Journal of Machine Learning Research 23 (2022) 1-49 Submitted 9/20; Published 9/22

Regularized and Smooth Double Core Tensor Factorization for Heterogeneous Data

Davoud Ataee Tarzanagh[⋆] TARZANAGH@UFL.EDU
George Michailidis[†] GMICHAIL@UFL.EDU
[⋆]*Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA*
[†]*Department of Statistics, University of Florida, Gainesville, FL 32603, USA*
[⋆,†]*University of Florida Informatics Institute, Gainesville, FL 32611, USA*

Editor: Jie Peng

Abstract

We introduce a general tensor model suitable for data analytic tasks for *heterogeneous* datasets, wherein there are joint low-rank structures within groups of observations, but also discriminative structures across different groups. To capture such complex structures, a double core tensor (DCOT) factorization model is introduced together with a family of smoothing loss functions. By leveraging the proposed smoothing function, the model accurately estimates the model factors, even in the presence of missing entries. A linearized ADMM method is employed to solve regularized versions of DCOT factorizations, that avoid large tensor operations and large memory storage requirements. Further, we establish theoretically its global convergence, together with consistency of the estimates of the model parameters. The effectiveness of the DCOT model is illustrated on several real-world examples including image completion, recommender systems, subspace clustering, and detecting modules in heterogeneous Omics multi-modal data, since it provides more insightful decompositions than conventional tensor methods.

Keywords: Double core tensor factorization, heterogeneity, smoothing loss functions, regularization, ADMM

1. Introduction

Tensor factorizations have received increasing attention over the last decade, due to new technical developments, as well as novel applications (Kolda and Bader, 2009; Cichocki et al., 2015; Bi et al., 2020). Popular decompositions include Tucker (Tucker, 1964), Canonical Polyadic (CP) (Carroll and Chang, 1970), higher-order SVD (HOSVD) (De Lathauwer et al., 2000b), tensor train (TT) (Oseledets, 2011), and tensor SVD (t-SVD) (Kilmer et al., 2013). Nevertheless, there is limited knowledge about the properties of the Tucker and CP ranks; further, computing such ranks has been shown to be NP-complete (Håstad, 1990). The ill-posedness of the best low-rank approximation of a tensor was investigated in De Silva and Lim (2008), while upper and lower bounds for tensor ranks have been studied in Alexeev et al. (2011). In fact, determining or even bounding the rank of an arbitrary tensor is quite difficult in contrast to the matrix rank (Allman et al., 2013).

In many data analysis applications where tensor decompositions are extensively used, the multi-dimensional data exhibit (i) heterogeneity, (ii) missing values, and (iii) sparse representations. The literature to date has addressed the 2nd and 3rd issues, as briefly discussed next. Missing values are ubiquitous in link prediction, recommender systems, chemometrics, image and video analytics applications. To that end, tensor completion methods were developed to address this issue (Liu et al., 2012; Kressner et al., 2014; Zhao et al., 2015; Zhang et al., 2014; Song et al., 2017; Tarzanagh and Michailidis, 2018b). The standard assumption underpinning such methods is that entries are missing at random and that the data admit low-rank decompositions. However, on many occasions additional *regularity* information is available, which may aid in improving the accuracy of tensor completion methods, especially in the presence of the large number of missing entries. For example, (Narita et al., 2012) proposed two regularization methods called "within-mode regularization" and "cross-mode regularization", to incorporate auxiliary regularity information in the tensor completion problems. The key idea is to construct within-mode or cross-mode regularity matrices and incorporate them as smooth regularizers and then combine them with a Tucker decomposition to solve the tensor completion problem. A similar idea was also explored in Bahadori et al. (2014); Chen and Hsu (2014); Ge et al. (2016).

Multi-way data often admit sparse representations. Due to the equivalence of the constrained Tucker model and the Kronecker representation of a tensor, the latter can be represented by separable sparse Kronecker dictionaries. A number of Kronecker-based dictionary learning methods have been proposed in literature (Hawe et al., 2013; Qi et al., 2018; Shakeri et al., 2018; Bahri et al., 2018) and associated algorithms. Further, recent work (Shakeri et al., 2018) shows that the sample complexity of dictionary learning for tensor data can be significantly lower than that for unstructured data, also supported by empirical evidence (Qi et al., 2018).

However, in a number of applications, *heterogeneity* is also present in the data. For example, in context-aware recommender systems that predict users' preferences, the user base exhibits heterogeneity due to different background and other characteristics. Note that such information can be a priori extracted from the available data. Similarly, image data exhibit heterogeneity due to differences in lighting and posing, which again can be extracted a priori from available metadata and utilized at analysis time. Analogous issues also are present in time varying data, wherein strong correlations can be seen across subsets of time points. A number of motivating examples are discussed in detail in Section 2.3, and how this paper addresses heterogeneity by adding an additional core in the tensor decomposition and applying a new tensor smoothing function. Note that a standard low-rank factorization of the tensor data would not suffice, since the extracted factors would not accurately reflect the joint structure across modes. Hence, to address heterogeneity in tensor factorizations, we introduce a novel decomposition of the core tensor into global homogeneous and local (subject–specific) heterogeneous cores. The latter encodes a priori available information on the *presence* and *structure* of heterogeneity on the datasets under consideration.

Hence, the *key contributions* of this work are:

I. The development of a novel *supervised* tensor decomposition, coined *Double Core Tensor Decomposition* (DCOT), wherein the core tensor comprises of the superposition of *homogeneous* and *heterogeneous* (subject specific) cores. This decomposition captures

local structure present due to variations in similarities across subjects/objects in the data.

II. The DCOT model is enhanced with a new tensor smoothing loss function. Specifically, a similarity function is introduced to capture neighborhood information from the data tensor to improve the accuracy of the decompositions, as well as the convergence rate of the algorithm employed for obtaining the decomposition. We show both theoretical and computational advantages of the proposed smoothing technique in comparison to the generalized CP (GCP) models (Bi et al., 2018; Hong et al., 2019).

III. A new linearized ADMM method for the DCOT model is developed that can handle both non-convex constraints and objectives and its global convergence under the posited tensor smoothing loss function is established. To the best of our knowledge, despite the wide use and effectiveness of ADMM for tensor factorization tasks (Liu et al., 2012; Zhao et al., 2015; Chen et al., 2013; Wang et al., 2015; Bahri et al., 2018) its global convergence does not seem to be available for tensor problems.

Finally, we illustrate the implementation of DCOT and the speed and robustness of the proposed linearized ADMM algorithm on a number of synthetic datasets, as well as analytic tasks involving large scale heterogeneous tensor data.

1.1 Related Literature

This work is related to a broad range of literature on tensor analysis. For example, tensor factorization approaches focus on the extraction of low-rank structures from noisy tensor observations (Zhang and Golub, 2001; Richard and Montanari, 2014; Anandkumar et al., 2014). Correspondingly, a number of methods have been proposed and analyzed under either deterministic or random Gaussian noise designs, such as maximum likelihood estimation (Richard and Montanari, 2014), HOSVD (De Lathauwer et al., 2000b), and higher-order orthogonal iteration (HOOI) (De Lathauwer et al., 2000a). Since non-Gaussian-valued tensor data also commonly appear in practice, Chi and Kolda (2012); Hong et al. (2019) considered the generalized tensor decomposition and introduced computational efficient algorithms. However, theoretical guarantees for many of these procedures and the statistical limits of the smooth tensor decomposition still remain open.

Our proposed framework includes the topic of tensor compression and dictionary learning. Various methods, such as convex regularization (Tomioka and Suzuki, 2013; Raskutti et al., 2019), alternating minimization (Zhou et al., 2013; Hawe et al., 2013; Qi et al., 2018; Han et al., 2020; Liu et al., 2012; Tarzanagh and Michailidis, 2018b), and (adaptive) gradient methods (Han et al., 2020; Kolda and Hong, 2020; Nazari et al., 2019, 2020) were introduced and studied. In addition, tensor block models (Smilde et al., 2000), supervised tensor learning (Tao et al., 2005; Wu et al., 2013; Lock and Li, 2018), multi-layer tensor factorization (Bi et al., 2018; Tang et al., 2020), coupled matrix and tensor factorizations (Banerjee et al., 2007; Yılmaz et al., 2011; Acar et al., 2011) are important topics in tensor analysis and have attracted significant attention in recent years. Departing from the existing results, this paper, to the best of our knowledge, is the first to give a unified treatment for a broad range of smooth and heterogeneous tensor estimation problems with both statistical optimality and computational efficiency.

This work is also related to a substantial body of literature on structured matrix factorization, wherein the goal is to estimate a low-rank matrix based on a limited number of observations. Specific examples on this topic include group-specific matrix factorization (Lock et al., 2013; Bi et al., 2017), local matrix factorization (Lee et al., 2013), and smooth matrix decomposition (Dai et al., 2019). Despite similarities of our consistency analysis to Dai et al. (2019), their results cannot be directly generalized to tensor problems for many reasons. First, many basic matrix concepts or methods cannot be directly generalized to high-order ones (Hillar and Lim, 2013). Naive generalization of matrix concepts such as kernels, operator norm, and singular values are possible, but most often computationally NP-hard. Second, tensors have more complicated algebraic structure than matrices. As what we will illustrate later, one has to simultaneously handle all factors matrices and the core tensors with distinct dimensions in the consistency and global convergence analyses. To this end, we develop new technical tools for tensor algebra and Kronecker smoothing functions; see, e.g., Definition 4. Additional technical issues related to generalized tensor estimation and the connections of our consistency bounds with prior results in the literature are addressed in Section 4.

The remainder of the paper is organized as follows: the DCOT formulation is presented in Section 2 together with illustrative motivating examples. The linearized ADMM algorithm and its convergence properties are presented in Section 3. The numerical performance of the DCOT model together with applications are discussed in Section 5. Section 6 concludes the paper. Proofs and other technical results are delegated to the Appendix.

Notation. Any notation is defined when it is used, but for reference the reader may also find it summarized in Table 6.

2. A Double Core Tensor Factorization (DCOT)

We start by introducing the DCOT model.

Definition 1 (DCOT) *Given an N-way tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with M subjects (units) each containing m_π subgroups for $m = 1, \cdots, M$, its* DCOT *decomposition is given by*

$$\begin{aligned}\boldsymbol{\mathcal{Z}} &= \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} \left(g_{r_1 \cdots r_N} + h_{r_1 \cdots r_N}\right) \mathbf{u}_{r_1}^{(1)} \circ \cdots \circ \mathbf{u}_{r_N}^{(N)} \\ &= (\boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{H}}) \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)}, \end{aligned} \tag{1}$$

where $\mathbf{U}^{(n)} = [\mathbf{u}_1^{(n)}, \mathbf{u}_2^{(n)}, \cdots, \mathbf{u}_{R_n}^{(n)}] \in \mathbb{R}^{I_n \times R_n}$, is the n-th factor matrix consisting of latent components $\mathbf{u}_r^{(n)}$; $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ is a global core tensor reflecting the connections (or links) between the latent components and factor matrices; and $\boldsymbol{\mathcal{H}}$ is another core tensor reflecting the joint connections between the latent components in each subject. Specifically, for each subject $m \in [M]$, we have

$$\boldsymbol{\mathcal{H}}_{m_1} = \boldsymbol{\mathcal{H}}_{m_2} = \cdots = \boldsymbol{\mathcal{H}}_{m_\pi}.$$

In Definition 1, the N-tuple $(R_1, R_2, \ldots, R_N)$ with $R_n = \operatorname{rank}(\mathbf{Z}_{(n)})$ is called the multilinear rank of $\boldsymbol{\mathcal{Z}}$. For a core tensor of minimal size, R_1 is the column rank (the dimension of

the subspace spanned by mode-1 fibers), R_2 is the row rank (the dimension of the subspace spanned by mode-2 fibers), and so on. An important difference from the matrix case is that the values of $R_1, R_2, \cdots, R_N$ can be different for $N \geq 3$. Note that similar to the Tucker decomposition (Tucker, 1964), DCOT factorization is said to be independent, if each of the factor matrices has full column rank; a DCOT decomposition is said to be orthonormal if each of the factor matrices has orthonormal columns. We also note that decomposition (1) can be expressed in a matrix form as:

$$\mathbf{Z}_{(n)} \quad = \quad \mathbf{U}^{(n)}(\mathbf{G}_{(n)} + \mathbf{H}_{(n)})(\bigotimes_{k \neq n} \mathbf{U}^{(k)})^{\top}. \tag{2}$$

The DCOT model formulation provides a generic tensor decomposition that encompasses many other popular tensor decomposition models. Indeed, when $\boldsymbol{\mathcal{H}} = 0$ and $\mathbf{U}^{(n)}$ for $n = 1, 2, \cdots, N$ are orthogonal, (1) corresponds to HOSVD. The CP decomposition can also be considered as a special case of the DCOT model with super-diagonal core tensors.

In the DCOT model, we assume that subjects can be categorized into subgroups, where tensor components within the same subgroup share similar characteristics and are dependent on each other. For subgrouping, we can incorporate prior information. For example, in recommender system we may use users' demographic information, item categories and functionality, and contextual similarity. If this kind of information is not available, one can use the missing pattern of the tensor data, or the number of records from each user and on each item (Salakhutdinov et al., 2007). In more general situations, clustering methods such as the k-means may be used to determine the subgroups (Wang, 2010; Fang and Wang, 2012).

Next, we propose an estimation method associated with the DCOT model. Let $\boldsymbol{\mathcal{X}}$ be a data tensor that admits a DCOT decomposition and $F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}})$ be a tensor smoothing loss function (introduced in Section 2.1) that depends on an unknown parameter $\boldsymbol{\mathcal{Z}}$ and regulated by a smoothing function s^h (details discussed in Section 2.1). To estimate $\boldsymbol{\mathcal{Z}}$ from data, we propose the "DCOT" estimator given by

$$\hat{\boldsymbol{\mathcal{Z}}} := \left(\hat{\boldsymbol{\mathcal{G}}} + \hat{\boldsymbol{\mathcal{H}}}\right) \times_1 \hat{\mathbf{U}}^{(1)} \times_2 \hat{\mathbf{U}}^{(2)} \cdots \times_N \hat{\mathbf{U}}^{(N)}, \tag{3}$$

where $\hat{\boldsymbol{\mathcal{Z}}}$ is determined by solving the following penalized optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\mathcal{T}}} \quad & F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) + \lambda_1 J_1(\boldsymbol{\mathcal{G}}) + \lambda_2 J_2(\boldsymbol{\mathcal{H}}) + \sum_{n=1}^{N} \lambda_{3,n} J_{3,n}(\mathbf{U}^{(n)}), \\ \text{s.t.} \quad & \boldsymbol{\mathcal{Z}} = \left(\boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{H}}\right) \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)}, \qquad \boldsymbol{\mathcal{T}} \in \Gamma(\boldsymbol{\mathcal{T}}). \end{aligned} \tag{4}$$

In the formulation of the problem, $\boldsymbol{\mathcal{T}} = (\boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{H}}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(N)})$ denotes the collection of optimization variables; $\{J_1(.), J_2(.), J_{3,1}(.), J_{3,2}(.), \cdots, J_{3,N}(.)\}$ are penalty functions; $\{\lambda_1, \lambda_2, \lambda_{3,1}, \lambda_{3,2}, \cdots, \lambda_{3,N}\}$ are penalty tuning parameters; and $\Gamma(\boldsymbol{\mathcal{T}})$ is the parameter space for $\boldsymbol{\mathcal{T}}$.

Throughout, we impose the following set of assumptions on Problem (4).

Assumption A (i) $J_1 : \mathbb{R}^{R_1 \times R_2 \cdots \times R_N} \to (-\infty, \infty]$, $J_2 : \mathbb{R}^{R_1 \times R_2 \cdots \times R_N} \to (-\infty, \infty]$, and $J_{3,n} : \mathbb{R}^{N_n \times R_n} \to (-\infty, \infty]$ are proper and lower semi-continuous such that $\inf_{\mathbb{R}^{R_1 \times R_2 \cdots \times R_N}} J_1 >$

$-\infty$, $\inf_{\mathbb{R}^{R_1 \times R_2 \cdots \times R_N}} J_2 > -\infty$, and $\inf_{\mathbb{R}^{I_n \times R_n}} J_{3,n} > -\infty$ for $n = 1, 2, \cdots, N$.
(ii) $F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) : \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \to \mathbb{R}$ is differentiable and $\inf_{\mathbb{R}^{I_1 \times I_2 \times, \cdots, \times I_N}} F > -\infty$.
(iii) The gradients $\nabla F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}})$ is Lipschitz continuous with moduli L_F, i.e.,

$$\|\nabla F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}^1) - \nabla F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}^2)\|_F^2 \leq L_F \|\boldsymbol{\mathcal{Z}}^1 - \boldsymbol{\mathcal{Z}}^2\|_F^2, \quad \text{for all} \quad \boldsymbol{\mathcal{Z}}^1, \boldsymbol{\mathcal{Z}}^2.$$

We need to recall the fundamental proximal map which is at the heart of the DCOT algorithm. Given a proper and lower semicontinuous function $J : \mathbb{R}^d \to (-\infty, \infty]$, the proximal mapping associated with J is defined by

$$\text{prox}_t^J (p) := \text{argmin} \left\{ J(q) + \frac{t}{2} \|q - p\|^2 : \ q \in \mathbb{R}^d \right\}, \quad (t > 0). \tag{5}$$

The following result can be found in Rockafellar and Wets (2009); Bolte et al. (2014).

Proposition 2 *Let Assumption A(i) hold. Then, for every $t \in (0, \infty)$, the set $prox_t^J (u)$ is nonempty and compact.*

We note that prox_t^J is a set-valued map. When J is the indicator function of a nonempty and closed set Ω, the proximal map reduces to the projection operator onto Ω. It is also worth mentioning that Assumptions (i)-(iii) make (4) have a solution and make the proposed linearized ADMM well defined. Besides these assumptions, many practical tensor factorization functions including ones provided in Subsection 2.3 satisfy the Kurdyka–Łojasiewicz property (see, Definition 4) which is required to obtain a globally convergent linearized ADMM.

Remark 3 *Note that separate identification of $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{H}}$ is not required; the DCOT estimator is designed to automatically recover the combination $\hat{\boldsymbol{\mathcal{G}}} + \hat{\boldsymbol{\mathcal{H}}}$ that leads to optimal prediction of $\boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{H}}$. Nevertheless, such identification can be beneficial from both a convergence and interpretation viewpoint. To that end, we show in Section 3 that Assumption A provides sufficient conditions to achieve identifiable cores based on a linearized multi-block ADMM approach.*

2.1 Smoothing Loss Functions for DCOT Factorization

In this section, we introduce a new class of loss functions for DCOT factorization that in addition to the unit/subject information reflected in the core $\boldsymbol{\mathcal{H}}$, it incorporates more nuanced information in the form of similarities between tensor fibers for each unit/subject under consideration. To that end, following common approaches in non-parametric statistics (Wand and Jones, 1994; Tibshirani and Hastie, 1987; Fan, 2018; Lee et al., 2013; Dai et al., 2019), we define the smoothing function used in this work. Our proposed smoothing approach is different from these studies, since we rely on joint label information and multi-dimensional kernels.

