problem while mining knowledge from GPS trajectory data for mobile recommendations on both locations and activities.

Recently, Ge et al. [60] propose a model called "Taper" which is a contextual tensor-based approach tailored towards the personalized expert recommendation. They work on recommending personalized topics produced by experts (high-quality content producers) in social media in which dataset is constructed as a third-order tensor (user-expert-topic). Through modeling the homogeneous similarities between homogeneous entities (e.g., users and users) and heterogeneous similarities between pair-wise entities (e.g., users and experts), the proposed framework can achieve high-quality recommendation measured by precision and recall.

Besides methods mentioned above, other models (e.g., Parafac2 model [153]) or optimization techniques (e.g., Riemannian optimization [95]) are also applied in recommender systems. In general, most of the applications on recommender systems are focused on decomposition-based approaches. This is because: (1) Factorization models are easy to implement and effective in learning latent factors among users or items by incorporating interactions among different types of modes. (2) They have advantages in dealing with sparsity or cold start problem which are usually encountered in recommender systems as described in [166].

7.1.3 Urban Computing. Urban computing deals with the human behaviors and mobilities with the help of computing technology to benefit living quality in urban areas. "Traffic and mobility" is one of the most significant topics in this field, which is correlated with tensor completion problem. For example, in intelligent transportation systems, outlier data may be generated due to the

malfunctions in collection procedure and record systems. Some researchers connect this problem with tensor completion problem with a variant setting, which they assume the positions of missing parts are not known beforehand [196]. In another word, partial existed entries might entirely be some noise data (outliers), and their actual values are still unknown [196]. This corrupted data recovery problem is addressed by robust tensor recovery framework [196], which is proved to outperform a traditional trace-norm approach proposed by Gandy et al. [59]. Wang et al. [209] focus on the problem of estimating the travel time of paths in a city. They construct a third-order tensor (road segments × drivers × time slots) based on the real-time GPS trajectories and the road network data. To address the high sparsity in this tensor, the authors build another denser tensor based on the historical trajectories and two auxiliary matrices storing the geographic similarity between different road segments and the correlation among the various time slots. A Tucker-based Coupled Matrix Tensor Factorization algorithm is used for jointly decomposition, recovery and estimation. Beyond the batch setting, Tan et al. [197] propose a Windows-Based Tensor Completion algorithm to tackle the short-term traffic prediction problem. They form the problem as a third-order tensor (week×day×point) to leverage the strong similarity between the same day in different weeks. The algorithm handles the time dependency through a sliding window. Each sliding window consists of a subset of matrix slices in the matrix stream, which is decomposed and imputed with Tucker-based completion methods.

7.1.4 Information Diffusion. Information diffusion refers to a process by which a piece of information or knowledge is spread initiated by a sender and reaches receivers through medium interactions [232]. Ge et al. [61] linked the problem of modeling and predicting the spatiotemporal dynamics of online memes with tensor completion. The meme data is constructed as a third-order tensor (meme × locations × time). To uncover the full (underlying) distribution, they incorporate the latent auxiliary relationships among locations, memes, and times into a novel recovery framework. Experiments on the real-world twitter hashtag dataset with three kinds of data missing scenarios (random missing, missing entire memes at specific locations, missing entire locations) validate the effectiveness of their tensor-based framework and the auxiliary spatiotemporal information.

7.1.5 Computer Network. Network traffic matrix records the amount of information exchanged between the source and destination pairs such as computers and routers. A third-tensor object could be constructed since the matrix evolves with time. Acar et al. [2] address the completion problem in computer network traffic where missing data arises due to the high expense of the data collection process. The dataset is formed as a third-order tensor denoted as source routers × destination routers × time, where entries indicate the amount of traffic flow sent from sources to destinations during specific time intervals. Their proposed method-CP Weighted OPTimization-enjoys both recovery accuracy and scalability.

There are also various other types of applications in social computing domain, which are correlated with tensor completion problems or leverage related approaches such as social-aware Image tag refinement [198], truth discovery [212], and so on. It is reasonable to believe that tensor completion problem would affects and promotes more and more applications in this area.

7.2 Healthcare, Bioinformatics, and Medical Applications

Medical images such as Magnetic Resonance Imaging (MRI) or Computed tomography (CT) scans are one of the most powerful tools for medical diagnosis. Due to the fast acquisition process, these image datasets are often incomplete [15, 59, 222]. Early applications by Gandy et al. [59] were performed on three medical MRI imaging (KNIX: MRI scans of a human knee, INCISIX: MRI dental scans, BRAINIX: MRI brain scans). Bazerque [15] apply a probability PARAFAC approach on

corrupted brain MRI images. By incorporating prior information under the Bayesian framework, their method is able to enhance the smoothing and prediction capabilities. Similarly, Liu et al. [134] also use Brain MRI dataset for testing their completion method, which is a combination of trace-norm based and decomposition-based approach. Furthermore, the dynamic algorithm proposed by Mardani et al. [138] has also been applied to streaming cardiac MRI data for fast online subspace tracking.

Besides the applications on medical images, several other types of datasets and analytics are also conducted in the realm of bioinformatics and healthcare. Dauwels et al. [41] apply CP decomposition to handle missing data in medical questionnaires. Bazerque et al. [15] perform experiments on RNA sequencing data which represents the gene expression levels in yeast. Wang et al. [208] focus on the problem of Computational phenotyping. By incorporating two types of knowledge-guided constraints-guidance constraint and pairwise constraint-into tensor completion process, the proposed framework "Rubik" can convert heterogeneous electronic health records (EHRs) into meaningful concepts for clinical analysis while proceeding completion and denoising tasks simultaneously. Finally, in an interdisciplinary field related to brain analysis and signal processing, Acar et al. [2] use multi-channel EEG (electroencephalogram) signals where missing data is encountered due to disconnections of electrodes and demonstrate the usefulness of scalable CP factorization method in capturing the underlying brain dynamics for incomplete data.

7.3 Chemometrics

Chemometrics is an area of using mathematical and statistical tools to improve the chemical analysis. Appellof and Davidson [7] are credited as the first one who applies tensor factorization in chemometrics [106]. Since then, tensor analysis has become actively researched in this field including low-rank approximation as well as missing data imputation, and so on. A famous dataset for tensor factorization and completion is the semi-realistic amino acid fluorescence data contributed by Bro and Andersson [19]. It consists of five laboratory-made solutions of three amino acids [200]. Tomasi et al. [199] first consider missing data problem using this dataset. They carry out experiments with three different types of missing elements: randomly missing values, randomly missing spectra/vectors, and systematically missing spectra/vectors. In a similar spirit, Narita et al. [144] test two kinds of assumptions of missing patterns (element-wise missing and slice-wise missing) using a different chemical benchmark dataset: flow injection. This dataset is a chemical substances dataset represented as a third-order tensor $substances \times wavelengths \times reaction\ times$. By incorporating auxiliary similarity information of each mode, their approach greatly improves the completion accuracy comparing to existing methods especially when observations are sparse. Other explorations of tensor completion problem in chemometrics can also be found in [36, 180]

7.4 Computer Vision

In the area of computer vision, various problems could be formulated as tensor completion problems, such as image inpainting [59], video decoding [129], compressed sensing [46], and spectral data analysis [178].

7.4.1 Image Completion. Image completion or image inpainting problem has been actively discussed in computer vision field for its practicability in many real-world applications, such as visual surveillance, human–machine interaction, image retrieval, and biometric identification. As images are in nature multi-dimensional dataset (e.g., an RGB image is a third-order tensor with the color channels (red, green, blue) as its third dimension), tensor-based algorithms are well-suited in coordinating these multiple dimensions and leveraging the multi-linear correlations.

Facial image analysis is one of the important research topics for multi-linear image analysis [204]. Signoretto et al. [177] consider the problem of facial image completion using a benchmark

Olivetti face dataset. They called the pure completion problem "hard completion problem." The corresponding soft completion problem is a semi-supervised problem which informs partial label information, i.e., simultaneously complete the images and a label matrix of images. Their spectral regularized framework tackles both pure completion problems and multi-task cases. Earlier works conducted by Geng et al. [64, 65] consider the "hard completion problem." They leverage the HOOI method to model face images for face completion, recognition, and facial age estimation. To address the missing value problem owing to data collection, they apply the EM-like approach to impute the missing entries and demonstrate the extraordinary performance of their multi-linear subspace analysis method in age estimation.

Hyperspectral data completion is also widely discussed in image completion work. An early approach of using hyperspectral images is proposed by Gandy et al. [59]. They use an URBAN hyperspectral dataset in which each image represents a different band of collected light wavelengths with a spatial resolution of 200×200 pixels. Another benchmark dataset is called "Ribeira"[9] hyperspectral image dataset, which was first used in [57] and further utilized by Signoretto et al. [178] for tensor completion. The datasets consist of collections of raw reflectances sensed from contiguous spectral bands. One could apply tensor-based approaches via treating "scene" as the third mode. See also [95, 110, 134] for more completion approaches on this dataset.

Besides these two kinds of images, many other types of images are considered in tensor completion problems, such as building facade image [134, 195], reflectance data [129, 130], and fMRI images [222]. Either of them is natural RGB images or reconstructed datasets with an additional channel mode or scene mode. We leave the discussion of medical image analysis (mainly focus on MRI images) in the section of healthcare and medical application.

7.4.2 Video Completion. Videos can be naturally represented as multi-dimensional arrays by combing the streaming of scenes. Besides the application of video inpainting, video compression could also be treated as a completion problem during the uncompress process. A user can remove unimportant or unwanted pixels from each frame of original videos and recover it using completion tools. The earliest application of tensor completion methods on video datasets may be the trace norm approach proposed by Liu et al. [129]. Mu et al. [143] also apply their completion methods on color videos, which are naturally represented as four-mode tensors (length×width×channels×frames). Kasai [94] focuses on streaming subspace tracking from incomplete data and exploits the RLS method for online CP decomposition. They evaluate the tracking performances using an airport hall surveillance video where frames of the video arrive in a sequential order. Furthermore, to test the performance of the proposed method in tracking dynamic moving background. They reconstruct the input video stream by reordering the cropped incomplete frames and show the rapid adaptivity of their method OLSTEC to the changed background. Other video completion examples could be found in [194, 234].

7.5 Compressed Sensing

Compressed sensing initiated by [29, 45] in 2006 allows the recovery of a sparse signal from a small number of nonadaptive, linear random measurements through efficient convex algorithms such as ℓ_1-minimization and iterative hard thresholding [49]. It does exact recoveries of high-dimensional sparse vectors after the dimensionality reduction. It is one of the most important applications in signal processing and embedded with multitudinous multi-linear datasets. For example, if we treat videos and images as signals, both of them are multi-dimensional data, and their compressions could be treated as the special cases of compressed sensing.

[9] http://personalpages.manchester.ac.uk/staff/d.h.foster/Hyperspectral_images_of_natural_scenes_04.html.

As a further extension of compressed sensing, the Kronecker-based compressed sensing model [46] has been proposed and explored in order to offer a practical implementation of compressed sensing for high-order tensors. Some researchers [23, 24, 123] focus on building generalized tensor compressed sensing algorithms by exploiting the Kronecker structure with various approaches such as ℓ_1-minimization. Sidiropoulos et al. [175] recovers sparse, low-rank tensors from few measurements by leveraging compressed sensing for multi-linear models of tensor data, and shows that the identifiability properties of the tensor CP decomposition model can be used to recover a low-rank tensor from Kronecker measurements (by compressing each mode separately). Concretely, it evolves into the following two steps: fitting a low-rank model in the compressed domain, and performing per-mode decompression. Rauhut [161] reconstructs a low-rank tensor from a relatively small number of measurements by utilizing the framework of HT formats based on a Hierarchical SVD-based approach followed by a similar work [40]. However, it should be noted that this method is hard to scale tensors with large mode sizes as it strongly relies on computing SVDs of large matrices. Friedland et al. [58] exploit a unified framework of compressed sensing for high-order tensors with preserving the intrinsic structure of tensor data, which provides an efficient representation for multi-dimensional with simultaneous acquisition and compression from all tensor modes. Li et al. [125] focus on incorporating structured sparsity into tensor representation to compress/recover the multi-dimensional signals with the varying non-stationary statistics, and inheriting the merit from tensor-based compressed sensing to alleviate the computational and storage burden in sampling and recovery. Caiafa and Cichocki [25] study a fast non-iterative tensor compressed sensing method based on a low multi-linear-rank model without assuming certain sparsity patterns, which is suitable for large-scale problems.

7.6 Other Applications

There are still many interdisciplinary applications that are hard to categorize into the categories above. By no means to integrate them all, we introduce two of them and leave others for interested readers to further explore.

7.6.1 Climate Data Analysis. How to fill in the missing entries of the climatological data is also a direction for applying tensor-based methods. Silva et al. [39] apply their hierarchical Tucker algorithms with careful choice of dimension tree to interpolate frequency slices from two generated seismic data. Bahadori et al. [11] analyze two datasets – the U.S. Historical Climatology Network Monthly and the Comprehensive Climate Datasets, which is a collection of the climate records of North America. Their completion method with spatial-temporal auxiliary information was tested with both cokriging and forecasting tasks and proved to be not only more efficient but also more effective in achieving lower estimation error.

7.6.2 Numerical Analysis. Some applications are directly related to numerical analysis, such as the reconstruction of function data and solving the parameterized linear system. One example of the former one is to recover the compress tensors related to functions with singularities [110]. An example of the latter one could be found in the same paper where they complete the solution tensor of a parametrized linear system obtained from a discretized stochastic elliptic PDE with Karhunen–Loève expansion.

8 EXPERIMENTAL SETUP

To facilitate the employment of these methods for practitioners and researchers, in this section, we introduce several important components in experimental settings including ways of synthetic data constructions and evaluation metrics.

8.1 Synthetic Data Construction

An ideal real-world tensor dataset with the known rank and sampling assumption are usually hard to acquire. Based on different assumptions of tensor rank, outside noises, sampling methods, and auxiliary information, researchers have provided different ways to construct synthetic tensors. We introduce three ways of constructions including how to generate a rank-R CP tensor, a rank (R_1, R_2, R_3) Tucker tensor, and a rank-R CP tensor with auxiliary similarity information on each mode. For convenience, we use third-order tensors as examples.

8.1.1 Rank-R CP Tensor. For CP rank, a commonly used way could be found in [2], they construct a third-order tensor $\mathcal{X}$ with CP rank-R by randomly sampling entries of the factor matrices $\mathbf{A} \in \mathbb{R}^{I\times R}, \mathbf{B} \in \mathbb{R}^{J\times R}, \mathbf{C} \in \mathbb{R}^{K\times R}$ from the standard normal distribution $\mathcal{N}(0, 1)$. Each column of the factor matrices is normalized to unit length, and the constructed tensor is denoted as

$$\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] + \eta \frac{||\mathcal{X}||}{||\mathcal{N}||}\mathcal{N}, \tag{49}$$

where $\mathcal{N}$ is a Gaussian noise tensor, and η represents the noise parameter which is sometimes described as Signal-to-Noise Ratio (SNR) [130] or Signal-to-Interference Rate (SIR) [240]. Acar et al. [2] also mentioned that this way of construction ensures the uniqueness of the CP decomposition with probability one since the columns of each factor matrix are linearly independent with probability one, which satisfies the necessary conditions for uniqueness defined in [115]. Missing entries could be denoted by a binary tensor that is uniformly sampled from Bernoulli distribution. Some structured missing entries in the form of missing fibers, slices can also be considered. Concrete missing ratio and sampling methods should base on the incoherence assumptions described in Section 3.4 and specific completion methods we use.

8.1.2 Rank (R_1, R_2, R_3) Tucker Tensor. Similarly, we can construct a tenor with Tucker rank (R_1, R_2, R_3) utilizing the Tucker decomposition $\mathcal{X} = \mathcal{C} \times_1 \mathbf{A}_1 \times \mathbf{A}_2 \times_3 \mathbf{A}_3$. Entries of the core tensor $\mathcal{C} \in \mathbb{R}^{R_1\times R_2\times R_3}$ and factorized matrices could be randomly sampled from $\mathcal{N}(0, 1)$. Factorized matrices are usually orthogonalized. Ways to generate missing entries are the same as we mentioned earlier.