2.2 A Smoothing Function Based on Tensor Similarity and Tensor Labels

The proposed loss function incorporates both tensor similarity information, as well as subgroup or label information. For example, in dictionary learning problems, in addition

to using similarity information between data points, we also associate label information (0-1 label) with each dictionary item to enforce discriminability in sparse codes during the dictionary learning process (Jiang et al., 2013). In recommender systems, the proposed loss function is constructed based on the closeness between continuous covariates in addition to a user-item specific label tensor (Frolov and Oseledets, 2017; Dai et al., 2019). In many imaging applications, additional variables of interest are available for multiway data objects. For instance, Kumar et al. (2009) provide several attributes for the images in the faces in the Wild database, which describe the individual (e.g., gender and race) or their expression (e.g., smiling/not smiling). It is shown in Lock and Li (2018) that incorporating such additional variables can improve both the accuracy and interpretation of the tensor factorization for imaging applications. =

Definition 4 (Kronecker Similarity) *Given an N-way data tensor $\boldsymbol{\mathcal{X}}$, assume there is additional information on each subject in the data, encoded by an N-way tensor $\boldsymbol{\mathcal{Y}}$. Let $s^h_{i_n,j_n}$ denote pairwise similarities between fibers i_n and j_n of $\boldsymbol{\mathcal{Y}}$. Each $s^h_{i_n,j_n}$ indicates how well fibers of $\boldsymbol{\mathcal{Y}}$ represent fibers of $\boldsymbol{\mathcal{X}}$, i.e., the smaller the value of $s^h_{i_n,j_n}$ is, the better $\boldsymbol{\mathcal{Y}}$ represents $\boldsymbol{\mathcal{X}}$. Under this setting, we define the Kronecker-product similarity as*

$$s^h_{i_1\cdots i_N,j_1\cdots j_N} = s^h_{i_1,j_1}c_{i_1,j_1}\cdots s^h_{i_N,j_N}c_{i_N,j_N}. \tag{6}$$

Here, $h > 0$ is the window size; each $s^h_{i_n,j_n}$ measures the distance between fibers (i_n, j_n) for $n = 1, \cdots, N$; and c_{i_n,j_n} is a label consistent which is set to a value close to 1 if fibers i_n and j_n share the same labels or belong to the same subgroups and close to 0, otherwise. We note that a large value of h implies that $s^h_{i_1\cdots i_N,j_1\cdots j_N}$ has a wide range, while a small h corresponds to a narrow range for $s^h_{i_1\cdots i_N,j_1\cdots j_N}$.

When appropriate vector-space representations of fibers of $\boldsymbol{\mathcal{Y}}$ are given, we can compute similarities using a predefined function. Such functions are the encoding error -$s^h_{i_n,j_n} = K(h^{-1}\|\mathbf{A}\mathbf{y}_{i_n} - \mathbf{y}_{j_n}\|_2)$ for an appropriate $\mathbf{A}$-, the Euclidean distance -$s^h_{i_n,j_n} = K(h^{-1}\|\mathbf{y}_{i_n} - \mathbf{y}_{j_n}\|_2)$-, or a truncated quadratic -$s^h_{i_n,j_n} = \min\{\xi, K(h^{-1}\|\mathbf{y}_{i_n} - \mathbf{y}_{j_n}\|_2)\}$-, where ξ is some constant and K denotes a kernel function (Tibshirani and Hastie, 1987). However, we may be given or can compute similarities without having access to vector-space representations; such instances include edges in a social network graph, subjective pairwise comparisons between images, or similarities between sentences computed via a string kernel. Finally, we may learn similarities by using metric learning methods (Xing et al., 2003; Davis et al., 2007; Elhamifar et al., 2015).

2.2.1 Generalized smoothing tensor loss functions

Next, we define smoothing tensor loss functions by looking at the statistical likelihood of a model for a given data tensor. Assume that we have a parameterized probability density function or probability mass function that gives the likelihood of each entry, i.e.,

$$x_{i_1\cdots i_N} \sim p(x_{i_1\cdots i_N}|\theta_{i_1\cdots i_N}), \quad \text{where} \quad \ell(\theta_{i_1\cdots i_N}) = z_{i_1\cdots i_N}.$$

Here, $x_{i_1\cdots i_N}$ is an observation of a random variable, and $\ell(\cdot)$ is an invertible *link function* that connects the model parameters $z_{i_1\cdots i_N}$ and the corresponding *natural parameters* of the distribution, $\theta_{i_1\cdots i_N}$.

Our goal is to obtain the maximum likelihood estimate $\boldsymbol{\mathcal{Z}}$. Let Ω be an index set of observed tensor components. Assuming that the samples are independent and identically distributed, we can obtain $\boldsymbol{\mathcal{Z}}$ by solving

$$\max_{\boldsymbol{\mathcal{Z}}} \; L(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) \equiv \prod_{(i_1,\cdots,i_N)\in\Omega} p(x_{i_1\cdots i_N}|\theta_{i_1\cdots i_N}), \quad \text{where} \;\; \ell(\theta_{i_1\cdots i_N}) = z_{i_1\cdots i_N}. \tag{7}$$

Working with the log-likelihood, one can easily obtain the following minimization problem

$$\min_{\boldsymbol{\mathcal{Z}}} \Big\{ \widehat{F}(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) = -\frac{1}{\Omega} \sum_{(i_1,\cdots,i_N)\in\Omega} \widehat{f}(x_{i_1\cdots i_N}; z_{i_1\cdots i_N}) \Big\},$$

where

$$\widehat{f}(x_{i_1\cdots i_N}; z_{i_1\cdots i_N}) = \log\big(p(x_{i_1\cdots i_N}|\ell^{-1}(z_{i_1\cdots i_N})\big). \tag{8}$$

In this paper, we propose a novel approach based on the idea of a tensor similarity and tensor labels to improve the prediction performance. Specifically, using the similarity function s^h, we consider the following cost function

$$\min_{\boldsymbol{\mathcal{Z}}} \Big\{ F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) = -\frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} f(s^h, \boldsymbol{\mathcal{X}}; z_{i_1\cdots i_N}) \Big\}, \tag{9}$$

where

$$f(s^h, \boldsymbol{\mathcal{X}}; z_{i_1\cdots i_N}) = \sum_{(j_1,\ldots,j_N)\in\Omega} s^h_{i_1\cdots i_N, j_1\cdots j_N} \widehat{f}(x_{j_1\cdots j_N}; z_{i_1\cdots i_N}) \tag{10}$$

is a smoothing probability density function.

An important feature of this smoothing function is to pool information across each $x_{i_1\cdots i_N}$ through the weights $s^h_{i_1\cdots i_N, j_1\cdots j_N}$ to increase effective sample size and improve prediction accuracy. Next, we present various loss functions corresponding to different types of data; e.g., numerical, binary, and count.

2.2.2 Numerical data

We are concerned with the situation where we have the data tensor $\boldsymbol{\mathcal{X}}$ corrupted by *white noise*. Specifically, we assume that

$$x_{i_1\cdots i_N} = z_{i_1\cdots i_N} + \epsilon_{i_1\cdots i_N} \quad \text{with} \quad \epsilon_{i_1\cdots i_N} \sim \mathcal{N}(0, \sigma^2) \;\; \text{for all} \;\; (i_1, \cdots, i_N) \in \Omega. \tag{11}$$

Here, $\mathcal{N}(\mu, \sigma^2)$ denotes the normal or Gaussian distribution with mean μ and variance σ^2 [1]. It follows from (11) that

$$x_{i_1\cdots i_N} \sim \mathcal{N}(\mu_{i_1\cdots i_N}, \sigma^2) \;\; \text{with} \;\; \mu_{i_1\cdots i_N} = z_{i_1\cdots i_N} \;\; \text{for all} \;\; (i_1, \cdots, i_N) \in \Omega.$$

In this case, the link function between $\mu_{i_1\cdots i_N}$ and $z_{i_1\cdots i_N}$ is the identity, i.e., $\ell(\mu_{i_1\cdots i_N}) = \mu_{i_1\cdots i_N}$. Plugging this link function into (8) yields $\widehat{f}(x_{i_1\cdots i_N}; z_{i_1\cdots i_N}) = (z_{i_1\cdots i_N} - x_{i_1\cdots i_N})^2$. Now, using (10), we obtain

$$f(s^h, \boldsymbol{\mathcal{X}}; z_{i_1\cdots i_N}) = \sum_{(j_1,\ldots,j_N)\in\Omega} s^h_{i_1\cdots i_N, j_1\cdots j_N} (z_{i_1\cdots i_N} - x_{j_1\cdots j_N})^2. \tag{12}$$

1. We assume σ is *constant across all entries*.

2.2.3 Binary data

The standard assumption of a data generating mechanism for such data is the Bernoulli distribution; specifically, a binary random variable $x \in \{0, 1\}$ is Bernoulli distributed with parameter $\rho \in [0, 1]$ if ρ is the probability of obtaining a value of 1 and $(1 - \rho)$ is the probability for obtaining a value of 0. The probability mass function is given by

$$p(x|\rho) = \rho^x (1 - \rho)^{(1-x)}, \quad x \in \{0, 1\}, \qquad x \sim \text{Bernoulli}(\rho). \tag{13}$$

A reasonable model for a binary data tensor $\boldsymbol{\mathcal{X}}$ is

$$x_{i_1 \cdots i_N} \sim \text{Bernoulli}(\rho_{i_1 \cdots i_N}), \quad \text{where} \quad \ell(\rho_{i_1 \cdots i_N}) = z_{i_1 \cdots i_N}. \tag{14}$$

A common option for the link function $\ell(\rho)$ is to work with the log-odds, i.e.,

$$\ell(\rho) = \log(\frac{\rho}{1 - \rho}). \tag{15}$$

Substituting the link function (15) into (8), gives $\widehat{f}(x_{i_1 \cdots i_N}; z_{i_1 \cdots i_N}) = \log(1 + e^{z_{i_1 \cdots i_N}}) - x_{i_1 \cdots i_N} z_{i_1 \cdots i_N}$. Now, using (10), we get the following smoothing tenor function

$$f(s^h, \boldsymbol{\mathcal{X}}; z_{i_1 \cdots i_N}) = \sum_{(j_1, \ldots, j_N) \in \Omega} s^h_{i_1 \cdots i_N, j_1 \cdots j_N} \big(\log(1 + e^{z_{i_1 \cdots i_N}}) - x_{j_1 \cdots j_N} z_{i_1 \cdots i_N} \big),$$

where $x_{j_1 \cdots j_N} \in \{0, 1\}$ and the associated probability is $\rho = e^{z_{j_1 \cdots j_N}} / (1 + e^{z_{j_1 \cdots j_N}})$.

2.2.4 Count data

For count data, it is common to model them as a Poisson distribution. The probability mass function for a Poisson distribution with mean λ is given by

$$p(x|\lambda) = \frac{e^{-\lambda} \lambda^x}{x!} \quad \text{for} \quad x \in \mathbb{N}. \tag{16}$$

If we use the identity link function, i.e., $\ell(\lambda) = \lambda$ and substitute (16) into (8), we obtain $\widehat{f}(x_{i_1 \cdots i_N}; z_{i_1 \cdots i_N}) = z_{i_1 \cdots i_N} - x_{i_1 \cdots i_N} \log z_{i_1 \cdots i_N}$. Now, it follows from (10) that

$$f(s^h, \boldsymbol{\mathcal{X}}; z_{i_1 \cdots i_N}) = \sum_{(j_1, \ldots, j_N) \in \Omega} s^h_{i_1 \cdots i_N, j_1 \cdots j_N} \big(z_{i_1 \cdots i_N} - x_{j_1 \cdots j_N} \log z_{i_1 \cdots i_N} \big), \tag{17}$$

where $x_{j_1 \cdots j_N} \in \mathbb{N}$ and $z_{i_1 \cdots i_N} \geq 0$.

2.2.5 Positive continuous data

There are several distributions for handling nonnegative continuous data: Gamma, Rayleigh, and even Gaussian with nonnegativity constraints. Next, we consider the Gamma distribution which is appropriate for strictly positive data. For $x > 0$, the probability density function is given by

$$p(x|t, \theta) = \frac{x^{t-1} e^{-x/\theta}}{\Gamma(t)\, \theta^t}, \tag{18}$$

where the parameters t and θ are positive real quantities as is the variable x and $\Gamma(\cdot)$ is the Gamma function.

A common choice for the link function is $\ell(t,\theta) = \theta/t$ which induces a positivity constraint on $z_{j_1\cdots j_N}$. Assume t is constant across all entries. Plugging the functions p and ℓ into (8) and removing the constant terms yields $\widehat{f}(x_{i_1\cdots i_N}; z_{i_1\cdots i_N}) = \log(z_{i_1\cdots i_N}) + x_{i_1\cdots i_N}/z_{i_1\cdots i_N}$. Hence, the smoothing loss function is defined by

$$f(s^h, \boldsymbol{\mathcal{X}}; z_{i_1\cdots i_N}) = \sum_{(j_1,\ldots,j_N)\in\Omega} s^h_{i_1\cdots i_N, j_1\cdots j_N}\big(\log(z_{i_1\cdots i_N}) + x_{j_1\cdots j_N}/z_{i_1\cdots i_N}\big), \tag{19}$$

where $x_{j_1\cdots j_N}$ and $z_{i_1\cdots i_N}$ are both positive. In practice, we use $z_{i_1\cdots i_N} \geq 0$ and replace $z_{i_1\cdots i_N}$ with $z_{i_1\cdots i_N} + \epsilon$ (with small ϵ) in the loss function (19).

2.3 Motivating Examples and Applications of the DCOT Model

Next, we discuss a number of motivating examples for DCOT and the associated smoothing loss function.

2.3.1 Context-Aware Recommender Systems

Recommender systems predict users' preferences across a set of items based on large past usage data, while also leveraging information from similar users. In multilayer recommender systems, a tensor based analysis is beneficial due to its flexibility to accommodate contextual information from data, and is also regarded as effective in developing context-aware recommender systems (CARS) (Adomavicius and Tuzhilin, 2011; Frolov and Oseledets, 2017; Bi et al., 2018, 2020; Zhang et al., 2020). Besides user and item information available in traditional recommender systems (Lang, 1995; Verbert et al., 2012; Bi et al., 2017), multilinear recommender systems also use additional contextual variables, including geolocation data, time stamps, store information, etc. Although CARS are capable of utilizing such additional information and thus furnishing more accurate recommendations, they are also hampered by the so-called "cold-start" problem, wherein not sufficient information is available on new users, items or contexts. To address these issues, we propose a new tensor model which incorporates smoothing loss functions and can accommodate heterogeneity across observation groups. More specifically, we consider the objective function

$$F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) + \lambda_1\|\boldsymbol{\mathcal{G}}\|_F^2 + \lambda_2\|\boldsymbol{\mathcal{H}}\|_F^2 + \lambda_3\|\mathbf{U}\|_F^2, \tag{20}$$

where λ_i for $i = 1, 2, 3$ denote regularization parameters and $\mathbf{U} = \bigotimes_{n\in[N]} \mathbf{U}^{(n)}$. Other regularization methods include, but are not limited to, the ℓ_0- and ℓ_1-penalty for sparse low-rank pursuit.

The function (20) enables pooling information from neighboring $(i_1, \cdots, i_N)$ points, through similarity function s^h. Further, it addresses satisfactorily the "cold start" problem, by leveraging information from similar users in similar contextual settings. Finally, the issue of missing data in a non-ignorable fashion can be easily addressed through appropriately constructed neighborhoods and similarities (6).

2.3.2 Discriminative and Separable Dictionary Learning

We aim to leverage the supervised information (i.e. subjects) of input signals to learn a discriminative and separable dictionary. Assume the available data are organized in a K-order tensor $\boldsymbol{\mathcal{X}}_n \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$. According to the separable dictionary models (Hawe et al., 2013), given *coordinate dictionaries* $\mathbf{U}^{(k)} \in \mathbb{R}^{I_k \times R_k}$, *coefficient tensors* $\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{H}} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_K}$, and a *noise tensor* $\boldsymbol{\mathcal{E}}_n$, we can express $\mathbf{x}_n = \text{vec}(\boldsymbol{\mathcal{X}}_n)$ as

$$\mathbf{x}_n = \Big(\bigotimes_{n \in [N]} \mathbf{U}^{(n)} \Big) (\mathbf{g}_n + \mathbf{h}_n) + \varepsilon_n, \tag{21}$$

where $\mathbf{g}_n = \text{vec}(\boldsymbol{\mathcal{G}}_n)$, $\mathbf{h}_n = \text{vec}(\boldsymbol{\mathcal{H}}_n)$ and $\varepsilon_n = \text{vec}(\boldsymbol{\mathcal{E}}_n)$.

Let

$$I = \prod_{k \in [K]} I_k \quad \text{and} \quad R = \prod_{k \in [K]} R_k.$$

By concatenating N noisy observations $\{\mathbf{x}_n\}_{n=1}^N$ that are realizations from the data generating process posited in (21) into $\mathbf{X} \in \mathbb{R}^{I \times N}$, we obtain the following discriminative dictionary learning model

$$F(s^h, \mathbf{X}; \mathbf{Z}) + \lambda_1 \|\mathbf{G}\|_1 + \lambda_2 \|\mathbf{H}\|_1 + \lambda_3 \sum_{m=1}^{M} \sum_{\kappa=1}^{\pi} \|\mathbf{H}_{m_\kappa}\|_F, \tag{22}$$

where $\mathbf{Z} = \mathbf{U}(\mathbf{G} + \mathbf{H})$; $\mathbf{U} = \bigotimes_{k \in [K]} \mathbf{U}^{(k)}$ is the basis matrix; $\mathbf{G}$ and $\mathbf{H}$ are coefficient matrices; and λ_1–λ_3 are regularization parameters.

Note that in (22), we consider a *sparse group lasso* penalty for the structured core $\boldsymbol{\mathcal{H}}$. This penalty yields solutions that are sparse at both the group and individual feature levels for all subjects $m = 1, 2, \cdots, M$.

2.3.3 Image Analytics

On many occasions, the available image dataset contains multiple shots of the same subject, as is the case in the CMU faces database (Sim et al., 2002). As an illustration, using 30 subjects from the data base and extracting 11 poses under 21 lighting conditions, we end up with a tensor comprising of $6930 = 30 \times 11 \times 21$ images, of 32×32 dimension each. Since each subject remains the same under different illuminations for the same pose, and there are also a number of other subjects with the same pose, we consider the following partition of the core tensor $\boldsymbol{\mathcal{H}}$

$$\boldsymbol{\mathcal{H}}_i = \boldsymbol{\mathcal{H}}(i, :, :, :), \quad \text{for} \quad i = 1, 2, \ldots, 30.$$

Further, if the resulting tensor is missing certain illuminations or poses for selective subjects, a completion task needs to be undertaken. Existing methods use either factorization or completion schemes to recover the missing components. However, as the number of missing entries increases, factorization schemes may overfit the model because of incorrectly predefined ranks, while completion schemes may fail to obtain easy to interpret model factors (Chen et al., 2013). To this end, we propose a model that combines a rank minimization technique (Chen et al., 2013) with the DCOT model decomposition. Moreover, as the model

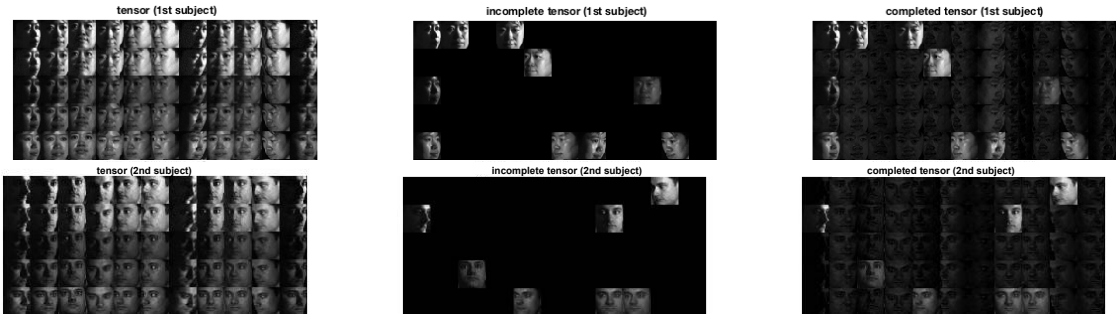

Figure 1: Illustration of the tensor completion with DCOT (third iteration step) on CMU face dataset of size $30 \times 11 \times 21 \times 1024$. The last column shows that DCOT encourages joint structures within subgroups of each subject and discriminative structures across members of different subjects.

structure is implicitly included in the DCOT model, we use the similarity function s^h to borrow neighborhood information from image data over an image-subject specific network.

The proposed method leverages the two schemes previously discussed and accurately estimates the model factors and missing entries via the following objective function

$$F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) + \lambda_1 \|\boldsymbol{\mathcal{G}}\|_F^2 + \lambda_2 \|\boldsymbol{\mathcal{H}}\|_F^2 + \sum_{n=1}^{N} \lambda_{3,n} \|\mathbf{U}^{(n)}\|_*, \tag{23}$$

where $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times I_n}$, $n = 1, \ldots, N$ are factor matrices; $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ are core tensors; and $\|\cdot\|_*$ denotes the trace norm. As an example, formulation (23) leverages similarity between members of I_3 in the CMU dataset, i.e, $\boldsymbol{\mathcal{H}}_{m_\kappa} = (1, :, \kappa, :)$ for $\kappa = 1, 2, \cdots, \pi$, and accross subjects I_1. The results are briefly depicted in Figure 1.