8.1.3 Rank-R CP Tensor with Auxiliary Information. We have introduced several methods for tensor completion with auxiliary similarity information in Section 4. By utilizing the auxiliary information, one might be able to perform exact recovery even for situations of missing slices. Narita et al. [144] propose a way of constructing synthetic CP tensors with tri-diagonal similarity matrices. The factor matrices $\mathbf{A} \in I \times R$, $\mathbf{B} \in J \times R$, and $\mathbf{C} \in K \times R$ are generated by the following linear formula [144]:

$$\begin{aligned}
\mathbf{A}(i, r) &= i\varepsilon_r + \varepsilon'_r, && i = 1, 2, \ldots, 100, r = 1, 2, \ldots, R \\
\mathbf{B}(j, r) &= j\zeta_r + \zeta'_r, && j = 1, 2, \ldots, 100, r = 1, 2, \ldots, R \\
\mathbf{C}(k, r) &= k\eta_r + \eta'_r, && k = 1, 2, \ldots, 100, r = 1, 2, \ldots, R,
\end{aligned}$$

where $\{\varepsilon_r, \varepsilon'_r, \zeta_r, \zeta'_r, \eta_r, \eta'_r\}_{r=1,2,\ldots,R}$ are constants generated from $N(0, 1)$. The synthetic tensor is defined by $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$. Since each factor matrix is linear constructed column by column, the neighboring rows are similar to each other, the similarity matrix of each mode is a tri-diagonal matrix defined as follows:

$$\Theta_i = \begin{bmatrix} 0 & 1 & 0 & \ldots \\ 1 & 0 & 1 & \ldots \\ 0 & 1 & 0 & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{50}$$

Table 5. Comparison of Several Tensor Decomposition/Completion Repositories

Platform	Repository name	Tensor decomposition	Tensor completion	Auxiliary information	Large-scale	Dynamic	GUI
Java	FlexiFaCT [16]	✓	✓	✓	✓		
R	rTensor [121]	✓					
	TensorBF [99]	✓					
	PTAk [120]	✓					
	tensr[10]	✓					
Matlab	Tensor Toolbox [10]	✓	✓				
	N-way Toolbox [6]	✓	✓				✓(CuBatch)
Python	TensorToolbox [157]	✓					
	ttpy[11]	✓				✓	
	scikit-tensor [146]	✓					
	PyTensor [226]	✓				✓	
	PyTen	✓	✓	✓	✓	✓	✓

PyTen: Our developed package.

8.2 Evaluation Metrics

As completion problems could be generalized as prediction problems, metrics are usually selected based on concrete applications such as RMSE in recommender systems, AUC in binary classification, nDCG in information retrieval, and so on. We introduce several commonly used metrics below. Not limited thereto, one could sselect more suitable ones based on specific problems and situations.

8.2.1 Relative Standard Error/Percentage of Fitness/Tensor Completion Score. Relative Error is the most commonly used evaluation metric, it is formally defined as

$$\text{Relative Standard Error} = \frac{\|\mathcal{X}_{\text{rec}} - \mathcal{X}_{\text{real}}\|_F}{\|\mathcal{X}_{\text{real}}\|_F}, \quad \text{Percentage of Fitness} = 1 - \text{Relative Standard Error}$$

where $\mathcal{X}_{\text{rec}}$ is the recovered tensor, $\mathcal{X}_{\text{real}}$ is the real tensor. Percentage of Fitness is a similar metric measuring the fitness of the reconstructed tensor and the ground-truth tensor. A refined relative error, called Tensor Completion Score by Acar et al. [2], is also used defined as

$$\text{TCS} = \frac{\|(1 - \mathcal{W}) * (\mathcal{X}_{\text{rec}} - \mathcal{X}_{\text{real}})\|_F}{\|(1 - \mathcal{W}) * \mathcal{X}_{\text{real}}\|_F},$$

where $\mathcal{W}$ is the indication tensor of the observations.

8.2.2 MSE/RMSE/MAE. Mean-square error and root-mean-square deviation, and mean absolute error are frequently-used metrics especially in recommendation problems [92, 214, 239]. Their formal definitions are

$$\begin{gathered} \text{MSE} = \frac{\|\mathcal{W} * (\mathcal{X}_{\text{rec}} - \mathcal{X}_{\text{real}})\|_F^2}{\|\mathcal{W}\|_F^2} \quad \text{RMSE} = \frac{\|\mathcal{W} * (\mathcal{X}_{\text{rec}} - \mathcal{X}_{\text{real}})\|_F}{\|\mathcal{W}\|_F} \\ \text{MAE} = \frac{1}{\sum |\mathcal{W}|} \sum \mathcal{W} * |\mathcal{X}_{\text{rec}} - \mathcal{X}_{\text{real}}| \end{gathered} . \tag{51}$$

[10] https://cran.r-project.org/web/packages/tensr/index.html.
[11] https://github.com/oseledets/ttpy.

8.2.3 Precision/Recall/Area Under Curve(AUC). Some applications could be treated as classification problems such as link predictions or recommendations. Thus, several metrics for classification problems such as precision (or precision@k), recall, F-measure and AUC score could be found in some tensor completion papers [60, 61, 156, 168, 192]. To calculate these metrics such as AUC score, one could either extract all missing entries or average these scores on each slice based on their requirements in real-world applications.

8.3 Softwares

Repositories for tensor analysis have been developed on various platforms in recent years [106]. Table 5 gives a brief integration and comparison of tensor decomposition and completion repositories on different platforms. Since lack of work has been done on developing Python packages for the tensor decomposition and completion, we develop a new open-source Python package covering several widely used tensor decomposition and completion algorithms for both researchers and practitioners. Details of the open-source package could be found on the website.[12]

9 CONCLUSION AND FUTURE OUTLOOK

Tensor completion problem that permeates through a wide range of real-world applications has become an actively studied field attracting increasing attention. Recent advances in both theory and practice provide us versatile and potent methods to apply on various problems even with auxiliary information, large-scale data, and dynamical patterns. However, the deeper the delving, the more challenges we will confront. Reviewing the past offers us a better vision to look forward to future directions.

Volume: Tensor Completion with Large-Scale and High-Order Data. Large-scale and high-order properties are usually involved in real-world tensors. The increasing volume of tensor-structured data puts higher demands of the completion algorithms from both time and space perspectives. As we have described in Section 5, two main challenges underlying in the large-scale tensor completion problem are intermediate data explosion and data decentralization. Though multiple approaches have been proposed to address the scalability problem, there still exists a vast room for innovation and promotion. First, space and time complexities still highly limit the practical application of tensor decomposition/completion methods. Second, most of the existing work only focuses on low-dimensional tensor completion problem. Although methods based on hierarchical Tucker representations have been utilized to address high-order problems [161], their generalizability and robustness remain unclear. Third, some methods such as sampling methods [18, 111] require specific assumptions (e.g., incoherence assumptions), which are hard to verify and guarantee for real-world complex tensors. Fourth, as deep learning models are usually data-hungry, is it possible to combine tensor-based algorithms with deep learning techniques to address the volume challenge as well as improving completion accuracy?

Velocity: Tensor Completion in Dynamic Data Analytics. With the rapid velocity of data growth, dynamic tensor completion problem has been paid more and more attention recently as we have outlined in Section 6. It is still riddled with various open problems in line with the intricate multi-aspect dynamic patterns. First, how to handle dynamically changing tensors? While some recent attempts have been made on solving the streaming tensor completion problem [138, 186], efforts are still needed for coordinating multi-dimensional dynamics induced by the uncertainty of tensor size and modes. Second, what are the influences of different dynamical patterns on the model construction, parameter selection, and imputation effectiveness? For example, dynamically changed tensors may cause the fixed rank assumption inapplicable. Arbitrarily adopting static

[12] https://github.com/tamu-helios/pyten.

strategies might be inappropriate and reduce the completion effectiveness. Third, whether there exist theoretical guarantees for the proposed algorithm under dynamic settings or not? To our best knowledge, bare of existing efforts have been put to provide theoretical analysis such as convergence or sufficient conditions and statistical assumptions for dynamic tensor completion.

Variety: Tensor Completion with Heterogeneous Data Sources. In practice, tensor structured data may exhibit heterogeneous properties when viewed obtained via different routes or viewed from different perspectives. It is critical, yet challenging, to consider heterogeneous data sources to benefit the tensor completion problem, especially for structured missing data [61, 222]. First, as introduced in Section 4, the heterogeneous data sources could serve as auxiliary information to mitigate statistical assumptions while enhancing the completion effectiveness. Thus, exploring potential ways to effectively incorporate these heterogeneous data source could be a valuable direction. For example, combining the heterogeneous deep learning techniques with tensor completion algorithms could be one possible approach. Second, in a wide variety of real-world data-driven applications, domain knowledge and expertise are beneficial for modeling, analysis, understanding, and completion. Several recent works have tried to incorporate human knowledge into completion tasks and prove to acquire favorable performance, such as in the area of healthcare [208]. How to combine domain knowledge with tensor completion problems could be another promising future direction for exploitation and exploration. Finally, incorporating the heterogeneous data under large-scale and dynamic settings may also be interesting and meaningful to pursue.

Other Aspects. Besides the aforementioned "3V" challenges, as the big data research may also concern about the "veracity" and "value," it is intriguing to explore the tensor completion problems from these two perspectives. From the "veracity" perspective, the uncertainties of the high-order datasets, the tensor completion algorithms, and completion results could be one meaningful and promising direction to pursue. From the "value" perspective, whether the data is valuable to utilize or not and if the completion assumptions, methods, and results are valuable and practical for real-world applications should also be concerned in the future.

ACKNOWLEDGMENTS

The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

REFERENCES

[1] Evrim Acar, Daniel M. Dunlavy, and Tamara G. Kolda. 2009. Link prediction on evolving data using matrix and tensor factorizations. In *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW'09).*

[2] Evrim Acar, Daniel M. Dunlavy, Tamara G. Kolda, and Morten Mørup. 2011. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems* 106, 1 (2011), 41–56.

[3] Evrim Acar, Tamara G. Kolda, and Daniel M. Dunlavy. 2011. All-at-once optimization for coupled matrix and tensor factorizations. arXiv:1105.3422.

[4] Evrim Acar and Bülent Yener. 2009. Unsupervised multiway data analysis: A literature survey. *IEEE Transactions on Knowledge and Data Engineering* 21, 1 (2009), 6–20.

[5] Claus A. Andersson and Rasmus Bro. 1998. Improving the speed of multi-way algorithms:: Part I. Tucker3. *Chemometrics and Intelligent Laboratory Systems* 42, 1 (1998), 93–103.

[6] Claus A. Andersson and Rasmus Bro. 2000. The N-way toolbox for MATLAB. *Chemometrics and Intelligent Laboratory Systems* 52, 1 (2000), 1–4.

[7] Carl J. Appellof and E. R. Davidson. 1981. Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents. *Analytical Chemistry* 53, 13 (1981), 2053–2056.

[8] Morteza Ashraphijuo and Xiaodong Wang. 2017. Fundamental conditions for low-CP-rank tensor completion. *Journal of Machine Learning Research* 18, 1 (2017), 2116–2145.

[9] Brett W. Bader and Tamara G. Kolda. 2006. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software* 32, 4 (2006), 635–653.

[10] Brett W. Bader, Tamara G. Kolda, et al. 2012. Matlab tensor toolbox version 2.5. (January 2012). Retrieved from https://www.tensortoolbox.org/.

[11] Mohammad Taha Bahadori, Qi Rose Yu, and Yan Liu. 2014. Fast multivariate spatio-temporal analysis via low rank tensor learning. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14)*.

[12] Arindam Banerjee, Sugato Basu, and Srujana Merugu. 2007. Multi-way clustering on relation graphs. In *Proceedings of the 2007 SIAM International Conference on Data Mining (ICDM'07)*.

[13] Boaz Barak and Ankur Moitra. 2015. Tensor prediction, rademacher complexity and random 3-xor. CoRR, abs/1501.06521.

[14] Boaz Barak and Ankur Moitra. 2016. Noisy tensor completion via the sum-of-squares hierarchy. In *Proceedings of Conference on Learning Theory*.

[15] Juan Andrés Bazerque, Gonzalo Mateos, and Georgios B. Giannakis. 2013. Rank regularization and Bayesian inference for tensor completion and extrapolation. *IEEE Transactions on Signal Processing* 61, 22 (2013), 5689–5703.

[16] Alex Beutel, Partha Pratim Talukdar, Abhimanu Kumar, Christos Faloutsos, Evangelos E. Papalexakis, and Eric P. Xing. 2014. Flexifact: Scalable flexible factorization of coupled tensors on hadoop. In *Proceedings of International Conference on Data Mining (ICDM'14)*.

[17] Mark Beyer. 2011. Gartner Says Solving 'Big data' Challenge Involves More Than Just Managing Volumes of Data. Gartner. Archived from the Original on (2011).

[18] Srinadh Bhojanapalli and Sujay Sanghavi. 2015. A new sampling technique for tensors. arXiv:1502.05023.

[19] Rasmus Bro. 1997. PARAFAC. Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems* 38 (1997), 149–171.

[20] Rasmus Bro. 1998. Multi-way analysis in the food industry: Models, algorithms, and applications. In *Proc. ICSLP 2000 MRI, EPG and EMA*.

[21] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S. Huang. 2011. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 8 (2011), 1548–1560.

[22] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20, 4 (2010), 1956–1982.

[23] Cesar F. Caiafa and Andrzej Cichocki. 2012. Block sparse representations of tensors using Kronecker bases. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

[24] Cesar F. Caiafa and Andrzej Cichocki. 2013. Computing sparse representations of multidimensional signals using kronecker bases. *Neural Computation* 25, 1 (2013), 186–220.

[25] Cesar F. Caiafa and Andrzej Cichocki. 2015. Stable, robust, and super fast reconstruction of tensors using multi-way projections. *IEEE Transactions on Signal Processing* 63, 3 (2015), 780–793.

[26] Emmanuel Candes and Benjamin Recht. 2012. Exact matrix completion via convex optimization. *Communications of the ACM* 55, 6 (2012), 111–119.

[27] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. 2011. Robust principal component analysis? *Journal of the ACM* 58, 3 (2011), Article 11.

[28] Emmanuel J. Candes and Yaniv Plan. 2010. Matrix completion with noise. *Proceedings of IEEE* 98, 6 (2010), 925–936.

[29] Emmanuel J. Candès, Justin Romberg, and Terence Tao. 2006. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory* 52, 2 (2006), 489–509.

[30] J. Douglas Carroll and Jih-Jie Chang. 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika* 53, 3 (1970), 283–319.

[31] J. Douglas Carroll, Sandra Pruzansky, and Joseph B. Kruskal. 1980. CANDELINC: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters. *Psychometrika* 45, 1 (1980), 3–24.

[32] Chris Chatfield and Mohammad Yar. 1988. Holt-Winters forecasting: Some practical issues. *Statistician* 37, 2 (1988), 129–140.

[33] Yi-Lei Chen, Chiou-Ting Hsu, and Hong-Yuan Mark Liao. 2014. Simultaneous tensor decomposition and completion using factor priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 3 (2014), 577–591.

[34] Hao Cheng, Yaoliang Yu, Xinhua Zhang, Eric Xing, and Dale Schuurmans. 2016. Scalable and sound low-rank tensor learning. In *Artificial Intelligence and Statistics*.

[35] Joon Hee Choi and S. Vishwanathan. 2014. DFacTo: Distributed factorization of tensors. In *Proceedings of NIPS*.

[36] Wei Chu and Zoubin Ghahramani. 2009. Probabilistic models for incomplete multi-dimensional arrays. In *Proceedings of AISTATS*.

[37] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. 2009. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. John Wiley and Sons.

[38] Pierre Comon. 2004. Canonical tensor decompositions. In *Proceedings of Workshop on Tensor Decompositions, Palo Alto, CA*.

[39] Curt Da Silva and Felix J. Herrmann. 2013. Hierarchical tucker tensor optimization-applications to tensor completion. In *Proceedings of International Conference on Sampling Theory and Applications*.

[40] Curt Da and Felix J. Herrmann. 2015. Optimization on the hierarchical tucker manifold–applications to tensor completion. *Linear Algebra and its Applications* 481 (2015), 131–173.

[41] Justin Dauwels, Lalit Garg, Arul Earnest, and Leong Khai Pang. 2011. Handling missing data in medical questionnaires using tensor decompositions. In *Proceedings of ICICS*.

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

[43] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. 2000. 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 (2000), 1324–1342.

[44] Inderjit S. Dhillon and Suvrit Sra. 2005. Generalized nonnegative matrix approximations with Bregman divergences. In *Proceedings of NIPS*.

[45] David L. Donoho. 2006. Compressed sensing. *IEEE Transactions on Information Theory* 52, 4 (2006), 1289–1306.

[46] Marco F. Duarte and Richard G. Baraniuk. 2012. Kronecker compressive sensing. *IEEE Transactions on Image Processing* 21, 2 (2012), 494–504.

[47] Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar. 2011. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data* 5, 2 (2011), Article 10.

[48] Jonathan Eckstein and Dimitri P. Bertsekas. 1992. On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming* 55, 1–3 (1992), 293–318.

[49] Yonina C. Eldar and Gitta Kutyniok. 2012. *Compressed Sensing: Theory and Applications*. Cambridge University Press.

[50] Yonina C. Eldar, Deanna Needell, and Yaniv Plan. 2012. Uniqueness conditions for low-rank matrix recovery. *Applied and Computational Harmonic Analysis* 33, 2 (2012), 309–314.

[51] Beyza Ermiş, Evrim Acar, and A. Taylan Cemgil. 2012. Link prediction via generalized coupled tensor factorisation. arXiv:1208.6231.

[52] Beyza Ermiş, Evrim Acar, and A. Taylan Cemgil. 2015. Link prediction in heterogeneous data via generalized coupled tensor factorization. *Data Mining and Knowledge Discovery* 29, 1 (2015), 203–236.