2.3.4 Integrative Tensor Factorization for Omics Multi-Modal Data

A major challenge for integrative analysis of multi-modal Omics data is the heterogeneity present across samples, as well as across different Omics data sources, which makes it difficult to identify the coordinated signal of interest from source-specific noise or extraneous effects. Tensor factorization methods are broadly used across multiple domains to analyze genomic datasets (Hore et al., 2016; Kim et al., 2017; Lee et al., 2018; Taguchi, 2017; Wang et al., 2015). In contrast to these methods, DCOT provides an approach for jointly decomposing the data matrices as slices of the data tensor. Formally, for non-negative observationally-linked datasets $\mathbf{X}_1, \ldots, \mathbf{X}_{I_3}$, we form a 3-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. Then, based on a non-negative DCOT factorization, the objective function becomes

$$F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) \quad + \quad \lambda_1 \mathbb{1}_{\boldsymbol{\mathcal{G}} \geq 0}(\boldsymbol{\mathcal{G}}) + \lambda_2 \mathbb{1}_{\boldsymbol{\mathcal{H}} \geq 0}(\boldsymbol{\mathcal{H}}) + \sum_{n=1}^{N} \lambda_{3,n} \mathbb{1}_{\mathbf{U}^{(n)} \geq 0}(\mathbf{U}^{(n)}). \tag{24}$$

Here, $\mathbb{1}_A(.)$ is the indicator function of set A; $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ is the n–th nonegative factor matrix for $n = 1, 2$; $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ is a core tensor reflecting the connections (or links) between the latent components and is able to capture the homogeneous part across sources; and $\boldsymbol{\mathcal{H}}$ is defined as $\boldsymbol{\mathcal{H}}(:,:,i) = \boldsymbol{\mathcal{H}}(:,:,j)$ for all $i, j \in [R_3]$ in order to detect coordinated activity (heterogeneous part) across multiple genomic variables in the form of multi-dimensional modules.

3. A Linearized ADMM Method for Penalized DCOT Decomposition

We develop a *linearized* ADMM to solve the regularized DCOT decomposition problem posited in (4). Let $\boldsymbol{\mathcal{T}} = (\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)}, \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{H}}, \boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{W}})$. To obtain the updates in the standard ADMM, we first formulate (4) as follows:

$$\begin{aligned} \underset{\boldsymbol{\mathcal{T}}}{\text{minimize}} \quad & F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) + \lambda_1 J_1(\boldsymbol{\mathcal{G}}) + \lambda_2 J_2(\boldsymbol{\mathcal{H}}) + \sum_{n=1}^{N} \lambda_{3,n} J_{3,n}(\mathbf{U}^{(n)}), \\ \text{s.t.} \quad & \big(\boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{H}}\big) \times_1 \mathbf{U}^{(1)} \cdots \times_N \mathbf{U}^{(N)} - \boldsymbol{\mathcal{Z}} = 0. \end{aligned} \tag{25}$$

By introducing the dual variable $\boldsymbol{\mathcal{W}}$ and parameter $\gamma > 0$, the standard ADMM is constructed for an augmented Lagrangian function defined by

$$\begin{aligned} \mathcal{L}(\boldsymbol{\mathcal{T}}) &= F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) + \lambda_1 J_1(\boldsymbol{\mathcal{G}}) + \lambda_2 J_2(\boldsymbol{\mathcal{H}}) + \sum_{n=1}^{N} \lambda_{3,n} J_{3,n}(\mathbf{U}^{(n)}) \\ &- \langle \boldsymbol{\mathcal{W}}, (\boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{H}}) \times_1 \mathbf{U}^{(1)} \cdots \times_N \mathbf{U}^{(N)} - \boldsymbol{\mathcal{Z}} \rangle \\ &+ \frac{\gamma}{2} \| (\boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{H}}) \times_1 \mathbf{U}^{(1)} \cdots \times_N \mathbf{U}^{(N)} - \boldsymbol{\mathcal{Z}} \|_F^2. \end{aligned} \tag{26}$$

In a typical iteration of the ADMM for solving (25), the following updates are implemented:

$$\begin{aligned} \mathbf{U}_{k+1}^{(1)} &= \underset{\mathbf{U}^{(1)}}{\text{argmin}} \quad \mathcal{L}\big(\underbrace{\mathbf{U}^{(1)}, \mathbf{U}_k^{(2)}, \ldots, \mathbf{U}_k^{(N)}, \boldsymbol{\mathcal{G}}_k, \boldsymbol{\mathcal{H}}_k, \boldsymbol{\mathcal{Z}}_k, \boldsymbol{\mathcal{W}}_k}_{=: \boldsymbol{\mathcal{T}}_k^{\mathbf{U}^{(1)}}} \big), \\ \mathbf{U}_{k+1}^{(n)} &= \underset{\mathbf{U}^{(n)}}{\text{argmin}} \quad \mathcal{L}\big(\underbrace{\mathbf{U}_{k+1}^{(1)}, \ldots, \mathbf{U}_{k+1}^{(n-1)}, \mathbf{U}^{(n)}, \mathbf{U}_k^{(n+1)}, \ldots, \mathbf{U}_k^{(N)}, \boldsymbol{\mathcal{G}}_k, \boldsymbol{\mathcal{H}}_k, \boldsymbol{\mathcal{Z}}_k, \boldsymbol{\mathcal{W}}_k}_{=: \boldsymbol{\mathcal{T}}_k^{\mathbf{U}^{(n)}}} \big), \\ \mathbf{U}_{k+1}^{(N)} &= \underset{\mathbf{U}^{(N)}}{\text{argmin}} \quad \mathcal{L}\big(\underbrace{\mathbf{U}_{k+1}^{(1)}, \mathbf{U}_{k+1}^{(2)}, \ldots, \mathbf{U}_{k+1}^{(N-1)}, \mathbf{U}^{(N)}, \boldsymbol{\mathcal{G}}_k, \boldsymbol{\mathcal{H}}_k, \boldsymbol{\mathcal{Z}}_k, \boldsymbol{\mathcal{W}}_k}_{=: \boldsymbol{\mathcal{T}}_k^{\mathbf{U}^{(N)}}} \big), \\ \boldsymbol{\mathcal{G}}_{k+1} &= \underset{\boldsymbol{\mathcal{G}}}{\text{argmin}} \quad \mathcal{L}\big(\underbrace{\mathbf{U}_{k+1}^{(1)}, \ldots \mathbf{U}_{k+1}^{(N)}, \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{H}}_k, \boldsymbol{\mathcal{Z}}_k, \boldsymbol{\mathcal{W}}_k}_{=: \boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{G}}}} \big), \\ \boldsymbol{\mathcal{H}}_{k+1} &= \underset{\boldsymbol{\mathcal{H}}}{\text{argmin}} \quad \mathcal{L}\big(\underbrace{\mathbf{U}_{k+1}^{(1)}, \ldots \mathbf{U}_{k+1}^{(N)}, \boldsymbol{\mathcal{G}}_{k+1}, \boldsymbol{\mathcal{H}}, \boldsymbol{\mathcal{Z}}_k, \boldsymbol{\mathcal{W}}_k}_{=: \boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{H}}}} \big), \\ \boldsymbol{\mathcal{Z}}_{k+1} &= \underset{\boldsymbol{\mathcal{Z}}}{\text{argmin}} \quad \mathcal{L}\big(\underbrace{\mathbf{U}_{k+1}^{(1)}, \ldots \mathbf{U}_{k+1}^{(N)}, \boldsymbol{\mathcal{G}}_{k+1}, \boldsymbol{\mathcal{H}}_{k+1}, \boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{W}}_k}_{=: \boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{Z}}}} \big), \\ \boldsymbol{\mathcal{W}}_{k+1} &= \boldsymbol{\mathcal{W}}_k - \gamma\big((\boldsymbol{\mathcal{G}}_{k+1} + \boldsymbol{\mathcal{H}}_{k+1}) \times_1 \mathbf{U}_{k+1}^{(1)} \cdots \times_N \mathbf{U}_{k+1}^{(N)} - \boldsymbol{\mathcal{Z}}_{k+1} \big), \end{aligned} \tag{27}$$

where $n \in \{2, 3, \cdots, N-1\}$.

Note that problem (25) is non-convex; hence, the global convergence of ADMM is a priori not guaranteed. Recent work (Hong et al., 2016; Wang et al., 2019; Lin et al., 2016; Tarzanagh and Michailidis, 2018a) studied the convergence of ADMM for non-convex and non-smooth problems under linear constraints. However, the constraints in the tensor factorization problem are nonlinear. To avoid introducing auxiliary variables and still solving (25) efficiently, we propose to approximate each sub-problem in (27) by linearizing the smooth terms with respect to the factor matrices and core tensors. With this linearization, the resulting approximation to (27) is then simple enough to have a closed form solution, and we are able to provide the global convergence under mild conditions.

To do so, we regularize each subproblem in (27) and consider the following updates:

$$\mathbf{U}_{k+1}^{(n)} = \underset{\mathbf{U}^{(n)}}{\operatorname{argmin}} \quad \mathcal{L}(\boldsymbol{\mathcal{T}}_k^{\mathbf{U}^{(n)}}) + \frac{\varrho^n}{2}\|\mathbf{U}^{(n)} - \mathbf{U}_k^{(n)}\|_F^2, \qquad n = 1, \cdots N, \tag{28a}$$

$$\boldsymbol{\mathcal{G}}_{k+1} = \underset{\boldsymbol{\mathcal{G}}}{\operatorname{argmin}} \quad \mathcal{L}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{G}}}) + \frac{\varrho^g}{2}\|\boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{G}}_k\|_F^2, \tag{28b}$$

$$\boldsymbol{\mathcal{H}}_{k+1} = \underset{\boldsymbol{\mathcal{H}}}{\operatorname{argmin}} \quad \mathcal{L}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{H}}}) + \frac{\varrho^h}{2}\|\boldsymbol{\mathcal{H}} - \boldsymbol{\mathcal{H}}_k\|_F^2, \tag{28c}$$

$$\boldsymbol{\mathcal{Z}}_{k+1} = \underset{\boldsymbol{\mathcal{Z}}}{\operatorname{argmin}} \quad \mathcal{L}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{Z}}}), \tag{28d}$$

$$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k - \gamma\big((\boldsymbol{\mathcal{G}}_{k+1} + \boldsymbol{\mathcal{H}}_{k+1}) \times_1 \mathbf{U}_{k+1}^{(1)} \cdots \times_N \mathbf{U}_{k+1}^{(N)} - \boldsymbol{\mathcal{Z}}_{k+1}\big), \tag{28e}$$

where positive constants ϱ^g, ϱ^h, and $\{\varrho^n\}_{n=1}^N$ correspond to the regularization parameters.

It follows from (26) that

$$\mathcal{L}(\boldsymbol{\mathcal{T}}) = F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) + \lambda_1 J_1(\boldsymbol{\mathcal{G}}) + \lambda_2 J_2(\boldsymbol{\mathcal{H}}) + \sum_{n=1}^{N} \lambda_{3,n} J_{3,n}(\mathbf{U}^{(n)}) + \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}), \tag{29}$$

where

$$\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}) := \frac{\gamma}{2}\|(\boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{H}}) \times_1 \mathbf{U}^{(1)} \cdots \times_N \mathbf{U}^{(N)} - \boldsymbol{\mathcal{Z}} - \frac{1}{\gamma}\boldsymbol{\mathcal{W}}\|_F^2. \tag{30}$$

Now, using (29), we approximate (28a)-(28c) by linearizing the function $\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}})$ with respect to $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)}, \boldsymbol{\mathcal{G}}$, and $\boldsymbol{\mathcal{H}}$ as follows:

$$\mathbf{U}_{k+1}^{(n)} = \underset{\mathbf{U}^{(n)}}{\operatorname{argmin}} \langle \nabla_{\mathbf{U}^{(n)}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\mathbf{U}_k^{(n)}}), \mathbf{U}^{(n)} - \mathbf{U}_k^{(n)} \rangle + \lambda_{3,n} J_3(\mathbf{U}^{(n)}) + \frac{\varrho^n}{2}\|\mathbf{U}^{(n)} - \mathbf{U}_k^{(n)}\|_F^2, \tag{31a}$$

$$\boldsymbol{\mathcal{G}}_{k+1} = \underset{\boldsymbol{\mathcal{G}}}{\operatorname{argmin}} \langle \nabla_{\boldsymbol{\mathcal{G}}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{G}}_k}), \boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{G}}_k \rangle + \lambda_1 J_1(\boldsymbol{\mathcal{G}}) + \frac{\varrho^g}{2}\|\boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{G}}_k\|_F^2, \tag{31b}$$

$$\boldsymbol{\mathcal{H}}_{k+1} = \underset{\boldsymbol{\mathcal{H}}}{\operatorname{argmin}} \langle \nabla_{\boldsymbol{\mathcal{H}}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{H}}_k}), \boldsymbol{\mathcal{H}} - \boldsymbol{\mathcal{H}}_k \rangle + \lambda_2 J_2(\boldsymbol{\mathcal{H}}) + \frac{\varrho^h}{2}\|\boldsymbol{\mathcal{H}} - \boldsymbol{\mathcal{H}}_k\|_F^2. \tag{31c}$$

Here, $\nabla_{\mathbf{U}^{(n)}} \bar{\mathcal{L}}$, $\nabla_{\boldsymbol{\mathcal{G}}} \bar{\mathcal{L}}$, and $\nabla_{\boldsymbol{\mathcal{H}}} \bar{\mathcal{L}}$ denote the gradients of (30) w.r.t. $\mathbf{U}^{(n)}$, $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{H}}$, respectively.

The following lemma gives the partial gradients of $\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}})$ w.r.t. $\mathbf{U}^{(n)}$, $\boldsymbol{\mathcal{G}}$, and $\boldsymbol{\mathcal{H}}$.

Algorithm 1 Regularized and Smooth DCOT Factorization via Linearized ADMM

Input: $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, positive constants $\lambda_1, \lambda_2, \lambda_{3,i}, i = 1, \ldots, N$, factor matrices $\mathbf{U}_0^{(n)} \in \mathbb{R}^{I_n \times R_n}, n = 1, \ldots, N$, dual variable $\boldsymbol{\mathcal{W}}_0 \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, smoothing function s^h, and a dual step size $\gamma > 2L_F$ where L_F is a Lipschitz constant for the gradient of F.
Initialize: $k = 0$, $\mathbf{U}_k^{(n)} = \mathbf{U}_0^{(n)}$ for $n = 1, \ldots, N$, $\boldsymbol{\mathcal{Z}}_k = \boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{G}}_k = \boldsymbol{\mathcal{H}}_k = \boldsymbol{\mathcal{X}} \times_1 \mathbf{U}_k^{(1)} \cdots \times_N \mathbf{U}_k^{(N)}$, and $\boldsymbol{\mathcal{W}}_k = \boldsymbol{\mathcal{W}}_0$.
For $k = 1, 2, \ldots$

- For $n = 1, \ldots, N$, update the factor matrix $\mathbf{U}^{(n)}$:

$$\mathbf{U}_{k+1}^{(n)} \quad = \quad \text{prox}_{\varrho^n}^{J_{3,n}} \left(\mathbf{U}_k^{(n)} - \frac{1}{\varrho^n} \nabla_{\mathbf{U}^{(n)}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}^{\mathbf{U}_k^{(n)}}), \frac{\lambda_{3,n}}{\varrho^n} \right).$$

- Update the homogeneous core $\boldsymbol{\mathcal{G}}$:

$$\boldsymbol{\mathcal{G}}_{k+1} \quad = \quad \text{prox}_{\varrho^g}^{J_1} \left(\boldsymbol{\mathcal{G}}_k - \frac{1}{\varrho^g} \nabla_{\boldsymbol{\mathcal{G}}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{G}}_k}), \frac{\lambda_1}{\varrho^g} \right).$$

- For $m = 1, \ldots, M$, update the heterogeneous core $\boldsymbol{\mathcal{H}}_{m_\pi}$:

$$\boldsymbol{\mathcal{H}}_{m_\pi,k+1} \quad = \quad \text{prox}_{\varrho^h}^{J_2} \left(\boldsymbol{\mathcal{H}}_{m_\pi,k} - \frac{1}{\varrho^h} \nabla_{\boldsymbol{\mathcal{H}}_{m_\pi}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{H}}_{m_\pi,k}}), \frac{\lambda_2}{\varrho^h} \right),$$

  and set $\boldsymbol{\mathcal{H}}_{m_1,k+1} = \boldsymbol{\mathcal{H}}_{m_2,k+1} = \cdots = \boldsymbol{\mathcal{H}}_{m_\pi,k+1}$.
- Update the model parameter $\boldsymbol{\mathcal{Z}}$:

$$\boldsymbol{\mathcal{Z}}_{k+1} \quad = \quad \underset{\boldsymbol{\mathcal{Z}}}{\text{argmin}} \left\{ F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) + \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{Z}}}) \right\}.$$

- Update the dual variable $\boldsymbol{\mathcal{W}}$:

$$\boldsymbol{\mathcal{W}}_{k+1} \quad = \quad \boldsymbol{\mathcal{W}}_k - \gamma \left((\boldsymbol{\mathcal{G}}_{k+1} + \boldsymbol{\mathcal{H}}_{k+1}) \times_1 \mathbf{U}_{k+1}^{(1)} \cdots \times_N \mathbf{U}_{k+1}^{(N)} - \boldsymbol{\mathcal{Z}}_{k+1} \right).$$

End

Lemma 5 *The partial gradients of* $\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}})$ *are*

$$\nabla_{\mathbf{U}^{(n)}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}) = \mathbf{M}_{(t)} (\bigotimes_{t \neq n} \mathbf{U}^{(t)}) (\mathbf{H}_{(t)} + \mathbf{G}_{(t)})^\top, \qquad n = 1, \ldots, N,$$
$$\nabla_{\boldsymbol{\mathcal{H}}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}) = \nabla_{\boldsymbol{\mathcal{G}}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}) = \boldsymbol{\mathcal{M}} \times_1 \mathbf{U}^{(1)} \cdots \times_N \mathbf{U}^{(N)},$$

where $\mathbf{M}_{(t)}$ *denotes mode-"t" matricization of* $\boldsymbol{\mathcal{M}}$*, and*

$$\boldsymbol{\mathcal{M}} = \gamma \Big((\boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{H}}) \times_1 \mathbf{U}^{(1)} \cdots \times_N \mathbf{U}^{(N)} - \boldsymbol{\mathcal{Z}} - \frac{1}{\gamma} \boldsymbol{\mathcal{W}} \Big). \tag{32}$$

A schematic description of the proposed ADMM is given in Algorithm 1.

3.1 Global Convergence

Before establishing the global convergence result of our algorithm for DCOT, we provide the necessary definitions used in the proofs. Most of the concepts that we use in this paper can be found in Rockafellar and Wets (2009); Bauschke et al. (2011).

For any proper, lower semi-continuous function $g : H \to (-\infty, \infty]$, we let $\partial_L g : H \to 2^H$ denote the *limiting subdifferential* of g; see (Rockafellar and Wets, 2009, Definition 8.3).

For any $\eta \in (0, \infty)$, we let F_η denote the class of concave continuous functions $\varphi : [0, \eta) \to \mathbb{R}_+$ for which $\varphi(0) = 0$; φ is C^1 on $(0, \eta)$ and continuous at 0; and for all $s \in (0, \eta)$, we have $\varphi'(s) > 0$.

Definition 6 (Kurdyka–Łojasiewicz Property) *A function $g : H \to (-\infty, \infty]$ has the Kurdyka-Łojasiewicz (KL) property at $\overline{u} \in \operatorname{dom}(\partial_L g)$ provided that there exists $\eta \in (0, \infty)$, a neighborhood U of $\overline{u}$, and a function $\varphi \in F_\eta$ such that*

$$\left(\forall u \in U \cap \{u' \mid g(\overline{u}) < g(u') < g(\overline{u}) + \eta\}\right), \qquad \varphi'(g(u) - g(\overline{u}))\, dist(0, \partial_L g(u)) \geq 1.$$

The function g is said to be a KL function provided it has the KL property at each point $u \in \operatorname{dom}(g)$.