[53] Nicolaas Klaas M. Faber, Rasmus Bro, and Philip K. Hopke. 2003. Recent developments in CANDECOMP/PARAFAC algorithms: A critical review. *Chemometrics and Intelligent Laboratory Systems* 79, 11 (2003), 1103–1108.

[54] Hadi Fanaee-T. and João Gama. 2015. Multi-aspect-streaming tensor analysis. *Knowledge-Based Systems* 89 (2015), 332–345.

[55] Jiashi Feng, Huan Xu, and Shuicheng Yan. 2013. Online robust PCA via stochastic optimization. In *Proceedings of NIPS*.

[56] Marko Filipović and Ante Jukić. 2015. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimensional Systems and Signal Processing* 26, 3 (2015), 677–692.

[57] David H. Foster, Sérgio M. C. Nascimento, and Kinjiro Amano. 2004. Information limits on neural identification of colored surfaces in natural scenes. *Visual Neuroscience* 21, 3 (2004), 331–336.

[58] Shmuel Friedland, Qun Li, and Dan Schonfeld. 2014. Compressive sensing of sparse tensors. *IEEE Transactions on Image Processing* 23, 10 (2014), 4438–4447.

[59] Silvia Gandy, Benjamin Recht, and Isao Yamada. 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems* 27, 2 (2011), Article 025010.

[60] Hancheng Ge, James Caverlee, and Haokai Lu. 2016. Taper: A contextual tensor-based approach for personalized expert recommendation. In *Proceedings of RecSys*.

[61] Hancheng Ge, James Caverlee, Nan Zhang, and Anna Squicciarini. 2016. Uncovering the spatio-temporal dynamics of memes in the presence of incomplete information. In *Proceedings of CIKM*.

[62] Hancheng Ge, Kai Zhang, Majid Alfifi, Xia Hu, and James Caverlee. 2018. DisTenC: A distributed algorithm for scalable tensor completion on spark. In *Proceedings of ICDE*.

[63] Rainer Gemulla, Erik Nijkamp, Peter J. Haas, and Yannis Sismanis. 2011. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 69–77.

[64] Xin Geng and Kate Smith-Miles. 2009. Facial age estimation by multilinear subspace analysis. In *Proceedings of ICASSP*.

[65] Xin Geng, Kate Smith-Miles, Zhi-Hua Zhou, and Liang Wang. 2011. Face image modeling by multilinear subspace analysis with missing values. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41, 3 (2011), 881–892.

[66] Navid Ghadermarzy, Yaniv Plan, and Özgür Yılmaz. 2017. Near-optimal sample complexity for convex tensor completion. arXiv:1711.04965.

[67] Donald Goldfarb and Zhiwei Qin. 2014. Robust low-rank tensor recovery: Models and algorithms. *SIAM Journal on Matrix Analysis and Applications* 35, 1 (2014), 225–253.
[68] Lars Grasedyck, Melanie Kluge, and Sebastian Krämer. 2015. Alternating least squares tensor completion in the TT-format. arXiv:1509.00311.
[69] Lars Grasedyck, Daniel Kressner, and Christine Tobler. 2013. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen* 36, 1 (2013), 53–78.
[70] David Gross. 2011. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory* 57, 3 (2011), 1548–1566.
[71] Xiawei Guo, Quanming Yao, and James Tin-Yau Kwok. 2017. Efficient sparse low-rank tensor completion using the frank-wolfe algorithm. In *Proceedings of AAAI*.
[72] Wolfgang Hackbusch. 2012. *Tensor Spaces and Numerical Tensor Calculus*. Springer Science & Business Media.
[73] Zaid Harchaoui, Anatoli Juditsky, and Arkadi Nemirovski. 2015. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming* 152, 1–2 (2015), 75–112.
[74] Richard A. Harshman. 1970. *Foundations of the PARAFAC Procedure: Models and Conditions for an "Explanatory" Multi-Modal Factor Analysis. University of California at Los Angeles, Los Angeles, CA*.
[75] Johan Håstad. 1990. Tensor rank is NP-complete. *Journal of Algorithms* 11, 4 (1990), 644–654.
[76] René Henrion. 1994. N-way principal component analysis theory, algorithms and applications. *Chemometrics and Intelligent Laboratory Systems* 25, 1 (1994), 1–23.
[77] David Hernandez. 2010. Simple tensor products. *Inventiones Mathematicae* 181, 3 (2010), 649–675.
[78] Frank L. Hitchcock. 1928. Multiple invariants and generalized rank of a p-way matrix or tensor. *Journal of Mathematics and Physics* 7, 1–4 (1928), 39–79.
[79] Frank L. Hitchcock. 1927. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics* 6, 1–4 (1927), 164–189.
[80] Joyce C. Ho, Joydeep Ghosh, Steve R. Steinhubl, Walter F. Stewart, Joshua C. Denny, Bradley A. Malin, and Jimeng Sun. 2014. Limestone: High-throughput candidate phenotype generation via tensor factorization. *Journal of Biomedical Informatics* 52 (2014), 199–211.
[81] Weiming Hu, Xi Li, Xiaoqin Zhang, Xinchu Shi, Stephen Maybank, and Zhongfei Zhang. 2011. Incremental tensor subspace learning and its applications to foreground segmentation and tracking. *International Journal of Computer Vision* 91, 3 (2011), 303–327.
[82] Bo Huang, Cun Mu, Donald Goldfarb, and John Wright. 2014. Provable low-rank tensor recovery. *Optimization-Online*.
[83] Furong Huang, U. N. Niranjan, M. Hakeem, and Animashree Anandkumar. 2013. Fast detection of overlapping communities via online tensor methods. arXiv:1309.0787.
[84] Masaaki Imaizumi, Takanori Maehara, and Kohei Hayashi. 2017. On tensor train rank minimization: Statistical efficiency and scalable algorithm. In *Proceedings of NIPS*.
[85] Prateek Jain and Sewoong Oh. 2014. Provable tensor factorization with missing data. In *Proceedings of NIPS*.
[86] Swayambhoo Jain, Alexander Gutierrez, and Jarvis Haupt. 2017. Noisy tensor completion for tensors with a sparse canonical polyadic factor. In *Proceedings of the 2017 IEEE International Symposium on Information Theory (ISIT'17)*. IEEE, 2153–2157.
[87] Sajid Javed, Thierry Bouwmans, and Soon Ki Jung. 2015. Stochastic decomposition into low rank and sparse tensor for robust background subtraction. In *Proceedings of IET*.
[88] Inah Jeon, Evangelos E. Papalexakis, U. Kang, and Christos Faloutsos. 2015. Haten2: Billion-scale tensor decompositions. In *Proceedings of IEEE ICDE*.
[89] Teng-Yu Ji, Ting-Zhu Huang, Xi-Le Zhao, Tian-Hui Ma, and Gang Liu. 2016. Tensor completion using total variation and low-rank matrix factorization. *Information Sciences* 326 (2016), 243–257.
[90] Charles R. Johnson. 1990. Matrix completion problems: A survey. *Symposia in Applied Mathematics* 40 (1990), 171–198.
[91] U. Kang, Evangelos Papalexakis, Abhay Harpale, and Christos Faloutsos. 2012. Gigatensor: Scaling tensor analysis up by 100 times-algorithms and discoveries. In *Proceedings of SIGKDD*.
[92] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems*.
[93] Lars Karlsson, Daniel Kressner, and André Uschmajew. 2016. Parallel algorithms for tensor completion in the CP format. *Parallel Computing* 57 (2016), 222–234.
[94] Hiroyuki Kasai. 2016. Online low-rank tensor subspace tracking from incomplete data by CP decomposition using recursive least squares. In *Proceedings of ICASSP*.

[95] Hiroyuki Kasai and Bamdev Mishra. 2015. Riemannian preconditioning for tensor completion. In *Proceedings of the NeurIPS Workshop on Optimization for Machine Learning (OPT'15)*.

[96] Hiroyuki Kasai and Bamdev Mishra. 2016. Low-rank tensor completion: A Riemannian manifold preconditioning approach. In *Proceedings of the International Conference on Machine Learning*. 1012–1021.

[97] Hisashi Kashima, Tsuyoshi Kato, Yoshihiro Yamanishi, Masashi Sugiyama, and Koji Tsuda. 2009. Link propagation: A fast semi-supervised learning algorithm for link prediction. In *Proceedings of ICDM*.

[98] Oguz Kaya and Bora Uçar. 2016. *Parallel CP decomposition of sparse tensors using dimension trees*. Ph.D. dissertation. Inria-Research Centre Grenoble–Rhône-Alpes.

[99] Suleiman A. Khan and Muhammad Ammad-ud-din. 2016. tensorBF: An R package for Bayesian tensor factorization. bioRxiv.

[100] Henk A. L. Kiers. 1997. Weighted least squares fitting using ordinary least squares algorithms. *Psychometrika* 62, 2 (1997), 251–266.

[101] Henk A. L. Kiers. 2000. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics* 14,3 (2000), 105–122.

[102] Henk A. L. Kiers and Iven Van Mechelen. 2001. Three-way component analysis: Principles and illustrative application. *Psychological Methods* 6, 1 (2001), 84–110.

[103] Henk A. L. Kiers, Jos M.F. Ten Berge, and Rasmus Bro. 1999. PARAFAC2-Part I. A direct fitting algorithm for the PARAFAC2 model. *Journal of Chemometrics* 13 (1999), 275–294.

[104] Misha E. Kilmer, Karen Braman, Ning Hao, and Randy C. Hoover. 2013. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications* 34, 1 (2013), 148–172.

[105] Morris Kline. 1990. *Mathematical Thought From Ancient to Modern Times: Volume 3*. OUP.

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

[107] Tamara G. Kolda and Jimeng Sun. 2008. Scalable tensor decompositions for multi-aspect data mining. In *Proceedings of IEEE ICDM*.

[108] Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of ACM SIGKDD* .

[109] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[110] Daniel Kressner, Michael Steinlechner, and Bart Vandereycken. 2014. Low-rank tensor completion by riemannian optimization. *BIT Numerical Mathematics* 54, 2 (2014), 447–468.

[111] Akshay Krishnamurthy and Aarti Singh. 2013. Low-rank matrix and tensor completion via adaptive sampling. In *Proceedings of NIPS*.

[112] Pieter M. Kroonenberg. 1983. *Three-Mode Principal Component Analysis: Theory and Applications*. DSWO Press.

[113] Pieter M. Kroonenberg. 2008. *Applied Multiway Data Analysis*, Vol. 702. John Wiley and Sons.

[114] Pieter M. Kroonenberg and Jan De Leeuw. 1980. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika* 45, 1(1980), 69–97.

[115] Joseph B. Kruskal. 1977. Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications* 18, 2 (1977), 95–138.

[116] Joseph B. Kruskal. 1989. Rank, decomposition, and uniqueness for 3-way and n-way arrays. In *Multiway Data Analysis*. R. Coppi and S. Bolasco (Eds.). North-Holland Publishing Co.

[117] Hemank Lamba, Vaishnavh Nagarajan, Kijung Shin, and Naji Shajarisales. 2016. Incorporating side information in tensor completion. In *Proceedings of WWW*.

[118] Monique Laurent. 2009. Matrix completion problems. In *Encyclopedia of Optimization*. Springer.

[119] John Lee. 2012. *Introduction to Smooth Manifolds*. Springer Science and Business Media.

[120] Didier G. Leibovici. 2010. Spatio-temporal multiway decompositions using principal tensor analysis on k-modes: The R package PTAk. *Journal of Statistical Software* 34, 10 (2010), 1–34.

[121] James Li, Jacob Bien, Martin Wells, and Maintainer James Li. 2015. Tools for Tensor Analysis and Decomposition. Package 'rTensor' .

[122] Jiajia Li, Jee Choi, Ioakeim Perros, Jimeng Sun, and Richard Vuduc. 2017. Model-driven sparse CP decomposition for higher-order tensors. In *Proceedings of the IEEE International IPDPS*.

[123] Qun Li, Dan Schonfeld, and Shmuel Friedland. 2013. Generalized tensor compressive sensing. In *Proceedings of the IEEE ICME*.

[124] Wu-Jun Li and Dit Yan Yeung. 2009. Relation regularized matrix factorization. In *Proceedings of IJCAI*.

[125] Yong Li, Wenrui Dai, and Hongkai Xiong. 2016. Compressive tensor sampling with structured sparsity. In *Proceedings of the IEEE DCC*.

[126] Yin Li, Junchi Yan, Yue Zhou, and Jie Yang. 2010. Optimum subspace learning and error correction for tensors. In *Proceedings of Computer Vision–ECCV*.
[127] Yu-Ru Lin, Jimeng Sun, Paul Castro, Ravi Konuru, Hari Sundaram, and Aisling Kelliher. 2009. Metafac: Community discovery via relational hypergraph factorization. In *Proceedings of KDD*.
[128] Zhouchen Lin, Minming Chen, and Yi Ma. 2010. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv:1009.5055.
[129] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. 2009. Tensor completion for estimating missing values in visual data. In *Proceedings of ICCV*.
[130] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. 2013. Tensor completion for estimating missing values in visual data. In *Proceedings of TPAMI*.
[131] Yuanyuan Liu and Fanhua Shang. 2013. An efficient matrix factorization method for tensor completion. *IEEE Signal Processing Letters* 20,4 (2013), 307–310.
[132] Yuanyuan Liu, Fanhua Shang, Hong Cheng, James Cheng, and Hanghang Tong. 2014. Factor matrix trace norm minimization for low-rank tensor completion. In *Proceedings of ICDM*.
[133] Yuanyuan Liu, Fanhua Shang, Wei Fan, James Cheng, and Hong Cheng. 2014. Generalized higher-order orthogonal iteration for tensor decomposition and completion. In *Proceedings of NIPS*.
[134] Yuanyuan Liu, Fanhua Shang, Licheng Jiao, James Cheng, and Hong Cheng. 2015. Trace norm regularized CANDECOMP/PARAFAC decomposition with missing data. *IEEE Transactions on Cybernetics* 45, 11 (2015), 2437–2448.
[135] Haiping Lu, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. 2011. A survey of multilinear subspace learning for tensor data. *Pattern Recognition* 44, 7 (2011), 1540–1551.
[136] Xiang Ma, Dan Schonfeld, and Ashfaq Khokhar. 2009. Dynamic updating and downdating matrix SVD and tensor HOSVD for adaptive indexing and retrieval of motion trajectories. In *Proceedings of ICASSP*.
[137] Morteza Mardani, Gonzalo Mateos, and Georgios B. Giannakis. 2014. Imputation of streaming low-rank tensor data. In *Proceedings of IEEE SAM*.
[138] Morteza Mardani, Gonzalo Mateos, and Georgios B. Giannakis. 2015. Subspace learning and imputation for streaming big data matrices and tensors. In *Proceedings of IEEE TSP*.
[139] Yasuko Matsubara, Yasushi Sakurai, Christos Faloutsos, Tomoharu Iwata, and Masatoshi Yoshikawa. 2012. Fast mining and forecasting of complex time-stamped events. In *Proceedings of KDD*.
[140] X. Meng, A. J. Morris, and E. B. Martin. 2003. On-line monitoring of batch processes using a PARAFAC representation. *Journal of Chemometrics* 17, 1 (2003), 65–81.
[141] Bamdev Mishra and Rodolphe Sepulchre. 2014. R3MC: A Riemannian three-factor algorithm for low-rank matrix completion. In *Proceedings of CDC*.
[142] Andrea Montanari and Nike Sun. 2018. Spectral algorithms for tensor completion. *Communications on Pure and Applied Mathematics* 71, 11 (2018), 2381–2425.
[143] Cun Mu, Bo Huang, John Wright, and Donald Goldfarb. 2014. Square deal: Lower bounds and improved relaxations for tensor recovery. In*Proceedings of ICML*.
[144] Atsuhiro Narita, Kohei Hayashi, Ryota Tomioka, and Hisashi Kashima. 2011. Tensor factorization using auxiliary information. In *Proceedings of ECML PKDD*.
[145] Nagarajan Natarajan and Inderjit S. Dhillon. 2014. Inductive matrix completion for predicting gene–disease associations. *Bioinformatics* 30, 12 (2014), i60–68.
[146] Maximilian Nickel. [n. d.]. Scikit-tensor: Python library for multilinear algebra and tensor factorizations.
[147] Madhav Nimishakavi, Bamdev Mishra, Manish Gupta, and Partha Talukdar. 2018. Inductive framework for multi-aspect streaming tensor completion with side information. In *Proceedings of the Conference on Information and Knowledge Management*.
[148] Dimitri Nion and Nicholas D. Sidiropoulos. 2009. Adaptive algorithms to track the PARAFAC decomposition of a third-order tensor. *IEEE Transactions on Signal Processing* 57, 6 (2009), 2299–2310.
[149] Yasunori Nishimori and Shotaro Akaho. 2005. Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold. *Neurocomputing* 67 (2005), 106–135.
[150] Sejoon Oh, Namyong Park, Lee Sael, and U. Kang. 2018. Scalable tucker factorization for sparse tensors-algorithms and discoveries. In *Proceedings of the 2018 IEEE 34th International Conference on Data Engineering (ICDE'18)*. IEEE, 1120–1131.
[151] Ivan V. Oseledets. 2011. Tensor-train decomposition. *SIAM Journal on Scientific Computing* 33, 5 (2011), 2295–2317.
[152] Pentti Paatero. 1997. A weighted non-negative least squares algorithm for three-way 'PARAFAC' factor analysis. *Chemometrics and Intelligent Laboratory Systems* 38, 2 (1997), 223–242.
[153] Evangelia Pantraki and Constantine Kotropoulos. 2015. Automatic image tagging and recommendation via PARAFAC2. In *Proceedings IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP'15)*.