In the following Theorem 7, we establish the global convergence of the standard multi-block ADMM for solving the DCOT decomposition problem, by using the KL property of the objective function in (26).

Theorem 7 (Global Convergence) *Suppose Assumption A holds and the augmented Lagrangian $\mathcal{L}(\boldsymbol{\mathcal{T}})$ is a KL function. Then, the sequence $\boldsymbol{\mathcal{T}}_k = (\mathbf{U}_k^{(1)}, \ldots, \mathbf{U}_k^{(N)}, \boldsymbol{\mathcal{G}}_k, \boldsymbol{\mathcal{H}}_k, \boldsymbol{\mathcal{Z}}_k, \boldsymbol{\mathcal{W}}_k)$ generated by Algorithm 1 from any starting point converges to a stationary point of Problem (26).*

Semi-Algebraic functions are an important class of objectives for which Algorithm 1 converges:

Definition 8 (Semi-Algebraic Functions) *A function $\Psi : H \to (0, \infty]$ is* semi-algebraic *provided that the graph $\mathbb{G}(\Psi) = \{(x, \Psi(x)) \mid x \in H\}$ is a semi-algebraic set, which in turn means that there exists a finite number of real polynomials $g_{ij}, h_{ij} : H \times \mathbb{R} \to \mathbb{R}$ such that*

$$\mathbb{G}(\Psi) := \bigcup_{j=1}^{p} \bigcap_{i=1}^{q} \{u \in H \mid g_{ij}(u) = 0 \text{ and } h_{ij}(u) < 0\}.$$

Definition 9 (Sub-Analytic Functions) *A function $\Psi : H \to (0, \infty]$ is* sub-analytic *provided that the graph $\mathbb{G}(\Psi) = \{(x, \Psi(x)) \mid x \in H\}$ is a sub-analytic set, which in turn means that there exists a finite number of real analytic functions $g_{ij}, h_{ij} : H \times \mathbb{R} \to \mathbb{R}$ such that*

$$\mathbb{G}(\Psi) := \bigcup_{j=1}^{p} \bigcap_{i=1}^{q} \{u \in H \mid g_{ij}(u) = 0 \text{ and } h_{ij}(u) < 0\}.$$

It can be easily seen that both real analytic and semi-algebraic functions are sub-analytic. In general, the sum of two sub-analytic functions is not necessarily sub-analytic. However, it is easy to show that for two sub-analytic functions, if at least one function maps bounded sets to bounded sets, then their sum is also sub-analytic (Bolte et al., 2014).

The KL property has been shown to hold for a large class of functions including sub-analytic and semi-algebraic functions such as indicator functions of semi-algebraic sets, vector (semi)-norms $\|\cdot\|_p$ with $p \geq 0$ be any rational number, and matrix (semi)-norms (e.g., operator, trace, and Frobenious norm). These function classes cover most of smooth and nonconvex objective functions encountered in practical applications; see Bolte et al. (2014) for a comprehensive list.

Remark 10 *Each penalty function J_i in* (26) *is a semi-algebraic function, while the loss function F is sub-analytic. Hence, the augmented Lagrangian function*

$$\begin{aligned}\mathcal{L}(\boldsymbol{\mathcal{T}}) &= F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) + \lambda_1 J_1(\boldsymbol{\mathcal{G}}) + \lambda_2 J_2(\boldsymbol{\mathcal{H}}) + \sum_{n=1}^{N} \lambda_{3,n} J_{3,n}(\mathbf{U}^{(n)}) \\ &- \langle \boldsymbol{\mathcal{W}}, (\boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{H}}) \times_1 \mathbf{U}^{(1)} \cdots \times_N \mathbf{U}^{(N)} - \boldsymbol{\mathcal{Z}} \rangle \\ &+ \frac{\gamma}{2} \|(\boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{H}}) \times_1 \mathbf{U}^{(1)} \cdots \times_N \mathbf{U}^{(N)} - \boldsymbol{\mathcal{Z}}\|_F^2,\end{aligned}$$

which is the summation of semi-algebraic functions is itself semi-algebraic. augmented Lagrangian function $\mathcal{L}(\boldsymbol{\mathcal{T}})$ satisfies the KL property.

4. Consistency of the DCOT Factorization

In this section, we derive asymptotic properties for the proposed DCOT factorization using the ℓ_2-smoothing loss function defined in (12). In particular, we focus on the Gaussian case where $x_{i_1,\cdots,i_N}$ satisfies (11). Under this setting, we provide the estimation error rate as a function of the sample size Ω, the maximum rank $R_{\max} = \max\{R_1, R_2, \cdots, R_N\}$, and the tuning parameter λ and show the necessity of the smoothing function s^h for providing a faster convergence rate and a small prediction error.

Let $\widehat{\boldsymbol{\mathcal{Z}}} \in \Gamma(\boldsymbol{\mathcal{Z}})$ denote an estimator of $\boldsymbol{\mathcal{Z}}^*$. The prediction accuracy of $\hat{\boldsymbol{\mathcal{Z}}}$ is defined by the root mean square error (RMSE):

$$\rho(\widehat{\boldsymbol{\mathcal{Z}}}, \boldsymbol{\mathcal{Z}}^*) = \left(\frac{1}{\Omega} \sum_{(i_1,\cdots,i_N)\in\Omega} (\hat{z}_{i_1\cdots i_N} - z^*_{i_1\cdots i_N})^2 \right)^{\frac{1}{2}}. \tag{33}$$

In order to provide the asymptotic behavior of the penalized DCOT, we require the following technical assumptions:

Assumption B *Let $\{c_{i_n,j_n}\}_{n=1}^N$ be the label constraints defined in* (6)*. Then, there exist constants $a_1 \geq 0$ and $\alpha > 0$, such that for any N-tuples $(i_1, i_2, \cdots, i_N)$ and $(j_1, j_2, \cdots, j_N)$*

$$\left|z^*_{i_1\cdots i_N} - z^*_{j_1\cdots j_N}\right| \leq a_1 R_{\max} \max\Big\{ \sum_{n=1}^{N} \big(d(\mathbf{y}_{i_n}, \mathbf{y}_{j_n})\big)^{\alpha}, \mathbb{1}_{\prod_{n=1}^N c_{i_n,j_n}=0} \Big\},$$

where $d(\mathbf{y}_{i_n}, \mathbf{y}_{j_n})$ denotes the distance between $\mathbf{y}_{i_n}$ and $\mathbf{y}_{j_n}$, and $R_{\max} = \max\{R_1, R_2, \cdots, R_N\}$.

Assumption B describes the smoothness of $z^*_{i_1\cdots i_N}$ in terms of the side information $\boldsymbol{\mathcal{Y}}$. We that if $d(\mathbf{y}_{i_n}, \mathbf{y}_{j_n}) = 0$, and $c_{i_n,j_n} = 1$ for all $n \in [N]$, Assumption B degenerates to $z^*_{i_1\cdots i_N} = z^*_{j_1 j_2 \cdots j_N}$. This assumption is mild when for example all fibers $\mathbf{y}_{i_n}$ of $\boldsymbol{\mathcal{Y}}$ are available, and is relatively more restrictive when they are absent. In the case when $N = 2$, Assumption B reduces to a variant of the regularity condition used in Vieu (1991); Wasserman (2006); Stone et al. (1984); Marron et al. (1987); Dai et al. (2019).

Assumption C *The tensor $\boldsymbol{\mathcal{Y}}$ has bounded support $\Psi \subset \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and the error term $\epsilon_{i_1\cdots i_N}$ defined in* (11) *has a sub-Gaussian distribution with variance σ^2.*

This assumption is the *regularity condition* for the underlying probability distribution, and similar assumptions are widely used in literature to provide the asymptotic behavior of the matrix factorization methods (Bi et al., 2017; Dai et al., 2019).

The next result provides a general upper bound of the root mean square error $\rho(\widehat{\boldsymbol{\mathcal{Z}}}, \boldsymbol{\mathcal{Z}}^*)$, which may vary by the window size h, the maximum rank $R_{\max} = \max\{R_1, R_2, \cdots, R_N\}$, and the number of observed variables Ω.

Theorem 11 *Suppose Assumptions B and C hold. Let*

$$\phi_1 := \max_{(i_1,\cdots i_N)} \sum_{(j_1,\cdots,j_N)\in\Omega} s^h_{i_1\cdots i_N, j_1\cdots j_N} \sum_{n=1}^{N} \left(d(\mathbf{y}_{i_n}, \mathbf{y}_{j_n})\right)^{\alpha}, \quad \text{and}$$
$$\phi_2 := \max_{(i_1,\cdots i_N)} \sum_{(j_1,\cdots,j_N)\in\Omega} \left(s^h_{i_1\cdots i_N, j_1\cdots j_N}\right)^2.$$

Then, for some positive constant a_2, we have

$$P\left(\rho(\widehat{\boldsymbol{\mathcal{Z}}}, \boldsymbol{\mathcal{Z}}^*) \geq \eta\right) \leq \exp\left(-\frac{a_2\eta^2}{\phi_2} + \sum_{n\in[N]} \log I_n\right),$$

provided that

$$\eta \geq \max\left\{\sqrt{R_{\max}}\phi_1, \sqrt{\phi_2}\right\} \sum_{n\in[N]} \log I_n, \quad \text{and}$$
$$\lambda_1 J_1(\boldsymbol{\mathcal{G}}^*) + \lambda_2 J_2(\boldsymbol{\mathcal{H}}^*) + \sum_{n=1}^{N} \lambda_{3,n} J_{3,n}(\mathbf{U}^{(n)*}) \leq \eta^2.$$

Theorem 11 is quite general in terms of the rates of $I_1, \cdots, I_N$. If ϕ_1 and ϕ_2 tend to zero and can be computed for some specific smoothing parameters, the convergence rate then becomes

$$\rho(\widehat{\boldsymbol{\mathcal{Z}}}, \boldsymbol{\mathcal{Z}}^*) \leq \max\left\{\sqrt{R_{\max}}\phi_1, \sqrt{\phi_2}\right\} \sum_{n\in[N]} \log I_n.$$

The result of Theorem 11, i.e., the upper bound of $\rho(\widehat{\boldsymbol{\mathcal{Z}}}, \boldsymbol{\mathcal{Z}}^*)$ may vary by the choice of parameters ϕ_1 and ϕ_2. Next, we provide an explicit convergence rate under some additional assumptions.

Assumption D *Let* $s^h_{i_n,j_n} = K(h^{-1}\|\mathbf{y}_{i_n} - \mathbf{y}_{j_n}\|_2)$ *and assume that the kernel function* $K(.)$ *satisfies*

$$\max\left\{\int_0^\infty K^2(u)du, \int_0^\infty K(u)u^\alpha du\right\} \leq a_3 \tag{34}$$

for α *defined in Assumption B and some finite* $a_3 > 0$.

Assumption D is widely used in literature for smoothing kernels (Bi et al., 2017; Dai et al., 2019). Kernels with an exponential decay rate, such as the RBF and Gaussian kernels always satisfy Assumption D.

For any $(i_1, \cdots, i_N)$ and $(j_1, \cdots, j_N)$, let

$$U_{i_1\cdots i_N, j_1\cdots j_N} := \sum_{n=1}^{N} \|\mathbf{y}_{i_n} - \mathbf{y}_{j_n}\|_2, \tag{35}$$

and

$$\Delta_{i_1\cdots i_n} := \begin{cases} 1 & \text{if } x_{i_1\cdots i_N} \in \Omega \\ 0 & \text{otherwise} \end{cases}$$

Assume that $(y_{i_1\cdots i_N}, \Delta_{i_1\cdots i_N})$ are independent and identically distributed, but the distribution of $\Delta_{i_1\cdots i_N}$ may depend on $y_{i_1\cdots i_N}$.

Assumption E *For any* $(i_1, \cdots, i_N)$ *and* $(j_1, \cdots, j_N)$, $P(c_{i_1\cdots i_N, j_1\cdots j_N} = 1 | \Delta_{j_1\cdots j_N} = 1)$ *is bounded away from zero, and the conditional density*

$$f_{U_{i_1\cdots i_N, j_1\cdots j_N} | c_{i_1\cdots i_N, j_1\cdots j_N}=1, \Delta=1}$$

is continuous and bounded away from zero. Here, $c_{i_1\cdots i_N, j_1\cdots j_N} = \prod_{n=1}^N c_{i_n,j_n}$.

Assumption E ensures that for any pair $(i_1, \cdots, i_N)$, the probability of $\Delta_{i_1\cdots i_N}$ may depend on $y_{i_1\cdots i_N}$ and $s^h_{i_1\cdots i_N, j_1\cdots j_N}$ and that the corresponding neighboring pairs are observed with positive probability.

The following corollary provides an explicit value of ϕ_1 and ϕ_2, and the convergence rate for DCOT factorization using the smoothing loss function defined in (10).

Corollary 12 (Convergence Rate) *Suppose Assumptions B–E hold. Then, we have* $\phi_1 = h^\alpha$, $\phi_2 = (|\Omega|h)^{-1}$, *and*

$$\rho(\widehat{\boldsymbol{\mathcal{Z}}}, \boldsymbol{\mathcal{Z}}^*) = O\left(\frac{\log\left(\prod_{n\in[N]} I_n\right)}{|\Omega|^{\frac{\alpha}{2\alpha+1}}}\right) \tag{36}$$

provided that $R_{\max} = O(1)$.

Remark 13 *Next we discuss the connections of our bound* (36) *with prior results in the literature. For* $\alpha > 1/2$, *since* $|\Omega| \leq \prod_{n\in[N]} I_n < (\sum_{n\in[N]} I_n)^2$, *smoothing DCOT can achieve significantly better rate than* $O\big((\frac{\sum_{n\in[N]} I_n}{|\Omega|}\log(\frac{\sqrt{\prod_{n\in[N]} I_n}}{|\Omega|}))^{\frac{1}{2}}\big)$ *established in Bi et al. (2017, 2018) for matrix and tensors, respectively. In addition, Corollary 12 reveals an*

interesting theoretical property of the smooth matrix factorization proposed by Dai et al. (2019) and suggests the convergence rate of the smooth tensor factorization for tensor (structured) data can be significantly better than that for matrix (unstructured) data (Dai et al., 2019). Indeed, when $N = 2$, it shows that the estimation error is bounded by $O\big(\frac{(I_1+I_2)^{1/(2\alpha+1)}}{|\Omega|^{1/2}}\log(I_1I_2)\big)$ which is similar to the one provided in Dai et al. (2019, Corollary 1). This indicates a disadvantage of matricizing (unfolding) a data tensor for completion tasks such as recommender systems. More specifically, for unstructured data the bound scales linearly as $\alpha \to 0^+$ with the product of the factor matrices dimensions, whereas for tensor-structured data the bound scales linearly with the sum of the factor matrices dimensions.

5. Experimental Results

We test the performance of DCOT and its smoothing version (called S-DCOT) on a number of data analytics tasks, including subspace clustering, imaging tensor completion and denoising, recommender systems, dictionary learning, and multi-platform cancer analysis in terms of accuracy and scalability.

Algorithm 1 requires a good initializer to achieve good performance, which is also the case for the Tucker decomposition. To that end, we use DCOT with HOSVD (De Lathauwer et al., 2000b) and random initialization, called DCOT(H) and DCOT(R), respectively. In the first setting, given a tensor $\boldsymbol{\mathcal{X}}_0$, we construct the mode-"n" matricization $\boldsymbol{\mathcal{X}}_{0,(n)}$. Then, we compute the singular value decomposition $\boldsymbol{\mathcal{X}}_{0,(n)} = \mathbf{U}_r^{(n)}\mathbf{D}_r^{(n)}\mathbf{V}_r^{(n)}$, and store the left singular vectors $\mathbf{U}^{(n)}$. In both cases, the core tensor $\boldsymbol{\mathcal{G}}$ is the projection of $\boldsymbol{\mathcal{X}}$ onto the tensor basis formed by the factor matrices $\{\mathbf{U}^{(n)}\}_{n=1}^N$, i.e., $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{X}} \times_{n=1}^N \mathbf{U}^{(n)^\top}$. The initial heterogeneous core $\boldsymbol{\mathcal{H}}$ is set equal to $\boldsymbol{\mathcal{G}}$.

To select tuning parameters $\{\lambda_1, \lambda_2, \lambda_{3,1}, \lambda_{3,2}, \cdots, \lambda_{3,N}\}$, $\{R_i\}_{i=1}^N$, we search over a set of grid points aiming to minimize the RMSE defined in (33) or the average detection accuracy of clustering on the validation set. Specifically, we used the following grids of values for the parameter search:

- Regualrization paramaters $\{\lambda_1, \lambda_2, \lambda_{3,1}, \lambda_{3,2}, \cdots, \lambda_{3,N}\}$ are selected in

$$\lambda_1, \lambda_2 \in \left\{ \frac{1}{\|\boldsymbol{\mathcal{X}}\|_F} \cdot 10^{0.1(\nu-21)};\ \nu = 1, \cdots, 41 \right\},$$
$$\lambda_{3,1}, \ldots, \lambda_{3,N} \in \left\{ 10^{0.1(\nu-21)};\ \nu = 1, \cdots, 41 \right\}, \tag{37a}$$

- Tensor ranks $\{R_i\}_{i=1}^N$ ranging from

$$\{5, 10, 15, 20, 25, 30, 50\}. \tag{37b}$$

We note that scaling $\{\lambda_1, \lambda_2, \lambda_{3,1}, \lambda_{3,2}, \cdots, \lambda_{3,N}\}$ with tensor norms is motivated by the STDC model (Chen et al., 2013, 3.5.2), and provides an adaptive way to balance the impacts of the factor matrices and tensor cores for tensor factorization with smooth and gradient Lipschitz losses. In the implementation of S-DCOT, we use the average of 10 multiple

Gaussian kernels with h selected in $\{0.75, 1, 1.25, \cdots, 3\}$ to define $s^h_{i_n,j_n}$. The choice of a Gaussian kernel is due to the better empirical performance obtained, compared to other possibilities. We set c_{i_n,j_n} to 0.8 if fibers belong to same clusters (subjects) and 0.2 otherwise. The Kronecker-product similarity function is defined as in (6). The smoothing functions are normalized such that $\sum_{j_1 \cdots j_N} s^h_{i_1 \cdots i_N, j_1 \cdots j_N} = 1$. We also set $\gamma = 1/\sigma_1(\mathbf{X}_{(1)}{\mathbf{X}_{(1)}}^\top)$ as suggested by (Chen et al., 2013).

Regarding the selection of the number of subjects M, we note that a rather small M may not be adequately "powered" to distinguish between the proposed method and the Tucker method. In practice, if subjects (clusters) are based on categorical variables, then we can use existing categories, and hence M is known. However, if clustering is based on a continuous variable, we can apply the quantiles of the continuous variable to determine M and "quantize" the dataset accordingly; see, (Wang, 2010).

5.1 DCOT for Tensor Completion Problems

Next, we examine the performance of DCOT [2] factorization for different tensor completion tasks.

5.1.1 Image Completion and Denoising Problems

We use S-DCOT for image completion and compare it with the following tensor factorization methods for image processing: fully Bayesian CP factorization using mixture prior (FBCP-MP) (Zhao et al., 2015), simultaneous tensor decomposition and completion (STDC) using factor prior (Chen et al., 2013), high accuracy low rank tensor completion (HaLRTC) (Liu et al., 2013), exact tensor completion using TSVD (Zhang and Aeron, 2016), and Low-rank Tensor Completion by Parallel Matrix Factorization (TMAC) (Xu et al., 2013) [3].

We applied our tensor completion method proposed in Subsection 2.3.3 to the 4D CMU faces database (Sim et al., 2002) and the Cine Cardiac dataset (Lingala et al., 2011). The CMU dataset (Sim et al., 2002) comprises of 65 subjects with 11 poses, and 21 types of illumination. All face images are aligned by their eye coordinates and then cropped and resized into 32×32 images. Images are vectorized, and the dataset is arranged as a fourth-order tensor. Thus, the size of the CMU data is $65 \times 11 \times 21 \times 1024$. Since each facial image is similar under different illuminations, but for similar faces, poses are not necessarily similar, we consider the following partitions

$$\boldsymbol{\mathcal{H}}_{m_\kappa} = \boldsymbol{\mathcal{H}}(m, :, \kappa, :), \quad \kappa = 1, \ldots, 21, \quad m = 1, \ldots, 65,$$

which enforces the similarity across members of I_3.

Dynamic cardiac imaging is performed either in cine or real-time mode. Cine MRI, the clinical gold-standard for measuring cardiac function/volumetrics (Bogaert et al., 2012), produces a movie of roughly 20 cardiac phases over a single cardiac cycle (heart beat). However, by exploiting the semi-periodic nature of cardiac motion, it is actually formed over many heart beats. Cine sampling is gated to a patient's heart beat, and as each data

2. https://github.com/Tarzanagh/DCOT
3. The codes can be obtained from `https://github.com/qbzhao/BCPF`, `https://sites.google.com/site/fallcolor/projects/stdc`, `http://www.cs.rochester.edu/u/jliu/`, and `https://xu-yangyang.github.io/software.html`, respectively.

		S-DCOT(H)		FBCP-MP		STDC		HaLRTC		TSVD		TMAC	
Data	ρ	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	time	RMSE	Time
CMU	0.8	15.04e-2	47.22	30.28e-2	125.19	21.77e-2	79.48	31.89e-2	225.15	40.11e-2	112.48	29.19e-2	40.69
	0.85	20.17e-2	42.31	38.49e-15	108.23	28.28e-2	57.19	35.14e-2	189.05	43.12e-2	99.82	24.15e-2	39.18
	0.9	26.13e-2	29.27	44.15e-2	97.37	43.21e-2	35.72	49.17e-2	155.28	56.71e-2	104.33	44.26e-2	27.04
	0.95	54.84e-2	19.79	76.41e-2	88.19	81.85e-2	13.11	67.77e-2	114.36	88.79e-2	111.49	85.14e-2	95.6
Cine	0.8	18.34e-2	41.39	42.31e-2	112.39	33.71e-2	88.39	41.11e-2	178.38	49.71e-2	110.13	33.12e-2	55.01
	0.85	23.54e-2	39.29	48.25e-2	110.18	44.15e-2	76.32	55.10e-2	144.82	55.19e-2	114.23	35.11e-2	47.33
	0.9	29.13e-2	34.45	30.24e-2	108.34	55.38e-2	88.72	55.19e-2	107.71	59.13e-2	101.41	46.13e-2	41.55
	0.95	51.33e-2	27.88	77.25e-2	76.85	78.31e-2	23.05	78.08e-2	99.12	76.44e-2	78.45	57.14e-2	39.01

Table 1: RMSE and runtime (seconds) of tensor completion methods on imaging datasets.

measurement is captured it is associated with a particular cardiac phase. This process continues until enough data has been collected such that all image frames are complete. Typically, an entire 2D cine cardiac MRI series is acquired within a single breath hold (less than 30 secs). In our experiment, we consider a bSSFP long-axis cine sequence ($n = 192$, $t = 19$) acquired at 1.5 T (Tesla) magnets, using an phased-array cardiac receiver coil ($l = 8$ channels) (Candes et al., 2013). Hence, the size of the Cine Cardiac data is $192 \times 192 \times 8 \times 19$. We use the following partitions

$$\boldsymbol{\mathcal{H}}_{m_\kappa} = \boldsymbol{\mathcal{H}}(:,:,m,\kappa), \quad \kappa = 1,\dots,19, \quad m = 1,\cdots,8,$$

which enforces the similarity across time dimension.

Table 1 shows the completion results on the imaging datasets. The running time and RMSE corresponds to regularization parameters $\lambda_1 = \frac{10^{1.1}}{\|\boldsymbol{\mathcal{X}}\|_F}$, $\lambda_2 = \frac{0.1}{\|\boldsymbol{\mathcal{X}}\|_F}$, $\lambda_{3,i} = 10^{0.7}$ for all $i = 1,\dots,N$ and ranks $R_1 = R_4 = 10$ and $R_2 = R_3 = 5$ which are determined by an exhaustive grid search over (37a) and (37a). The reason to use an exhaustive grid search is that other approaches such as a train-test procedure may not be suitable, when the size of the data set available is small and the estimated performance could be overly optimistic or overly pessimistic (Brownlee, 2020).

For each given ratio ρ, 5 test runs were conducted and RMSE is used to evaluate the performance. Table 1 and Figure 2 show that S-DCOT significantly outperforms its competitors.

5.1.2 Rainfall in India

We consider monthly rainfall data for different regions in India for the period 1901–2015, available from Kaggle[4]. For each of 36 regions, 12 months and 115 years, we have the total rainfall in millimeters. Since the monthly rainfall is similar within the time periods Jan-Mar, Apr-Jun, Jul-Sep and Oct-Dec, we consider the following partitions

$$\begin{aligned}
\boldsymbol{\mathcal{H}}(m,1,:) &= \boldsymbol{\mathcal{H}}(m,2,:),\\
\boldsymbol{\mathcal{H}}(m,3,:) &= \boldsymbol{\mathcal{H}}(m,4,:) = \boldsymbol{\mathcal{H}}(m,5,:),\\
\boldsymbol{\mathcal{H}}(m,6,:) &= \boldsymbol{\mathcal{H}}(m,7,:) = \boldsymbol{\mathcal{H}}(m,8,:),\\
\boldsymbol{\mathcal{H}}(m,9,:) &= \boldsymbol{\mathcal{H}}(m,10,:) = \boldsymbol{\mathcal{H}}(m,11,:) = \boldsymbol{\mathcal{H}}(m,12,:), \quad m = 1,2,\cdots,36,
\end{aligned}$$

4. https://www.kaggle.com/rajanand/rainfall-in-india

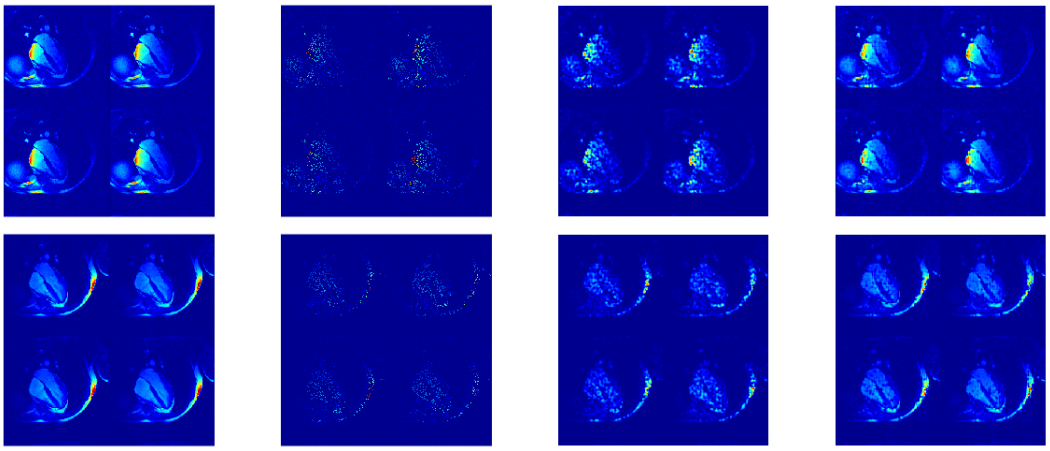

Figure 2: Tucker (third column) and DCOT (fourth column) completion results for a cine cardiac MRI series ($192 \times 192 \times 8 \times 19$) with 85% missing rate. The effect of using the supervised core is demonstrated (four images per channel). The first and second rows show some images from the channels one and two, respectively. The last column illustrates that S-DCOT outperforms the Tucker-based completion.

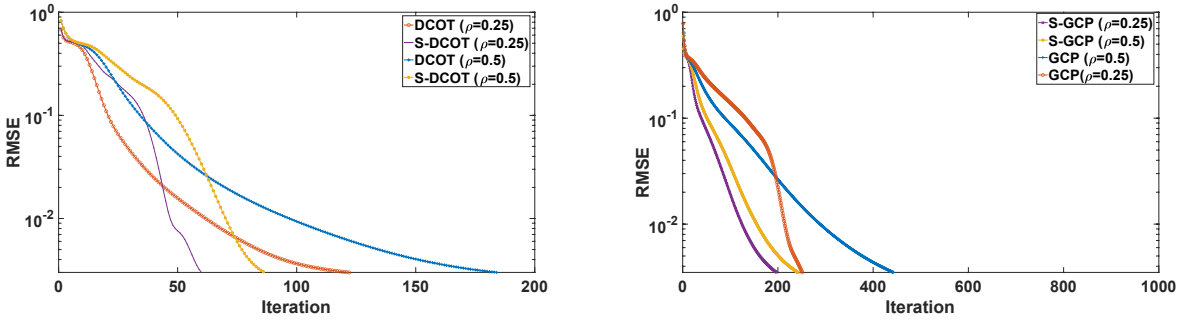

Figure 3: RMSE and the number of iterations of generalized tensor methods on rainfall dataset. **Left**: DCOT vs S-DCOT **Right**: GCP vs S-GCP.

which enforces similarity across members of $I_2 = 12$.

There are several distributions for Rainfall data. As mentioned previously, one option is to assume a Gaussian distribution but impose a nonnegativity constraint. Recently, Hong et al. (2019) showed that the Gamma distribution is potentially a reasonable model for this dataset. Hence, we investigate the performance of DCOT with smoothing loss (19) applied to Rainfall dataset and compare it with the GCP (Hong et al., 2019). For all completion algorithms, the regularization parameters and tensor ranks are determined by a grid search over (37a) and (37b) aiming to minimize the RMSE. We run each method with 5 different random starting points and report the average RMSE. Figure 3 indicates that the proposed S-DCOT has the best performance in terms of both RMSE and number of iterations.

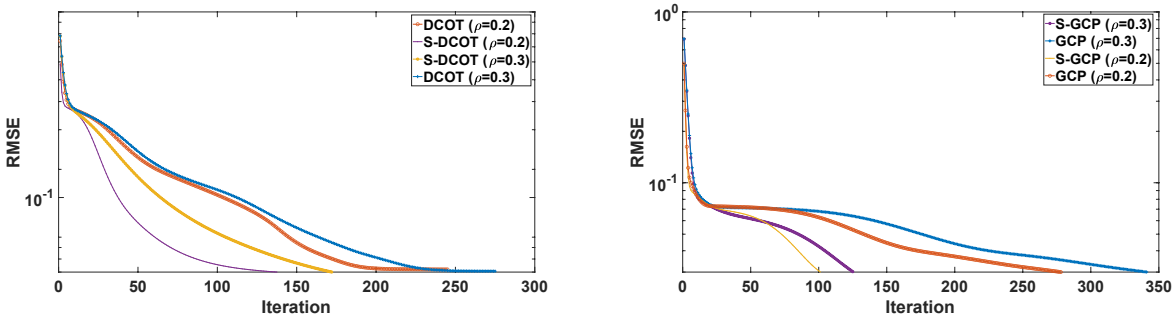

Figure 4: RMSE and the number of iterations of generalized tensor methods on count crime dataset. **Left**: DCOT vs S-DCOT **Right**: GCP vs S-GCP.

5.1.3 DCOT Applied to Sparse Count Crime Data

Next, we examine the performance of smooth DCOT factorization for completion and factorization of count datasets. To do so, we consider a real-world crime statistics dataset containing more than 15 years of crime data from the city of Chicago. The data[5] is organized as a 4-way tensor and obtained from FROSTT [6]. The tensor modes correspond to 6,186 days from 2001 to 2017, 24 hours per day, 77 communities, and 32 types of crimes. Each $\boldsymbol{\mathcal{X}}(i_1, i_2, i_3, :)$ is the number of times that a crime occurred in neighborhood i_3 during hour i_2 on day i_1. To enforce similarity within each community, we consider the following partitions

$$\boldsymbol{\mathcal{H}}_{m_\kappa} = \boldsymbol{\mathcal{H}}(:, m, \kappa, :), \quad \kappa = 1, \ldots, R_3, \quad m = 1, \ldots, R_1.$$

We use the DCOT model with the nonnegativity constraints and the proposed smoothing Poisson loss function defined in (17). We compare DCOT with GCP (Hong et al., 2019). For both DCOT and GCP and their smoothing variants, regularization parameters and ranks are determined by a grid search over (37a) and (37b). We run each method with 5 different random starting points and report the average RMSE.

The results are provided in Figure 4. The GCP and S-GCP methods descend much more quickly, but do not reduce the loss quite as much, though this failure to achieve the same final minimum is likely an artifact of the function estimation. On the other hand, Figure 4 indicates that the proposed S-DCOT has the best performance in terms of RMSE. The RMSE of the proposed method is less than that of S-GCP, illustrating that S-DCOT has better performance among the competing tensor factorization methods.

5.2 DCOT Applied to Multi-Platform Genomic Data

DCOT and S-DCOT models have been applied to understand latent relationships between patients and genes for multi-platform genomic data.

We use the PanCan12 dataset (Weinstein et al., 2013) and the Hallmark gene sets collections from MSigDB (Liberzon et al., 2015) for obtaining the input tensor and label

5. `www.cityofchicago.org`
6. `http://frostt.io/`

Dataset	S-DCOT(R)	DCOT(R)	S-Tucker(R)	Tucker(R)
PanCan12	23.79e-2 (0.009)	25.04e-2(0.008)	75.42e-2(0.007)	89.2e-4(0.005)

Table 2: RMSE of tensor methods applied to the PanCan12 dataset.

functions c_{i_n,j_n}, respectively. The PanCan12 contains multi-platform data with mapped clinical information of patient groups into cohorts of twelve cancer types including glioblastoma multiform, lymphoblastic acute myeloid leukemia, head and neck squamous carcinoma, lung adenocarcinoma, lung squamous carcinoma, breast carcinoma, kidney renal clear cell carcinoma, ovarian carcinoma, bladder carcinoma, colon adenocarcinoma, uterine cervical and endometrial carcinoma, and rectal adenocarcinoma. They are selected based on data maturity, adequate sample size, and publication or submission for publication of the primary analyses. The five Omics platforms used are miRNA-seq, methylation, somatic mutation, gene expression, and copy number variation.

The PanCan12 dataset was downloaded from the Sage Bionetworks repository by Synapse (Omberg et al., 2013) and was transformed to a 3rd-order tensor ($4555 \times 15351 \times 5$), containing 4555 samples, 14351 genes, and 5 Omics platforms. The data for each platform was min-max normalized and was further normalized such that the Frobenius norm became one. In order to efficiently fuse the date into the interpretable latent factors, we consider the following partitions

$$\boldsymbol{\mathcal{H}}(:,:,1) = \boldsymbol{\mathcal{H}}(:,:,1) = \cdots = \boldsymbol{\mathcal{H}}(:,:,5),$$

which enforces the similarity across third dimension, i.e., platform.

For the gene subgroups, we chose the Hallmark gene sets collection from MSigDB (Liberzon et al., 2015) and set $c_{i_n,j_n} = 1$ if genes belong to same subgroups. A gene smoothing function $s^h_{i_n,j_n}$ is generated in the form of gene-gene interaction within each subgroup. Test RMSE is used to measure the accuracy of tensor methods on this dataset. We split the data into a 50% training set, a 25% validation set and a 25% testing set, randomly. The regularization parameters and tensor ranks are determined by a grid search over (37a) and (37b) aiming to minimize the RMSE on the validation set. Table 2 indicates that the proposed DCOT has the best performance in terms of RMSE. The test RMSE of Tucker is $5\times$ higher than that of DCOT. Furthermore, test RMSE of S-DCOT is slightly higher or even better than that of Tucker and S-Tucker.

5.3 DCOT Applied to Recommender Systems

Next, we consider S-DCOT for recommender systems and compare it with five competing factorization methods. Three methods correspond to existing ones, namely, Bayesian probabilistic tensor factorization (BPTF) (Xiong et al., 2010), the factorization machine (libFM) (Rendle, 2012), and the Gaussian process factorization machine (GPFM) (Nguyen et al., 2014).[7] In addition, we also investigate the performance of the structured matrix factorization (MF) (Bi et al., 2017), and smooth neighborhood matrix factorization (S-MF)(Dai et al., 2019) with the proposed linearized ADMM.

7. The codes can be obtained from `https://www.cs.cmu.edu/~lxiong/bptf/bptf.html`, `http://www.libfm.org/`, and `http://trungngv.github.io/gpfm/`, respectively.

Dataset	S-DCOT(R)	libFM	GPFM	BPTF	S-MF	MF
MovieLens 1M	0.970(0.004)	0.989(0.006)	1.071(0.005)	1.027(0.007)	0.982(0.006)	1.056 (0.007)

Table 3: RMSE of completion methods on recommender systems dataset.

	Misspecification rate				
Dataset	5 %	10 %	15 %	20 %	30 %
MovieLens 1M	0.976(0.003)	0.981(0.005)	0.989(0.004)	0.983(0.005)	1.137 (0.005)

Table 4: RMSE of S-DCOT method on recommender systems dataset under 5%, 10%, 15%, 20%, and 30% subject (cluster) misspecification rate.

We apply the proposed method to MovieLens 1M data collected by GroupLens Research[8]. This dataset contains 1,000,209 ratings of 3883 movies by 6040 users, and rating scores range from 1 to 5. Also, the MovieLens 1M dataset provides demographic information for the users (age, gender, occupation, zipcode), genres, and release dates of the movies.

We define day as a context for the DCOT recommender model detailed in Subsection 2.3.1. Having the length of the context determined, we need to create *time bands* for days. Time bands specify the time resolution of a day, which are also data dependent. We can create time bands with equal or different length. For this dataset, we used time bands of 1 hours. Events are assigned to time bands according to their time stamp. Thus, we can create the `[user, item, day, time bands]` tensor. We factorize this tensor using the DCOT model and we get feature vectors for each user, for each item, for each day, and for each time bands.

Since we expect that at the same time offset in different days, the aggregated behavior of the users will be similar, we consider the following partitions

$$\boldsymbol{\mathcal{H}}_{m_\kappa} = \boldsymbol{\mathcal{H}}(:,:,m,\kappa), \quad \kappa = 1,\dots,R_4, \quad m = 1,\dots,R_3,$$

which enforces the similarity across members of R_4. For this application, in addition to the heterogeneous core $\boldsymbol{\mathcal{H}}$, we focus on employing a user-item smoothing function to solve the cold-start issue. We classify users based on the quantiles of the number of their ratings and set $c_{i_n,j_n} = 1$ if users belong to same clusters. On the other hand, the items are classified based on their release dates and $c_{i_n,j_n} = 1$ if they belong to same clusters.

We split the data into a 60% training set, a 15% validation set and a 25% testing set, randomly. The regularization parameters and tensor ranks are determined by a grid search over (37a) and (37b) aiming to minimize the RMSE on the validation set. Table 3 indicates that the proposed S-DCOT has the best performance in terms of RMSE. The RMSE of the proposed method is less than that of BPTF and libFM, illustrating that S-DCOT has better performance among the competing tensor factorization methods.

Next, we test the robustness of the proposed method when the clusters are misspecified. Specifically, we misassign users and items to adjacent clusters with 5%, 10%, 15%, 20%, and 30% chance and then construct the smoothing loss function and DCOT factorization. The results are summarized in Table 4 which shows that S-DCOT is robust against the misspecification of clusters. Indeed, in comparison with Table 3, S-DCOT method performs better than the other methods except when 30% of the cluster members are misclassified.

8. `http://grouplens.org/datasets/movielens`

5.4 DCOT Applied to Subspace Clustering and Dictionary Learning

Previous studies show that HOSVD is very powerful for clustering, especially in multiway data clustering tasks (Lu et al., 2011). It can achieve similar or better performance than most of the state-of-the-art clustering algorithms for mutliway data. Next, we evaluate the DCOT decomposition on a clustering problem. We compare DCOT with HOSVD and also four classical dimensionality reduction methods, including Principle Component Analysis (PCA) (Turk and Pentland, 1991), Linear Discriminant Analysis (LDA) (Belhumeur et al., 1997), Locality Preserving Projections (LPP) (He et al., 2005), and Marginal Fisher Analysis (MFA) (Yan et al., 2007).