[154] Evangelos E. Papalexakis, Christos Faloutsos, and Nicholas D. Sidiropoulos. 2016. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. In *Proceedings of TIST*.
[155] T. Papastergiou and V. Megalooikonomou. 2017. A distributed proximal gradient descent method for tensor completion. In *Proceedings of IEEE International Conference on Big Data (Big Data'17)*.
[156] Jing Peng, Daniel Dajun Zeng, Huimin Zhao, and Fei-yue Wang. 2010. Collaborative filtering in social tagging systems based on joint item-tag recommendations. In *Proceedings of ACM CIKM*.
[157] M. G. Peterson and D. M. Dunlavy. 2014. Tensor toolbox: Wrapping to PYTHON using SWIG. *CCR*.
[158] Anh Huy Phan and Andrzej Cichocki. 2011. PARAFAC algorithms for large-scale problems. *Neurocomputing* 74, 11 (2011), 1970–1984.
[159] Ho N. Phien, Hoang D. Tuan, Johann A. Bengua, and Minh N. Do. 2016. Efficient tensor completion: Low-rank tensor train. arXiv:1601.01083.
[160] Piyush Rai, Yingjian Wang, Shengbo Guo, Gary Chen, David Dunson, and Lawrence Carin. 2014. Scalable Bayesian low-rank decomposition of incomplete multiway tensors. In *Proceedings of International Conference on Machine Learning*.
[161] Holger Rauhut, Reinhold Schneider, and Željka Stojanac. 2015. Tensor completion in hierarchical tensor representations. In *Compressed Sensing and its Applications*. Birkhäuser Verlag.
[162] Holger Rauhut, Reinhold Schneider, and Željka Stojanac. 2017. Low rank tensor recovery via iterative hard thresholding. *Linear Algebra and Its Applications* 523 (2017), 220–262.
[163] Holger Rauhut and Željka Stojanac. 2015. Tensor theta norms and low rank recovery. arXiv:1505.05175.
[164] Benjamin Recht. 2011. A simpler approach to matrix completion. *Journal of Machine Learning Research* 12, (2011), 3413–3430.
[165] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. 2010. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review* 52, 3 (2010), 471–501.
[166] Steffen Rendle. 2010. Factorization machines. In *Proceedings of ICDM*.
[167] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. 2009. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of KDD*.
[168] Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of WSDM*.
[169] Bernardino Romera-Paredes and Massimiliano Pontil. 2013. A new convex relaxation for tensor completion. In *Proceedings of NIPS*.
[170] Lee Sael, Inah Jeon, and U. Kang. 2015. Scalable tensor mining. *Big Data Research* 2, 2 (2015), 82–86.
[171] Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of ICML*.
[172] Kijung Shin, Lee Sael, and U. Kang. 2017. Fully scalable methods for distributed tensor factorization. *IEEE Transactions on Knowledge and Data Engineering* 29, 1 (2017), 100–113.
[173] Si Si, Kai-Yang Chiang, Cho-Jui Hsieh, Nikhil Rao, and Inderjit S. Dhillon. 2016. Goal-directed inductive matrix completion. In *Proceedings of SIGKDD*.
[174] Nicholas D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E. Papalexakis, and Christos Faloutsos. 2016. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing* 65, 13 (2017), 3551–3582.
[175] Nicholas D. Sidiropoulos and Anastasios Kyrillidis. 2012. Multi-way compressed sensing for sparse low-rank tensors. *IEEE Signal Processing Letters* 19, 11 (2012), 757–760.
[176] Marco Signoretto, Lieven De Lathauwer, and Johan A. K. Suykens. 2010. Nuclear norms for tensors and their use for convex multilinear estimation. *Linear Algebra and Its Applications* 43 (2010), to be submitted.
[177] Marco Signoretto, Quoc Tran Dinh, Lieven De Lathauwer, and Johan A. K. Suykens. 2014. Learning with tensors: A framework based on convex optimization and spectral regularization. *Machine Learning* 94, 3 (2014), 303–351.
[178] Marco Signoretto, Raf Van de Plas, Bart De Moor, and Johan A. K. Suykens. 2011. Tensor versus matrix completion: A comparison with application to spectral data. *IEEE Signal Processing Letters* 18, 7 (2011), 403–406.
[179] Ajit P. Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. *KDD* (2008).
[180] Age Smilde, Rasmus Bro, and Paul Geladi. 2005. *Multi-Way Analysis: Applications in the Chemical Sciences*. John Wiley & Sons.
[181] Age K. Smilde, Johan A. Westerhuis, and Ricard Boque. 2000. Multiway multiblock component and covariates regression models. *Journal of Chemometrics* 14, 3 (2000), 301–331.
[182] Shaden Smith, Alec Beri, and George Karypis. 2017. Constrained tensor factorization with accelerated AO-ADMM. In *Proceedings of 46th International Conference on Parallel Processing (ICPP'17)*.
[183] Shaden Smith and George Karypis. 2017. Accelerating the tucker decomposition with compressed sparse tensors. In *Proceedings of European Conference on Parallel Processing*.

[184] Andrews Sobral, Christopher G. Baker, Thierry Bouwmans, and El-hadi Zahzah. 2014. Incremental and multi-feature tensor subspace learning applied for background modeling and subtraction. In *Proceedings of ICIAR*.
[185] Victor Solo and Xuan Kong. 1994. *Adaptive Signal Processing Algorithms: Stability and Performance*. Prentice-Hall, Inc.
[186] Qingquan Song, Xiao Huang, Hancheng Ge, James Caverlee, and Xia Hu. 2017. Multi-aspect streaming tensor completion. In *Proceedings of KDD*.
[187] Nathan Srebro and Adi Shraibman. 2005. Rank, trace-norm and max-norm. In *Proceedings of International Conference on Computational Learning Theory*.
[188] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence* 2009 (2009), Article 4.
[189] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. 2006. Beyond streams and graphs: Dynamic tensor analysis. In *Proceedings of KDD*.
[190] Jimeng Sun, Dacheng Tao, Spiros Papadimitriou, Philip S. Yu, and Christos Faloutsos. 2008. Incremental tensor analysis: Theory and applications. In *Proceedings of TKDD*.
[191] Panagiotis Symeonidis. 2016. Matrix and tensor decomposition in recommender systems. In *Proceedings of the 10th ACM Conference on Recommender Systems*.
[192] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. 2008. Tag recommendations based on tensor dimensionality reduction. In *Proceedings RecSys*.
[193] Koh Takeuchi and Naonori Ueda. 2016. Graph regularized non-negative tensor completion for spatio-temporal data analysis. In *Proceedings of the 2nd International Workshop on Smart*.
[194] Huachun Tan, Bin Cheng, Jianshuai Feng, Guangdong Feng, Wuhong Wang, and Yu-Jin Zhang. 2013. Low-n-rank tensor recovery based on multi-linear augmented Lagrange multiplier method. *Neurocomputing* 119 (2013), 144–152.
[195] Huachun Tan, Bin Cheng, Wuhong Wang, Yu-Jin Zhang, and Bin Ran. 2014. Tensor completion via a multi-linear low-n-rank factorization model. *Neurocomputing* 133 (2014), 161–169.
[196] Huachun Tan, Jianshuai Feng, Guangdong Feng, Wuhong Wang, and Yu-Jin Zhang. 2013. Traffic volume data outlier recovery via tensor model. *Mathematical Problems in Engineering* 2013 (2013), Article 164810.
[197] Huachun Tan, Yuankai Wu, Guangdong Feng, Wuhong Wang, and Bin Ran. 2013. A new traffic prediction method based on dynamic tensor completion. *Procedia-Social and Behavioral Sciences* 96 (2013), 2431–2442.
[198] Jinhui Tang, Xiangbo Shu, Guo-Jun Qi, Zechao Li, Meng Wang, Shuicheng Yan, and Ramesh Jain. 2017. Tri-clustered tensor completion for social-aware image tag refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 8 (2017), 1662–1674.
[199] Giorgio Tomasi and Rasmus Bro. 2005. PARAFAC and missing values. *Chemometrics and Intelligent Laboratory Systems* 75, 2 (2005), 163–180.
[200] Ryota Tomioka, Kohei Hayashi, and Hisashi Kashima. 2010. Estimation of low-rank tensors via convex optimization. arXiv:1010.0789.
[201] Ryota Tomioka, Taiji Suzuki, Kohei Hayashi, and Hisashi Kashima. 2011. Statistical performance of convex tensor decomposition. In *Proceedings of NIPS*.
[202] Joel A. Tropp and Anna C. Gilbert. 2007. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory* 53, 12 (2007), 4655–4666.
[203] Ledyard R. Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* 31, 3 (1966), 279–311.
[204] M. Vasilescu and Demetri Terzopoulos. 2002. Multilinear analysis of image ensembles: TensorFaces. In *Proceedings of the 7th European Conference on Computer Vision-Part I (ECCV'02)*.
[205] Nico Vervliet, Otto Debals, and Lieven De Lathauwer. 2017. *Canonical Polyadic Decomposition of Incomplete Tensors with Linearly Constrained Factors*. Technical Report. Technical Report 16–172, ESAT–STADIUS, KU Leuven, Belgium.
[206] B. Walczak and D. L. Massart. 2001. Dealing with missing data: Part I. *Chemometrics and Intelligent Laboratory Systems* 58, 1 (2001), 15–27.
[207] Hua Wang, Feiping Nie, and Heng Huang. 2014. Low-rank tensor completion with spatio-temporal consistency. In *Proceedings of AAAI*.
[208] Yichen Wang, Robert Chen, Joydeep Ghosh, Joshua C. Denny, Abel Kho, You Chen, Bradley A. Malin, and Jimeng Sun. 2015. Rubik: Knowledge guided tensor factorization and completion for health data analytics. In *Proceedings of KDD*.
[209] Yilun Wang, Yu Zheng, and Yexiang Xue. 2014. Travel time estimation of a path using sparse trajectories. In *Proceedings of KDD*.
[210] Kishan Wimalawarne, Makoto Yamada, and Hiroshi Mamitsuka. 2017. Convex coupled matrix and tensor completion. arXiv:1705.05197.