The DCOT and competing methods are evaluated on the CMU and CASIA databases. The CASIA gait B database (Yu et al., 2006) comprises of indoor walking sequences from 124 subjects with 11 camera views and 10 clothing styles. We represent each walking sequence by the Gait Energy Image (Man and Bhanu, 2006), which is resized into size 128×88. All the images are vectorized, and the dataset is arranged as a fourth-order tensor, three for the latent factors and one for the feature dimension. Thus, the size of dataset is $(124 \times 11 \times 10 \times 11264\ )$.

To leverage the supervised information (i.e. subjects), we consider the following partitions

$$\begin{aligned} \boldsymbol{\mathcal{H}}_{m_\kappa} &= \boldsymbol{\mathcal{H}}(m, :, \kappa, :), \quad \kappa = 1, \ldots, 21, \quad m = 1, \ldots, 65, \quad \text{CMU database}, \\ \boldsymbol{\mathcal{H}}_{m_\kappa} &= \boldsymbol{\mathcal{H}}(:, m, \kappa, :), \quad \kappa = 1, \ldots, 11, \quad m = 1, \cdots, 10, \quad \text{CASIA database}. \end{aligned}$$

We randomly select ϖ subjects, where $\varpi = 10, 20, 30$, with 5 selected poses or illuminations in the CMU-PIE dataset and with 4 selected views or clothing styles in the CASIA dataset, respectively. The remaining samples in each database are used for testing. We employ a nearest neighbor classifier and repeat the procedure 5 times and average the results. To avoid the singularity of this problem, we use the first P principal coefficients determined by 95% energy for all the methods. Note that the MGE, LPP and MFA are manifold-based methods and need to determine the k nearest neighbors in their graphs.

The regularization parameters and tensor ranks are determined by a grid search over (37a) and (37b). The overall performance is given in Table 5. We consider the following cases: "untrained pose (UP)", "untrained illumination (UI)" that refers to a subset of testing data whose corresponding factors (pose or illumination) are not available during training.

DCOT achieves a high detection rate on the samples even when other complex factors are unobserved. DCOT significantly improves over multilinear-based methods and better interprets the cross-factor variation hidden in multi-factor data, even when the factor variation is not given in the training stage. We believe the benefits mainly come from the discriminative core tensor and the similarity function which exploit all of the latent factors to embed factor-dependent data pairs in a unified way.

6. Conclusion

A new tensor model was introduced for data analytic tasks for heterogeneous datasets, wherein there are joint low-rank structures within groups of observations, but also discriminative structures across different groups. The proposed model uses a double core tensor (DCOT) factorization together with a family of smoothing loss functions. By leveraging the proposed

Data		S-DCOT(H)	DCOT(H)	HOSVD	PCA	LPP	MFA	LDA
CMU	UP	36.79	38.44	29.10	36.16	32.81	34.24	33.19
	UI	96.41	93.57	90.39	75.39	83.90	92.14	89.15
CASIA	UP	70.24	69.38	66.29	55.34	58.18	50.29	62.17
	UI	89.73	85.33	83.19	80.33	84.34	82.29	83.21

Table 5: Average Detection Accuracy of Clustering (%) methods on the Face Databases.

smoothing function, the model accurately estimates the model factors, even in the presence of missing entries. A linearized ADMM method was developed to solve regularized versions of DCOT factorizations, that avoid large tensor operations and large memory storage requirements. Further, we established theoretically its global convergence, together with consistency of the estimates of the model parameters. The effectiveness of the DCOT model was illustrated on several heterogeneous tensor data.

Acknowledgements

The authors would like to acknowledge constructive and useful comments by two anonymous referees and the Action Editor. The work of Davoud Ataee Tarzanagh was supported by ARO YIP award W911NF1910027, AFOSR YIP award FA9550-19-1-0026, NSF BIGDATA award IIS-1838179 and a Fellowship from the University of Florida Informatics Institute. The work of George Michailidis was supported in part by NSF grants DMS 1854476 and 2210358 and NIH grant 1U01CA235487-01.

Appendices

Appendix A. Updating Parameters in Algorithm 1

This section provides detailed implementation of the linearized ADMM method for solving problem (4).

Proof of Lemma 5

Proof For simplicity, we assume that $N = 3$ and $\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{H}}$. We calculate the partial gradient of $\bar{\mathcal{L}}([\boldsymbol{\mathcal{S}}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}])$ with respect to $\mathbf{U}^{(1)}$ by chain rule:

$$\begin{aligned}
\frac{\partial \bar{\mathcal{L}}}{\partial \mathbf{U}^{(1)}_{ij}} &= \sum_{a=1}^{I_1}\sum_{b=1}^{I_2}\sum_{c=1}^{I_3} \frac{\partial \bar{\mathcal{L}}}{\partial \boldsymbol{\mathcal{T}}_{abc}} \cdot \frac{\partial \boldsymbol{\mathcal{T}}_{abc}}{\partial \mathbf{U}^{(1)}_{ij}} \\
&= \sum_{a=1}^{I_1}\sum_{b=1}^{I_2}\sum_{c=1}^{I_3} \frac{\partial \bar{\mathcal{L}}}{\partial \boldsymbol{\mathcal{T}}_{abc}} \cdot \left(1_{\{a=i\}} \sum_{r_2=1}^{R_2}\sum_{r_3=1}^{R_3} \boldsymbol{\mathcal{S}}_{jr_2r_3} \mathbf{U}^{(2)}_{br_2} \mathbf{U}^{(3)}_{cr_3} \right) \\
&= \sum_{b=1}^{I_2}\sum_{c=1}^{I_3} \frac{\partial \bar{\mathcal{L}}}{\partial \boldsymbol{\mathcal{T}}_{ibc}} \left(\sum_{r_2=1}^{R_2}\sum_{r_3=1}^{R_3} \boldsymbol{\mathcal{S}}_{jr_2r_3} \mathbf{U}^{(2)}_{br_2} \mathbf{U}^{(3)}_{cr_3} \right),
\end{aligned}$$

Table 6: Basic notation and product

$\boldsymbol{\mathcal{A}}$, $\mathbf{A}$, $\mathbf{a}$, a	tensor, matrix, vector, scalar
$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_R]$	matrix $\mathbf{A}$ with column vectors $\mathbf{a}_r$
$a_{i_1 \cdots i_N}$	entry of tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$
$\mathbf{a}(:, i_2, \cdots, i_N)$	fiber of tensor $\boldsymbol{\mathcal{A}}$ obtained by fixing all but one index
$\mathbf{A}(:, :, i_3, \cdots, i_N)$	matrix slice of tensor $\boldsymbol{\mathcal{A}}$ obtained by fixing all but two indices
$\boldsymbol{\mathcal{A}}(\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_N)$	subtensor of $\boldsymbol{\mathcal{A}}$ obtained by restricting indices to belong to subsets $\mathcal{I}_n \subseteq [I_n] \equiv \{1, 2, \ldots, I_n\}$
$\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N}$	mode-n matricization of tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ whose entry at row i_n and column $(i_1 - 1)I_2 \cdots I_{n-1} I_{n+1} \cdots I_N + \cdots + (i_{N-1} - 1)I_N + i_N$ is equal to $a_{i_1 \cdots i_N}$
$\text{vec}(\boldsymbol{\mathcal{A}}) \in \mathbb{R}^{I_N I_{N-1} \cdots I_1}$	vectorization of tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with the entry at position $i_1 + \sum_{k=2}^{N} [(i_k - 1) I_1 I_2 \cdots I_{k-1}]$ equal to $a_{i_1 \cdots i_N}$
$\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle := \text{vec}(\boldsymbol{\mathcal{X}})^T \text{vec}(\boldsymbol{\mathcal{Y}})$	inner product of two tensors $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$
$\mathbf{D} = \text{diag}(a_1, a_2, \cdots, a_R)$	diagonal matrix with $d_{rr} = a_r$
$\mathbf{A}^\top$, $\mathbf{A}^{-1}$, $\mathbf{A}^\dagger$	transpose, inverse, and Moore-Penrose pseudo-inverse
$\|\mathbf{X}\|_F$, $\|\mathbf{X}\|_*$, $\|\mathbf{X}\|_1$	Frobenius norm, the trace norm or trace norm as the sum of singular values of $\mathbf{X}$, and the ℓ_1 norm.
$\|\mathbf{X}\|_2$, $\|\mathbf{X}\|_{b,a}$,	The spectral norm as the largest singular value of matrix, the ℓ_a/ℓ_b norm as the ℓ_a norm of the vector formed by the ℓ_b norm of every row.
$\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}} \times_n \mathbf{U}$	mode-n product of $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\mathbf{U} \in \mathbb{R}^{j_N \times I_n}$ yields $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times j_N \times I_{n+1} \times \cdots \times I_N}$ with entries $c_{i_1 \cdots i_{n-1}\, j_N\, i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} a_{i_1 \cdots i_{n-1}\, i_n\, i_{n+1} \cdots i_N} b_{j_N\, i_n}$ and matrix representation $\mathbf{C}_{(n)} = \mathbf{U}\, \mathbf{A}_{(n)}$
$\boldsymbol{\mathcal{C}} = [\boldsymbol{\mathcal{A}}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(N)}]$	full multilinear product, $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)}$
$\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}} \circ \boldsymbol{\mathcal{B}}$	tensor or outer product of $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times j_N}$ yields $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times J_1 \times J_2 \times \cdots \times j_N}$ with entries $c_{i_1 \cdots i_N\, j_1 \cdots j_N} = a_{i_1 \cdots i_N} b_{j_1 \cdots j_N}$
$\boldsymbol{\mathcal{X}} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(N)}$	tensor or outer product of vectors $\mathbf{a}^{(n)} \in \mathbb{R}^{I_n}$ $(n = 1, \cdots, N)$ yields a rank-1 tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with entries
$\mathbf{C} = \mathbf{A} \otimes \mathbf{U}$	Kronecker product of $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2}$ and $\mathbf{U} \in \mathbb{R}^{J_1 \times J_2}$ yields $\mathbf{C} \in \mathbb{R}^{I_1 J_1 \times I_2 J_2}$ with entries $c_{(i_1-1)J_1 + j_1, (i_2-1)J_2 + j_2} = a_{i_1 i_2}\, b_{j_1 j_2}$

where the second identity comes from the following fact:

$$\boldsymbol{\mathcal{T}}_{abc} = \sum_{r_1=1}^{R_1}\sum_{r_2=1}^{R_2}\sum_{r_3=1}^{R_3} \boldsymbol{\mathcal{S}}_{r_1r_2r_3}\mathbf{U}^{(1)}_{ar_1}\mathbf{U}^{(2)}_{br_2}\mathbf{U}^{(3)}_{cr_3}.$$

Let $\boldsymbol{\mathcal{M}} = \nabla\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}})$. One can verify that

$$\Big(\boldsymbol{\mathcal{M}}_{(1)}(\mathbf{U}^{(3)}\otimes\mathbf{U}^{(2)})\mathbf{S}^{\top}_{(1)}\Big)_{ij} = \sum_{k_1=1}^{I_2}\sum_{k_2=1}^{I_3}\sum_{k_3=1}^{R_2}\sum_{k_4=1}^{R_3}\frac{\partial\bar{\mathcal{L}}}{\partial\boldsymbol{\mathcal{T}}_{ik_1k_2}}\cdot\mathbf{U}^{(2)}_{k_1k_3}\mathbf{U}^{(3)}_{k_2k_4}\boldsymbol{\mathcal{S}}_{jk_3k_4}.$$

Here, $\boldsymbol{\mathcal{M}}_{(1)}$ and $\mathbf{S}_{(1)}$ are mode-1 matricization of $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{S}}$, respectively.

The partial gradient for $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$ can be similarly calculated. For core tensor $\boldsymbol{\mathcal{S}}$, we have

$$\begin{aligned}
\frac{\partial\bar{\mathcal{L}}}{\partial\boldsymbol{\mathcal{S}}_{ijk}} &= \sum_{a=1}^{I_1}\sum_{b=1}^{I_2}\sum_{c=1}^{I_3}\frac{\partial\bar{\mathcal{L}}}{\partial\boldsymbol{\mathcal{T}}_{abc}}\cdot\frac{\partial\boldsymbol{\mathcal{T}}_{abc}}{\partial\boldsymbol{\mathcal{S}}_{ijk}}\\
&= \sum_{a=1}^{I_1}\sum_{b=1}^{I_2}\sum_{c=1}^{I_3}\frac{\partial\bar{\mathcal{L}}}{\partial\boldsymbol{\mathcal{T}}_{abc}}\cdot\mathbf{U}^{(1)}_{ai}\mathbf{U}^{(2)}_{bj}\mathbf{U}^{(3)}_{ck}\\
&= \Big(\nabla\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}});\mathbf{U}^{(1)},\mathbf{U}^{(2)},\mathbf{U}^{(3)}\rrbracket\Big)_{ijk}\\
&= \Big([\boldsymbol{\mathcal{M}};\mathbf{U}^{(1)},\mathbf{U}^{(2)},\mathbf{U}^{(3)}]\Big)_{ijk},
\end{aligned}$$

which has finished the proof of this lemma. ■

Updating $\mathbf{U}^{(n)}$

We need the gradient of the function $\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}})$ in (30) with respect to the factor matrices. It follows from Lemma 5 that

$$\nabla_{\mathbf{U}^{(n)}}\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\mathbf{U}_k^{(n)}}) = \nabla\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\mathbf{U}_k^{(n)}})(\bigotimes_{t\neq n}\mathbf{U}_k^{(t)})(\mathbf{H}_{k,(t)}+\mathbf{G}_{k,(t)})^{\top}. \tag{38}$$

Substituting (38) into (31a), we obtain

$$\begin{aligned}
\mathbf{U}^{(n)}_{k+1} &= \underset{\mathbf{U}^{(n)}}{\operatorname{argmin}} \quad \lambda_{3,n}J_{3,n}(\mathbf{U}^{(n)}) + \langle\nabla_{\mathbf{U}^{(n)}}\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\mathbf{U}_k^{(n)}}),\mathbf{U}^{(n)}-\mathbf{U}_k^{(n)}\rangle\\
&+ \frac{\varrho^n}{2}\|\mathbf{U}^{(n)}-\mathbf{U}_k^{(n)}\|_F^2,
\end{aligned} \tag{39}$$

where ϱ^n is a constant equal or greater than the Lipschitz of the gradient $\nabla\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}^{\mathbf{U}^{(n)}})$.

It follows from (5) that (39) has the following solution

$$\mathbf{U}^{(n)}_{k+1} = \operatorname{prox}^{J_{3,n}}_{\varrho^n}\Big(\mathbf{U}_k^{(n)} - \frac{1}{\varrho^n}\nabla_{\mathbf{U}^{(n)}}\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\mathbf{U}_k^{(n)}}),\frac{\lambda_{3,n}}{\varrho^n}\Big). \tag{40}$$

It is worth mentioning that we can use (40) for different choices of penalty functions. As discussed in (2.3), typical examples of the function $J_{3,n}(\mathbf{U}^{(n)})$ include $\|\mathbf{U}^{(n)}\|_1$, $\|\mathbf{U}^{(n)}\|_*$ or the indicator of a closed convex convex set. For example, if $J_{3,n}(\mathbf{U}^{(n)}) = \|\mathbf{U}^{(n)}\|_1$, then we can apply the ℓ_1 proxiaml operator (Parikh et al., 2014).

Updating $\mathcal{G}$

From (31b), we have

$$\mathcal{G}_{k+1} \;=\; \underset{\mathcal{G}}{\operatorname{argmin}} \;\; \langle \nabla \bar{\mathcal{L}}(\mathcal{T}_k^{\mathcal{G}}), \mathcal{G} - \mathcal{G}_k \rangle + \lambda_1 J_1(\mathcal{G}) + \frac{\varrho^g}{2} \|\mathcal{G} - \mathcal{G}_k\|_F^2$$

It follows from Lemma 5 that

$$\nabla_{\mathcal{G}} \bar{\mathcal{L}}(\mathcal{T}_k^{\mathcal{G}}) = \boldsymbol{\mathcal{M}}(\mathcal{T}_k^{\mathcal{G}}) \times_1 \mathbf{U}_{k+1}^{(1)} \cdots \times_N \mathbf{U}_{k+1}^{(N)},$$

where

$$\boldsymbol{\mathcal{M}}(\mathcal{T}_k^{\mathcal{G}}) = \gamma \Big((\mathcal{G}_k + \mathcal{H}_k) \times_1 \mathbf{U}_{k+1}^{(1)} \cdots \times_N \mathbf{U}_{k+1}^{(N)} - \boldsymbol{\mathcal{Z}}_k - \frac{1}{\gamma} \boldsymbol{\mathcal{W}}_k \Big). \tag{41}$$

Now, from (5), we obtain

$$\mathcal{G}_{k+1} = \operatorname{prox}_{\varrho^g}^{J_1} \big(\mathcal{G}_k - \frac{1}{\varrho^g} \nabla_{\mathcal{G}} \bar{\mathcal{L}}(\mathcal{T}_k^{\mathcal{G}_k}), \frac{\lambda_1}{\varrho^g} \big). \tag{42}$$

Updating $\mathcal{H}$

To minimize the sub-Lagrangian function w.r.t. $\mathcal{H}$, using (31c), we have

$$\mathcal{H}_{k+1} \;=\; \underset{\mathcal{H}}{\operatorname{argmin}} \;\; \langle \nabla \bar{\mathcal{L}}(\mathcal{T}_k^{\mathcal{H}_k}), \mathcal{H} - \mathcal{H}_k \rangle + \lambda_2 J_2(\mathcal{H}) + \frac{\varrho^h}{2} \|\mathcal{H} - \mathcal{H}_k\|_F^2.$$

This problem is separable w.r.t. $\{\mathcal{H}_{m_\pi}\}_{m=1}^M$ and we have that

$$\begin{aligned} \mathcal{H}_{m,k+1} = \underset{\mathcal{H}_{m_\pi}}{\operatorname{argmin}} \quad & \langle \nabla \bar{\mathcal{L}}(\mathcal{T}_k^{\mathcal{H}_{m_\pi}}), \mathcal{H}_{m_\pi} - \mathcal{H}_{m_\pi,k} \rangle \\ + \quad & \lambda_2 J_2(\mathcal{H}_{m_\pi}) + \frac{\varrho^h}{2} \|\mathcal{H}_{m_\pi} - \mathcal{H}_{m_\pi,k}\|_F^2. \end{aligned}$$

Let

$$\mathcal{R}_k^{\mathcal{G}} \;=\; \boldsymbol{\mathcal{Z}}_k - \gamma^{-1} \boldsymbol{\mathcal{W}}_k - [\mathcal{G}_{k+1}; \mathbf{U}_{k+1}^{(1)}, \ldots, \mathbf{U}_{k+1}^{(N)}]. \tag{43}$$

The gradient $\nabla \bar{\mathcal{L}}(\mathcal{T}_k^{\mathcal{H}_{m_\pi}})$ is equal to

$$\nabla_{\mathcal{H}_{m_\pi}} \Big(\frac{\gamma}{2} \|\mathcal{H} \times_1 \mathbf{U}_{k+1}^{(1)} \cdots \times_N \mathbf{U}_{k+1}^{(N)} - \mathcal{R}_k^{\mathcal{G}}\|_F \Big).$$

Let $\mathbf{r}_k^g = \operatorname{vec}(\mathcal{R}_k^{\mathcal{G}})$ and $\mathbf{h} = \operatorname{vec}(\mathcal{H})$. Then, we obtain

$$\|\mathbf{U}_{k+1}\mathbf{h} - \mathbf{r}_k^g\|_F^2 = \sum_{r_1 \cdots r_N} \sum_{i_1 \cdots i_N} ((\mathbf{U}_{k+1})_{i_1 i_2 \cdots i_N, r_1 \cdots r_N} \mathbf{h}_{r_1 \cdots r_N} - (\mathbf{r}_k^g)_{i_1 \cdots i_N})^2. \tag{44}$$

Hence

$$\begin{aligned}&\big(\nabla\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}^{\boldsymbol{\mathcal{H}}_{m_\pi}})\big)_{r_1\cdots r_N}\\ =\;&\gamma\sum_{r_1\cdots r_N\in\boldsymbol{\mathcal{H}}_m}\sum_{i_1\cdots i_N}(\mathbf{U}_{k+1})_{i_1\cdots i_N,r_1\cdots r_N}((\mathbf{U}_{k+1})_{i_1\cdots i_N r_1\cdots r_N}\mathbf{h}_{r_1\cdots r_N}-(\mathbf{r}_k^g)_{i_1\cdots i_N}).\end{aligned}\tag{45}$$

Using (45) and (31c), we have

$$\boldsymbol{\mathcal{H}}_{m_\pi,k+1}=\mathrm{Prox}_{\varrho^h}^{J_2}\Big(\boldsymbol{\mathcal{H}}_{m_\pi,k}-\frac{1}{\varrho^h}\nabla\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{H}}_{m_\pi,k}}),\frac{\lambda_2}{\varrho^h}\Big),\qquad m=1,2,\ldots,M.\tag{46}$$

Finally, we set

$$\boldsymbol{\mathcal{H}}_{m_1,k+1}=\boldsymbol{\mathcal{H}}_{m_2,k+1}=\cdots=\boldsymbol{\mathcal{H}}_{m_\pi,k+1}.\tag{47}$$

Updating $\boldsymbol{\mathcal{Z}}$

We derive an explicit formulation of the element-wise gradient w.r.t. $\boldsymbol{\mathcal{Z}}$ along with a straightforward way of handling missing data. To do so, from (28d), we have

$$\begin{aligned}\boldsymbol{\mathcal{Z}}_{k+1}\;=\;&\operatorname*{argmin}_{\boldsymbol{\mathcal{Z}}}\Big\{F(s^h,\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Z}})\\ &+\;\frac{\gamma}{2}\|\boldsymbol{\mathcal{Z}}-\gamma^{-1}\boldsymbol{\mathcal{W}}_k-(\boldsymbol{\mathcal{G}}_k+\boldsymbol{\mathcal{H}}_k)\times_1\mathbf{U}_{k+1}^{(1)}\cdots\times_N\mathbf{U}_{k+1}^{(N)}\|_F^2\Big\}.\end{aligned}\tag{48}$$

Next, we provide a generalized framework for computing gradients and handling missing data that enables the use of standard optimization methods for solving (48). Let

$$\begin{aligned}&\boldsymbol{\mathcal{R}}=\boldsymbol{\mathcal{Z}}_k-\gamma^{-1}\boldsymbol{\mathcal{W}}_k-(\boldsymbol{\mathcal{G}}_k+\boldsymbol{\mathcal{H}}_k)\times_1\mathbf{U}_{k+1}^{(1)}\cdots\times_N\mathbf{U}_{k+1}^{(N)},\quad\text{and}\\ &\boldsymbol{\mathcal{N}}=\nabla\bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{Z}}}).\end{aligned}$$

The gradient of the objective in (48) w.r.t. $z_{i_1\cdots i_N}\in\Omega$ is given by

$$\boldsymbol{\mathcal{N}}_{i_1\cdots i_N}\;=\;\nabla_{z_{i_1\cdots i_n}}F(s^h,\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Z}})+\gamma\boldsymbol{\mathcal{R}}_{z_{i_1\cdots i_N}},\tag{49}$$

where

$$F(s^h,\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Z}})=-\frac{1}{\prod_{n\in[N]}I_n}\sum_{i_1=1}^{I_1}\cdots\sum_{i_N=1}^{I_N}f_{i_1\cdots i_N}(s^h,\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Z}}).$$

Now, we can solve (48) via a gradient-based optimization method. In our implementation, we use the limited-memory BFGS method (Liu and Nocedal, 1989).

Appendix B. Convergence Analysis of Linearized ADMM

The next result shows that Algorithm 1 provides sufficient decrease of the augmented Lagrangian $\mathcal{L}(\boldsymbol{\mathcal{T}})$ in each iteration.

Lemma 14 *Let Assumption A hold, and $\gamma>L_F$, $\varrho^g>L^g,\varrho^h>L^h$, and $\varrho^n>L^n$ for $n=1,2,\ldots,N$, where L^n,L^g,L^h are Lipschitz constants of the gradients of Lagrangian*

function $\mathcal{L}$ w.r.t. $\{\mathbf{U}^{(n)}\}_{n=1}^N, \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{H}}$ and $\boldsymbol{\mathcal{Z}}$, respectively. Then, for the sequence $\{\boldsymbol{\mathcal{T}}_k\}_{k\geq 0}$ generated by Algorithm 1, we have

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\mathcal{T}}_k^{\mathbf{U}_k^{(n)}}) - \mathcal{L}(\boldsymbol{\mathcal{T}}_{k+1}^{\mathbf{U}_{k+1}^{(n)}}) &- \frac{\varrho^{(n)} - L^{(n)}}{2}\|\mathbf{U}_k^{(n)} - \mathbf{U}_{k+1}^{(n)}\|_F^2 \geq 0, \qquad n = 1, \ldots, N, \\
\mathcal{L}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{G}}_k}) - \mathcal{L}(\boldsymbol{\mathcal{T}}_{k+1}^{\boldsymbol{\mathcal{G}}_{k+1}}) &- \frac{\varrho^{g} - L^{g}}{2}\|\boldsymbol{\mathcal{G}}_k - \boldsymbol{\mathcal{G}}_{k+1}\|_F^2 \geq 0, \\
\mathcal{L}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{H}}_k}) - \mathcal{L}(\boldsymbol{\mathcal{H}}_{k+1}^{\boldsymbol{\mathcal{H}}_{k+1}}) &- \frac{\varrho^{h} - L^{h}}{2}\|\boldsymbol{\mathcal{H}}_k - \boldsymbol{\mathcal{H}}_{k+1}\|_F^2 \geq 0, \\
\mathcal{L}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{Z}}_k}) - \mathcal{L}(\boldsymbol{\mathcal{T}}_{k+1}^{\boldsymbol{\mathcal{Z}}_{k+1}}) &- \frac{\gamma - L_F}{2}\|\boldsymbol{\mathcal{Z}}_k - \boldsymbol{\mathcal{Z}}_{k+1}\|_F^2 \geq 0, \\
\mathcal{L}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{W}}_k}) - \mathcal{L}(\boldsymbol{\mathcal{T}}_{k+1}^{\boldsymbol{\mathcal{W}}_{k+1}}) &+ \frac{{L_F}^2}{\gamma}\|\boldsymbol{\mathcal{Z}}_{k+1} - \boldsymbol{\mathcal{Z}}_k\|_F^2 \geq 0.
\end{aligned} \tag{50}$$

Proof The first optimality condition for (48) is given by

$$\nabla F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}_{k+1}) - \boldsymbol{\mathcal{W}}_k - \gamma\Big((\boldsymbol{\mathcal{G}}_{k+1} + \boldsymbol{\mathcal{H}}_{k+1}) \times_1 \mathbf{U}_{k+1}^{(1)} \cdots \times_N \mathbf{U}_{k+1}^{(N)} - \boldsymbol{\mathcal{Z}}_{k+1}\Big) = 0$$

which together with (28e) implies that the iterative gap of dual variable can be bounded by that of primal variable, i.e.,

$$\boldsymbol{\mathcal{W}}_{k+1} = \nabla F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}_{k+1}).$$

Now, using Assumption A, we have

$$\begin{aligned}
\|\boldsymbol{\mathcal{W}}_{k+1} - \boldsymbol{\mathcal{W}}_k\|_F &\leq \|\nabla F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}_{k+1}) - \nabla F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}_k)\|_F \\
&\leq L_F\|\boldsymbol{\mathcal{Z}}_{k+1} - \boldsymbol{\mathcal{Z}}_k\|_F.
\end{aligned} \tag{51}$$

Hence, we obtain

$$\mathcal{L}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{W}}_k}) - \mathcal{L}(\boldsymbol{\mathcal{T}}_{k+1}^{\boldsymbol{\mathcal{W}}_{k+1}}) \geq \frac{1}{\gamma}\|\boldsymbol{\mathcal{W}}_{k+1} - \boldsymbol{\mathcal{W}}_k\|_F^2 \geq -\frac{{L_F}^2}{\gamma}\|\boldsymbol{\mathcal{Z}}_{k+1} - \boldsymbol{\mathcal{Z}}_k\|_F^2. \tag{52}$$

We note that the function $\mathcal{L}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{Z}}})$ in (48) is strongly convex w.r.t. $\boldsymbol{\mathcal{Z}}$ whenever $\gamma > L_F$, where L_F is a Lipschitz constant for $\nabla F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}})$; see, Assumption A. Thus, we have

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\mathcal{T}}_{k+1}^{\boldsymbol{\mathcal{Z}}_{k+1}}) - \mathcal{L}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{Z}}_k}) &= F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}_{k+1}) - F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}_k) \\
&\quad + \frac{\gamma}{2}\|\big(\boldsymbol{\mathcal{G}}_{k+1} + \boldsymbol{\mathcal{H}}_{k+1}\big) \times_1 \mathbf{U}_{k+1}^{(1)} \cdots \times_N \mathbf{U}_{k+1}^{(N)} - \boldsymbol{\mathcal{Z}}_{k+1} - \frac{1}{\gamma}\boldsymbol{\mathcal{W}}_{k+1}\|_F^2 \\
&\quad - \frac{\gamma}{2}\|\big(\boldsymbol{\mathcal{G}}_k + \boldsymbol{\mathcal{H}}_k\big) \times_1 \mathbf{U}_k^{(1)} \cdots \times_N \mathbf{U}_k^{(N)} - \boldsymbol{\mathcal{Z}}_k - \frac{1}{\gamma}\boldsymbol{\mathcal{W}}_k\|_F^2 \\
&\leq \langle \nabla\mathcal{L}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{Z}}_{k+1}}), \boldsymbol{\mathcal{Z}}_{k+1} - \boldsymbol{\mathcal{Z}}_k\rangle - \frac{\gamma - L_F}{2}\|\boldsymbol{\mathcal{Z}}_{k+1} - \boldsymbol{\mathcal{Z}}_k\|_F^2, \\
&= -\frac{\gamma - L_F}{2}\|\boldsymbol{\mathcal{Z}}_{k+1} - \boldsymbol{\mathcal{Z}}_k\|_F^2,
\end{aligned}$$

where the last equality follows from the first-order optimality condition for (28d).

The remainder of the proof of this lemma follows along similar lines to the proof of Bolte et al. (2014, Lemma 1, p. 470). ■

The following lemma shows that the augmented Lagrangian has a sufficient decrease in each iteration and it is uniformly lower bounded.

Lemma 15 *Let $\{\boldsymbol{\mathcal{T}}_k\}_{k\geq 0}$ be a sequence generated by Algorithm 1, and set*

$$\begin{aligned}\boldsymbol{\mathcal{D}}_k &= \sum_{n=1}^{N} \|\mathbf{U}_k^{(n)} - \mathbf{U}_{k+1}^{(n)}\|_F^2 + \|\boldsymbol{\mathcal{G}}_k - \boldsymbol{\mathcal{G}}_{k+1}\|_F^2 \\ &+ \|\boldsymbol{\mathcal{H}}_k - \boldsymbol{\mathcal{H}}_{k+1}\|_F^2 + \|\boldsymbol{\mathcal{Z}}_k - \boldsymbol{\mathcal{Z}}_{k+1}\|_F^2 + \|\boldsymbol{\mathcal{W}}_k - \boldsymbol{\mathcal{W}}_{k+1}\|_F^2. \end{aligned} \tag{53}$$

Then, there exists a positive constant ϑ such that

$$\mathcal{L}(\boldsymbol{\mathcal{T}}_{k+1}) \leq \mathcal{L}(\boldsymbol{\mathcal{T}}_k) - \frac{\vartheta}{2} D_k, \qquad \text{for all } \ k \geq 0, \tag{54}$$

and

$$D_k \leq \frac{2}{\vartheta}(\mathcal{L}(\boldsymbol{\mathcal{T}}_0) - \underline{\boldsymbol{\mathcal{L}}}). \tag{55}$$

Here, $\underline{\boldsymbol{\mathcal{L}}}$ is the uniform lower bound of $\mathcal{L}(\boldsymbol{\mathcal{T}}_k)$.

Proof Let $\bar{\gamma} = \frac{\gamma^2 - \gamma L_F - 2{L_F}^2}{2\gamma}$ and

$$\begin{aligned}\hat{\gamma} &= \max(\varrho^h - L^h, \varrho^g - L^g, \varrho^{(1)} - L^{(1)}, \ldots, \varrho^{(n)} - L^{(n)}), \\ \vartheta &= \max(\hat{\gamma}, \bar{\gamma}), \end{aligned} \tag{56}$$

where γ is a dual step-size.

We note that the roots of the quadratic equation $\gamma^2 - \gamma L_F - 2{L_F}^2 = 0$ are $-L_F$ and $2L_F$. Hence, ours choice of dual step size ($\gamma > 2L$) and regularization parameters in Algorithm 1 implies that $\vartheta > 0$. Now, using Lemma 14, we have

$$\begin{aligned}\mathcal{L}(\boldsymbol{\mathcal{T}}_k) - \mathcal{L}(\boldsymbol{\mathcal{T}}_{k+1}) &\geq \sum_{n=1}^{N} \frac{\varrho^n - L^n}{2} \|\mathbf{U}_k^{(n)} - \mathbf{U}_k^{(n+1)}\|_F^2 \\ &+ \frac{\varrho^h - L^h}{2} \|\boldsymbol{\mathcal{H}}_k - \boldsymbol{\mathcal{H}}_{k+1}\|_F^2 + \frac{\varrho^g - L^g}{2} \|\boldsymbol{\mathcal{G}}_k - \boldsymbol{\mathcal{G}}_{k+1}\|_F^2 \\ &+ \frac{\gamma - L_F}{2} \|\boldsymbol{\mathcal{Z}}_k - \boldsymbol{\mathcal{Z}}_{k+1}\|_F^2 - \frac{{L_F}^2}{\gamma} \|\boldsymbol{\mathcal{Z}}_{k+1} - \boldsymbol{\mathcal{Z}}_k\| \\ &\geq \frac{\hat{\gamma}}{2} \Big(\sum_{n=1}^{N} \|\mathbf{U}_k^{(n)} - \mathbf{U}_{k+1}^{(n)}\|_F^2 + \|\boldsymbol{\mathcal{H}}_k - \boldsymbol{\mathcal{H}}_{k+1}\|_F^2 + \|\boldsymbol{\mathcal{G}}_k - \boldsymbol{\mathcal{G}}_{k+1}\|_F^2\Big) \\ &+ \frac{\gamma^2 - \gamma L_F - 2{L_F}^2}{2\gamma} \cdot \frac{1}{2} \big(\|\boldsymbol{\mathcal{Z}}_k - \boldsymbol{\mathcal{Z}}_{k+1}\|_F^2 + \|\boldsymbol{\mathcal{Z}}^k - \boldsymbol{\mathcal{Z}}_{k+1}\|_F^2\big) \\ &\geq \frac{\vartheta}{2} \Big(\sum_{n=1}^{N} \|\mathbf{U}_k^{(n)} - \mathbf{U}_k^{(n+1)}\|_F^2 + \|\boldsymbol{\mathcal{H}}_k - \boldsymbol{\mathcal{H}}_{k+1}\|_F^2 + \|\boldsymbol{\mathcal{G}}_k - \boldsymbol{\mathcal{G}}_{k+1}\|_F^2 \\ &+ \|\boldsymbol{\mathcal{Z}}_k - \boldsymbol{\mathcal{Z}}_{k+1}\|_F^2 + \|\boldsymbol{\mathcal{W}}_k - \boldsymbol{\mathcal{W}}_{k+1}\|_F^2\Big), \end{aligned}$$

where the last inequality follows from (56).

To show (55), let $\tilde{\boldsymbol{\mathcal{Z}}} = (\tilde{\boldsymbol{\mathcal{G}}} + \tilde{\boldsymbol{\mathcal{H}}}) \times_1 \tilde{\mathbf{U}}^{(1)} \cdots \times_N \tilde{\mathbf{U}}^{(N)}$. Using Assumption A, we have

$$\begin{aligned} F(s^h, \boldsymbol{\mathcal{X}}; \tilde{\boldsymbol{\mathcal{Z}}}_{k+1}) &\leq F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}_k) + \langle \nabla F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}_{k+1}), \tilde{\boldsymbol{\mathcal{Z}}}_{k+1} - \boldsymbol{\mathcal{Z}}_{k+1} \rangle \\ &+ \frac{L_F}{2} \|\tilde{\boldsymbol{\mathcal{Z}}}_{k+1} - \boldsymbol{\mathcal{Z}}_{k+1}\|_F^2, \\ &= F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}_k) + \langle \boldsymbol{\mathcal{W}}, \tilde{\boldsymbol{\mathcal{Z}}}_{k+1} - \boldsymbol{\mathcal{Z}}_{k+1} \rangle + \frac{L_F}{2} \|\tilde{\boldsymbol{\mathcal{Z}}}_{k+1} - \boldsymbol{\mathcal{Z}}_{k+1}\|_F^2. \quad (57) \end{aligned}$$

By Assumption B, penalty functions $\{J_1(.), J_2(.), J_{3,1}(.), J_{3,2}(.), \cdots, J_{3,N}(.)\}$ and the loss function F are lower bounded. Now, since $\gamma > L_F$, we get

$$\begin{aligned} \mathcal{L}(\boldsymbol{\mathcal{T}}_{k+1}) &\geq F(s^h, \boldsymbol{\mathcal{X}}; \tilde{\boldsymbol{\mathcal{Z}}}_{k+1}) + \lambda_1 J_1(\boldsymbol{\mathcal{G}}_{k+1}) + \lambda_2 J_2(\boldsymbol{\mathcal{G}}_{k+1}) \\ &+ \lambda_3 J_3(\mathbf{U}_{k+1}) + \frac{\gamma - L_F}{2} \|\tilde{\boldsymbol{\mathcal{Z}}}_{k+1} - \boldsymbol{\mathcal{Z}}_{k+1}\|_F^2, \quad (58) \\ &\geq \underline{F} + \underline{J_1} + \underline{J_2} + \underline{J_3} \geq \underline{\boldsymbol{\mathcal{L}}}, \end{aligned}$$

where where $\underline{\boldsymbol{\mathcal{L}}}$ is the uniform lower bound of $\mathcal{L}(\boldsymbol{\mathcal{T}}_k)$.

Finally, using (54), we obtain (55). ■

Next, we give a formal definition of the limit point set. Let the sequence $\{\boldsymbol{\mathcal{T}}_k\}_{k\geq 0}$ be a sequence generated by the Algorithm 1 from a starting point $\boldsymbol{\mathcal{T}}_0$. The set of all limit points is denoted by $\Upsilon(\boldsymbol{\mathcal{T}}_0)$, i.e.,

$$\Upsilon(\boldsymbol{\mathcal{T}}_0) = \left\{ \bar{\boldsymbol{\mathcal{T}}} : \exists \text{ an infinite sequence } \{\boldsymbol{\mathcal{T}}_{k_s}\}_{s\geq 0} \text{ such that } \boldsymbol{\mathcal{T}}_{k_s} \to \bar{\boldsymbol{\mathcal{T}}} \text{ as } s \to \infty \right\}. \quad (59)$$

We now show that the set of accumulations points of the sequence $\{\boldsymbol{\mathcal{T}}_k\}_{k\geq 0}$ generated by Algorithm 1 is nonempty and it is a subset of the critical points of $\mathcal{L}$.

Lemma 16 *Let $\{\boldsymbol{\mathcal{T}}_k\}_{k\geq 0}$ be a sequence generated by Algorithm 1. Then,*
(i) $\Upsilon(\boldsymbol{\mathcal{T}}_0)$ is a non-empty set, and any point in $\Upsilon(\boldsymbol{\mathcal{T}}_0)$ is a critical point of $\mathcal{L}(\boldsymbol{\mathcal{T}})$;
(ii) $\Upsilon(\boldsymbol{\mathcal{T}}_0)$ is a compact and connected set;
(iii) The function $\mathcal{L}(\boldsymbol{\mathcal{T}})$ is finite and constant on $\Upsilon(\boldsymbol{\mathcal{T}}_0)$.