[211] Dong Xia and Ming Yuan. 2017. On polynomial time methods for exact low rank tensor completion. *Foundations of Computational Mathematics* (2017), to be appear.
[212] Houping Xiao, Yaliang Li, Jing Gao, Fei Wang, Liang Ge, Wei Fan, Long H. Vu, and Deepak S. Turaga. 2015. Believe it today or tomorrow? Detecting untrustworthy information from dynamic multi-source data. In *Proceedings of SDM*.
[213] Biao Xiong, Qiegen Liu, Jiaojiao Xiong, Sanqian Li, Shanshan Wang, and Dong Liang. 2018. Field-of-experts filters guided tensor completion. *IEEE Transactions on Multimedia* 20, 9 (2018), 2316–2329.
[214] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G. Carbonell. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of ICDM*.
[215] Yangyang Xu, Ruru Hao, Wotao Yin, and Zhixun Su. 2013. Parallel matrix factorization for low-rank tensor completion. *arXiv:1312.1254*.
[216] Yangyang Xu and Wotao Yin. 2013. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences* 6, 3 (2013), 1758–1789.
[217] Yangyang Xu, Wotao Yin, Zaiwen Wen, and Yin Zhang. 2012. An alternating direction algorithm for matrix completion with nonnegative factors. *Frontiers of Mathematics in China* 7, 2 (2012), 365–384.
[218] Yuning Yang, Yunlong Feng, and Johan AK Suykens. 2015. A rank-one tensor updating algorithm for tensor completion. *IEEE Signal Processing Letters* 22, 10 (2015), 1633–1637.
[219] Kenan Y. Yılmaz, Ali T. Cemgil, and Umut Simsekli. 2011. Generalised coupled tensor factorisation. In *Proceedings of NIPS*.
[220] Shen Yin and Okyay Kaynak. 2015. Big data for modern industry: Challenges and trends [point of view]. *Proc. IEEE* 103, 2 (2015), 143–146.
[221] Jiaxi Ying, Hengfa Lu, Qingtao Wei, Jian-Feng Cai, Di Guo, Jihui Wu, Zhong Chen, and Xiaobo Qu. 2017. Hankel matrix nuclear norm regularized tensor completion for N-dimensional exponential signals. *IEEE Transactions on Signal Processing* 65, 14 (2017), 3702–3717.
[222] Tatsuya Yokota, Burak Erem, Seyhmus Guler, Simon K. Warfield, and Hidekata Hontani. 2018. Missing slice recovery for tensors using a low-rank model in embedded space. In *Proceedings of CVPR*.
[223] Tatsuya Yokota and Hidekata Hontani. 2017. Simultaneous visual data completion and denoising based on tensor rank and total variation minimization and its primal-dual splitting algorithm. In *Proceedings of CVPR*.
[224] Tatsuya Yokota and Hidekata Hontani. 2018. Simultaneous tensor completion and denoising by noise inequality constrained convex optimization. arXiv:1801.03299.
[225] Tatsuya Yokota, Qibin Zhao, and Andrzej Cichocki. 2016. Smooth PARAFAC decomposition for tensor completion. *IEEE Transactions on Signal Processing* 64, 20 (2016), 5423–5436.
[226] Ji Oh Yoo, Arvind Ramanathan, and Christopher J. Langmead. 2010. *Pytensor: A Python Based Tensor Library*. Technical Report. Technical Report CMU-CS-10-102, Carnegie Mellon University.
[227] Hsiang-Fu Yu, Cho-Jui Hsieh, Si Si, and Inderjit Dhillon. 2012. Scalable coordinate descent approaches to parallel matrix factorization for recommender systems. In *Proceedings of the 12th International Conference on Data Mining (ICDM'12)*. IEEE, 765–774.
[228] Rose Yu, Dehua Cheng, and Yan Liu. 2015. Accelerated online low-rank tensor learning for multivariate spatio-temporal streams. In *Proceedings of ICML*.
[229] Longhao Yuan, Chao Li, Danilo Mandic, Jianting Cao, and Qibin Zhao. 2018. Rank minimization on tensor ring: A new paradigm in scalable tensor decomposition and completion. arXiv:1805.08468.
[230] Ming Yuan and Cun-Hui Zhang. 2017. Incoherent tensor norms and their applications in higher order tensor completion. *IEEE Transactions on Information Theory* 63, 10 (2017), 6753–6766.
[231] Ming Yuan and Cun-Hui Zhang. 2016. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics* 16, 4 (2016), 1031–1068.
[232] Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. 2014. *Social Media Mining: An Introduction*. Cambridge University Press.
[233] Zemin Zhang and Shuchin Aeron. 2017. Exact tensor completion using t-SVD. *IEEE Transactions on Signal Processing* 65, 6 (2017), 1511–1526.
[234] Zemin Zhang, Gregory Ely, Shuchin Aeron, Ning Hao, and Misha Kilmer. 2014. Novel methods for multilinear data completion and de-noising based on tensor-SVD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[235] Longhao Yuan Zhao, Jianting Cao, et al. 2018. High-dimension tensor completion via gradient-based optimization under tensor-train format. arXiv:1804.01983.
[236] Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. 2015. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 9 (2015), 1751–1763.

[237] Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. 2015. Bayesian sparse tucker models for dimension reduction and tensor completion. arXiv:1505.02343.

[238] Qibin Zhao, Guoxu Zhou, Liqing Zhang, Andrzej Cichocki, and Shun-ichi Amari. 2016. Robust Bayesian tensor factorization for incomplete multiway data. *IEEE Transactions on Neural Networks and Learning Systems* 27, 4 (2016), 736–748.

[239] Vincent Wenchen Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Proceedings of AAAI*.

[240] Shuo Zhou, Nguyen Xuan Vinh, James Bailey, Yunzhe Jia, and Ian Davidson. 2016. Accelerating online CP decompositions for higher order tensors. In *Proceedings of KDD*.

[241] Tengfei Zhou, Hui Qian, Zebang Shen, Chao Zhang, and Congfu Xu. 2017. Tensor completion with side information: A riemannian manifold approach. In *Proceedings of IJCAI*.

[242] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. 2008. Large-scale parallel collaborative filtering for the netflix prize. In *International Conference on Algorithmic Applications in Management*. Springer, 337–348.

[243] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J. Smola. 2010. Parallelized stochastic gradient descent. In *Proceedings of NIPS*.

Received November 2017; revised July 2018; accepted September 2018