Proof (i). It follows from (55) that the sequence $\{\boldsymbol{\mathcal{T}}_k\}_{k\geq 0}$ is bounded which implies that $\Upsilon(\boldsymbol{\mathcal{T}}_0)$ is non-empty due to the Bolzano-Weierstrass Theorem. Consequently, there exists a sub-sequence $\{\boldsymbol{\mathcal{T}}_{k_s}\}_{s\geq 0}$, such that

$$\boldsymbol{\mathcal{T}}_{k_s} \to \boldsymbol{\mathcal{T}}_*, \qquad \text{as} \quad s \to \infty. \quad (60)$$

Since $J_{3,n}$ is lower semi-continuous, (60) yields

$$\liminf_{s\to\infty} J_{3,n}(\mathbf{U}_{k_s}^{(n)}) \geq J_{3,n}(\mathbf{U}_*^{(n)}). \quad (61)$$

Further, from the iterative step (31a), we have

$$\mathbf{U}_{k+1}^{(n)} = \underset{\mathbf{U}^{(n)}}{\operatorname{argmin}} \quad \lambda_{3,n} J_{3,n}(\mathbf{U}^{(n)}) + \langle \nabla_{\mathbf{U}^{(n)}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\mathbf{U}_k^{(n)}}), \mathbf{U}^{(n)} - \mathbf{U}_k^{(n)} \rangle + \frac{\varrho^n}{2} \|\mathbf{U}^{(n)} - \mathbf{U}_k^{(n)}\|_F^2.$$

Thus, letting $\mathbf{U}^{(n)} = \mathbf{U}_*^{(n)}$ in the above, we get

$$\begin{aligned} & \lambda_{3,n} J_{3,n}(\mathbf{U}_{k+1}^{(n)}) + \langle \nabla_{\mathbf{U}^{(n)}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\mathbf{U}_k^{(n)}}), \mathbf{U}_{k+1}^{(n)} - \mathbf{U}_k^{(n)} \rangle + \frac{\varrho^n}{2} \|\mathbf{U}_{k+1}^{(n)} - \mathbf{U}_k^{(n)}\|_F^2, \\ \leq \quad & \lambda_{3,n} J_{3,n}(\mathbf{U}_*^{(n)}) + \langle \nabla_{\mathbf{U}^{(n)}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\mathbf{U}_k^{(n)}}), \mathbf{U}_*^{(n)} - \mathbf{U}_k^{(n)} \rangle + \frac{\varrho^n}{2} \|\mathbf{U}_*^{(n)} - \mathbf{U}_k^{(n)}\|_F^2, \end{aligned} \tag{62}$$

Choosing $k = k_s - 1$ in the above inequality and letting s goes to ∞, we obtain

$$\limsup_{s \to \infty} J_{3,n}(\mathbf{U}_{k_s}^{(n)}) \leq J_{3,n}(\mathbf{U}_*^{(n)}). \tag{63}$$

Here, we have used the fact that $\nabla_{\mathbf{U}^{(n)}} \bar{\mathcal{L}}$ is a gradient Lipchitz continuous function w.r.t. $\mathbf{U}^{(n)}$, the sequence $\mathbf{U}_k^{(n)}$ is bounded and that the distance between two successive iterates tends to zero; see, (55). Now, we combine (61) and (63) to obtain

$$\lim_{s \to \infty} J_{3,n}(\mathbf{U}_{k_s}^{(n)}) = J_{3,n}(\mathbf{U}_*^{(n)}) \quad \text{for all} \quad n \in N. \tag{64}$$

Arguing similarly with other variables, we obtain

$$\begin{aligned} \lim_{s \to \infty} J_1(\boldsymbol{\mathcal{G}}_{k_s}) &= J_1(\boldsymbol{\mathcal{G}}_*), && \text{(65a)} \\ \lim_{s \to \infty} J_2(\boldsymbol{\mathcal{H}}_{k_s}) &= J_2(\boldsymbol{\mathcal{H}}_*), && \text{(65b)} \\ \lim_{s \to \infty} F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}_{k_s}) &= F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}_*), && \text{(65c)} \\ \lim_{s \to \infty} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_{k_s}) &= \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}^*), && \text{(65d)} \end{aligned}$$

where (65a) and (65b) follow since J_1 and J_2 are lower semi-continuous; (65c) and (65d) are obtained from the continuity of functions F and $\bar{\mathcal{L}}$. Thus, $\lim_{s\to\infty} \mathcal{L}(\boldsymbol{\mathcal{T}}_{k_s}) = \mathcal{L}(\boldsymbol{\mathcal{T}}_*)$.

Next, we show that $\boldsymbol{\mathcal{T}}_*$ is a critical point of $\mathcal{L}(.)$. By the first-order optimality condition for the augmented Lagrangian function in (26), we have

$$\begin{aligned} & \partial J_{3,n}(\mathbf{U}_{k+1}^{(n)}) + \nabla_{\mathbf{U}^{(n)}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_{k+1}^{\mathbf{U}_{k+1}^{(n)}}) \in \partial_{\mathbf{U}^{(n)}} \mathcal{L}(\mathbf{U}_{k+1}^{(n)}), \qquad n = 1, \ldots, N, \\ & \partial J_2(\boldsymbol{\mathcal{H}}_{k+1}) + \nabla_{\boldsymbol{\mathcal{H}}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_{k+1}^{\boldsymbol{\mathcal{H}}_{k+1}}) \in \partial_{\boldsymbol{\mathcal{H}}} \mathcal{L}(\boldsymbol{\mathcal{H}}_{k+1}), \\ & \partial J_1(\boldsymbol{\mathcal{G}}_{k+1}) + \nabla_{\boldsymbol{\mathcal{G}}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_{k+1}^{\boldsymbol{\mathcal{G}}_{k+1}}) \in \partial_{\boldsymbol{\mathcal{G}}} \mathcal{L}(\boldsymbol{\mathcal{G}}_{k+1}), \\ & \nabla_{\boldsymbol{\mathcal{Z}}} F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}_{k+1}) + \nabla_{\boldsymbol{\mathcal{Z}}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_{k+1}^{\boldsymbol{\mathcal{Z}}_{k+1}}) = \nabla_{\boldsymbol{\mathcal{Z}}} \mathcal{L}(\boldsymbol{\mathcal{Z}}_{k+1}), \\ & \gamma\big((\boldsymbol{\mathcal{G}}_{k+1} + \boldsymbol{\mathcal{H}}_{k+1}) \times_1 \mathbf{U}_{k+1}^{(1)} \cdots \times_N \mathbf{U}_{k+1}^{(N)} - \boldsymbol{\mathcal{Z}}_{k+1}\big) = -\nabla_{\boldsymbol{\mathcal{W}}} \mathcal{L}(\boldsymbol{\mathcal{W}}^{k+1}). \end{aligned} \tag{66}$$

Similarly, by the first-order optimality condition for subproblems (31a)–(28d), we have

$$\begin{aligned} & \partial J_{3,n}(\mathbf{U}_{k+1}^{(n)}) + \nabla_{\mathbf{U}^{(n)}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\mathbf{U}_k^{(n)}}) + \rho^{(n)}(\mathbf{U}_k^{(n)} - \mathbf{U}_{k+1}^{(n)}) = 0, \qquad n = 1, \ldots, N, \\ & \partial J_2(\boldsymbol{\mathcal{H}}_{k+1}) + \nabla_{\boldsymbol{\mathcal{H}}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{H}}_k}) + \rho^{\boldsymbol{\mathcal{H}}}(\boldsymbol{\mathcal{H}}_k - \boldsymbol{\mathcal{H}}_{k+1}) = 0, \\ & \partial J_1(\boldsymbol{\mathcal{G}}_{k+1}) + \nabla_{\boldsymbol{\mathcal{G}}} \bar{\mathcal{L}}(\boldsymbol{\mathcal{T}}_k^{\boldsymbol{\mathcal{G}}_k}) + \rho^{\boldsymbol{\mathcal{G}}}(\boldsymbol{\mathcal{G}}_k - \boldsymbol{\mathcal{G}}_{k+1}) = 0, \\ & \nabla_{\boldsymbol{\mathcal{Z}}} F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}_{k+1}) = 0. \end{aligned} \tag{67}$$

Combine (66) with (67) to obtain

$$(\xi_{k+1}^1, \ldots, \xi_{k+1}^N, \xi_{k+1}^{\mathcal{G}}, \xi_{k+1}^{\mathcal{H}}, \xi_{k+1}^{\mathcal{Z}}, \xi_{k+1}^{\mathcal{W}}) \in \partial\mathcal{L}(\mathcal{T}_{k+1}), \tag{68}$$

where

$$\begin{aligned}
\xi_{\mathbf{U}^{(n)}}^{k+1} &:= \nabla_{\mathbf{U}^{(n)}}\bar{\mathcal{L}}(\mathcal{T}_{k+1}^{\mathbf{U}_{k+1}^{(n)}}) - \nabla_{\mathbf{U}^{(n)}}\bar{\mathcal{L}}(\mathcal{T}_{k}^{\mathbf{U}_{k}^{(n)}}) - \rho^{(n)}(\mathbf{U}_k^{(n)} - \mathbf{U}_{k+1}^{(n)}), \qquad n = 1, \ldots, N,\\
\xi_{\mathcal{H}}^{k+1} &:= \nabla_{\mathcal{H}}\bar{\mathcal{L}}(\mathcal{T}_{k+1}^{\mathcal{H}_{k+1}}) - \nabla_{\mathcal{H}}\bar{\mathcal{L}}(\mathcal{T}_{k}^{\mathcal{H}_{k}}) - \rho^{\mathcal{H}}(\mathcal{H}_k - \mathcal{H}_{k+1}),\\
\xi_{\mathcal{G}}^{k+1} &:= \nabla_{\mathcal{G}}\bar{\mathcal{L}}(\mathcal{T}_{k+1}^{\mathcal{G}_{k+1}}) - \nabla_{\mathcal{G}}\bar{\mathcal{L}}(\mathcal{T}_{k}^{\mathcal{G}_{k}}) - \rho^{\mathcal{G}}(\mathcal{G}_k - \mathcal{G}_{k+1}),\\
\xi_{\mathcal{Z}}^{k+1} &:= \nabla_{\mathcal{Z}}\bar{\mathcal{L}}(\mathcal{T}_{k+1}^{\mathcal{Z}_{k+1}}) = \mathcal{W}_k - \mathcal{W}_{k+1},\\
\xi_{\mathcal{W}}^{k+1} &:= \frac{1}{\gamma}(\mathcal{W}_k - \mathcal{W}_{k+1}).
\end{aligned} \tag{69}$$

Note that the function $\bar{\mathcal{L}}(\mathcal{T})$ defined in (30) is a gradient Lipchitz continuous function w.r.t. $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)}, \mathcal{G}, \mathcal{H}$. Thus,

$$\begin{aligned}
\|\nabla_{\mathbf{U}^{(n)}}\bar{\mathcal{L}}(\mathcal{T}_{k+1}^{\mathbf{U}_{k+1}^{(n)}}) - \nabla_{\mathbf{U}^{(n)}}\bar{\mathcal{L}}(\mathcal{T}_{k}^{\mathbf{U}_{k}^{(n)}})\| &\leq \rho^{(n)}\|\mathbf{U}_k^{(n)} - \mathbf{U}_{k+1}^{(n)}\|, \qquad n = 1, \ldots, N,\\
\|\nabla_{\mathcal{H}}\bar{\mathcal{L}}(\mathcal{T}_{k+1}^{\mathcal{H}_{k+1}}) - \nabla_{\mathcal{H}}\bar{\mathcal{L}}(\mathcal{T}_{k}^{\mathcal{H}_{k}})\| &\leq \rho^{\mathcal{H}}\|\mathcal{H}_k - \mathcal{H}_{k+1}\|,\\
\|\nabla_{\mathcal{G}}\bar{\mathcal{L}}(\mathcal{T}_{k+1}^{\mathcal{G}_{k+1}}) - \nabla_{\mathcal{G}}\bar{\mathcal{L}}(\mathcal{T}_{k}^{\mathcal{G}_{k}})\| &\leq \rho^{\mathcal{G}}\|\mathcal{G}_k - \mathcal{G}_{k+1}\|,
\end{aligned} \tag{70}$$

Using (70), (55) and (69), we obtain

$$\lim_{k\to\infty} \big(\|\xi_{k+1}^1\|, \ldots, \|\xi_{k+1}^N\|, \|\xi_{k+1}^{\mathcal{G}}\|, \|\xi_{k+1}^{\mathcal{H}}\|, \|\xi_{k+1}^{\mathcal{Z}}\|, \|\xi_{k+1}^{\mathcal{W}}\|\big) = \big(0, \ldots, 0\big). \tag{71}$$

Now, from (68) and (71), we conclude that $(0, \ldots, 0) \in \partial\mathcal{L}(\mathcal{T}_*)$ due to the closedness property of $\partial\mathcal{L}$. Therefore, $\mathcal{T}_*$ is a critical point of $\mathcal{L}(.)$. This completes the proof of (i).
(ii). The proof follows from Bolte et al. (2014, Lemma 5 and Remark 5).
(iii). Let $\mathcal{L}_* = \lim_{k\to\infty} \mathcal{L}(\mathcal{T}_k)$. Choose $\mathcal{T}_* \in \Upsilon(\mathcal{T}_0)$. There exists a subsequence $\mathcal{T}_{k_s}$ converging to $\mathcal{T}_*$ as s goes to infinity. Since we have proven that $\lim_{s\to\infty} \mathcal{L}(\mathcal{T}_{k_s}) = \mathcal{L}(\mathcal{T}_*)$, and $\mathcal{L}(\mathcal{T}_k)$ is a non-increasing sequence, we conclude that $\mathcal{L}(\mathcal{T}_*) = \mathcal{L}_*$, hence the restriction of $\mathcal{L}(\mathcal{T})$ to $\Upsilon(\mathcal{T}_0)$ equals $\mathcal{L}_*$. ■

Proof of Theorem 7

Proof The augmented Lagrangian function $\mathcal{L}(\mathcal{T})$ is a Kurdyka-Lojasiewicz function. Further, by Lemma 16, $\mathcal{L}(\mathcal{T})$ is constant on $\Upsilon(\mathcal{T}_0)$ and the set $\Upsilon(\mathcal{T}_0)$ defined in (59) is compact. Putting all these together, the proof of this theorem follows along similar lines to the proof of Bolte et al. (2014, Theorem 1), with function $\psi(x)$ replaced by $\mathcal{L}(\mathcal{T})$. ■

Appendix C. Consistency of DCOT

In this section, we provide the proof of Theorem 11 and Corollary 12.

Proof of Theorem 11

Proof We bound empirical processes induced by $\rho(.,.)$ defiend in (33) by a chaining argument as in Wong et al. (1995); Shen (1998). We define a partition of $\Gamma(\boldsymbol{\mathcal{Z}})$ (the parameter space of $\boldsymbol{\mathcal{Z}}$) as follows:

$$A(\zeta) \quad = \quad \Big\{\boldsymbol{\mathcal{Z}} \in \Gamma(\boldsymbol{\mathcal{Z}}) : \zeta \leq \rho(\boldsymbol{\mathcal{Z}}^*, \boldsymbol{\mathcal{Z}}) \leq 2\zeta\Big\}. \tag{72}$$

From the definition of $\rho(.,.)$ in (33), for any $\boldsymbol{\mathcal{Z}} \in A(\zeta)$, we have

$$\zeta^2 \leq \frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} (z^*_{i_1\cdots i_N} - z_{i_1\cdots i_N})^2 \leq 4\zeta^2. \tag{73}$$

Since $\widehat{\boldsymbol{\mathcal{Z}}}$ is a minimizer of (4), we obtain

$$\begin{aligned} &P\Big(\rho(\widehat{\boldsymbol{\mathcal{Z}}}, \boldsymbol{\mathcal{Z}}^*)) \geq \eta\Big) \\ &\leq P^*\Big(\sup_{\boldsymbol{\mathcal{Z}}\in\bigcup_\zeta A(\zeta)} (F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}^*) - F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) + \lambda(J(\boldsymbol{\mathcal{Z}}^*) - J(\boldsymbol{\mathcal{Z}})) \geq 0\Big), \end{aligned} \tag{74}$$

where P^* denotes the outer probability measure Billingsley (2013).

From the definition of the objective F in (9), we get

$$\begin{aligned} &F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}^*) - F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) \\ &= \frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \underbrace{f(s^h, \boldsymbol{\mathcal{X}}; z^*_{i_1\cdots i_N}) - f(s^h, \boldsymbol{\mathcal{X}}; z_{i_1\cdots i_N})}_{f^{\Delta}_{i_1\cdots i_N}}. \end{aligned} \tag{75}$$

Now, using the definition of ℓ_2–smoothing loss in (12), we obtain an upper bound for the loss difference $f^{\Delta}_{i_1\cdots i_N}$ as follows:

$$\begin{aligned} f^{\Delta}_{i_1\cdots i_N} &= \sum_{(j_1,\cdots,j_N)\in\Omega} s^h_{i_1\cdots i_N, j_1\cdots j_N}(z_{i_1\cdots i_N} - z^*_{i_1\cdots i_N}) \cdot (2x_{j_1\cdots j_N} - z^*_{i_1\cdots i_N} - z_{i_1\cdots i_N}) \\ &= \sum_{(j_1,\cdots,j_N)\in\Omega} 2s^h_{i_1\cdots i_N, j_1\cdots j_N}(z_{i_1\cdots i_N} - z^*_{i_1\cdots i_N})(z^*_{j_1\cdots j_N} - z^*_{i_1\cdots i_N}) \\ &+ \sum_{(j_1,\cdots,j_N)\in\Omega} 2s^h_{i_1\cdots i_N, j_1\cdots j_N}(z_{i_1\cdots i_N} - z^*_{i_1\cdots i_N})\varepsilon_{j_1\cdots j_N} - (z^*_{i_1\cdots i_N} - z_{i_1\cdots i_N})^2 \\ &\leq \underbrace{a_1\sqrt{R_{\max}}|z^*_{i_1\cdots i_N} - z_{i_1\cdots i_N}| \sum_{(j_1,\cdots,j_N)\in\Omega} 2s^h_{i_1\cdots i_N, j_1\cdots j_N} \sum_{n=1}^{N} \big(d(\mathbf{y}_{i_n}, \mathbf{y}_{j_n})\big)^{\alpha}}_{T^{(1)}_{i_1\cdots i_N}} \\ &+ \underbrace{\sum_{(j_1,\cdots,j_N)\in\Omega} 2s^h_{i_1\cdots i_N, j_1\cdots j_N}(z_{i_1\cdots i_N} - z^*_{i_1\cdots i_N})\varepsilon_{j_1\cdots j_N}}_{T^{(2)}_{i_1\cdots i_N}} - (z^*_{i_1\cdots i_N} - z_{i_1\cdots i_N})^2, \end{aligned} \tag{76}$$

where the second equality uses our assumption that $x_{j_1\cdots j_N} = z^*_{j_1\cdots j_N} + \epsilon_{j_1\cdots j_N}$ and the second inequality follows form Assumption B.

By substituting (76) into (75) and using (73), we obtain

$$\begin{aligned}
&\sup_{A(\zeta)} F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}^*) - F(s^h, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{Z}}) \\
&= \sup_{A(\zeta)} \Big(\frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} T^{(2)}_{i_1\cdots i_N} + T^{(1)}_{i_1\cdots i_N} \Big) - \zeta^2. \qquad (77)
\end{aligned}$$

Since $\sum_i a_i b_i \le (\sum_i a_i^2)^{\frac{1}{2}} (\sum_i b_i^2)^{\frac{1}{2}}$, it follows from (76) that

$$\begin{aligned}
&\frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} T^{(1)}_{i_1\cdots i_N} \\
\le\ & a_1 \sqrt{R_{\max}} \Big(\frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} (z_{i_1\cdots i_N} - z^*_{i_1\cdots i_N})^2 \Big)^{1/2} \\
\cdot\ & \Big(\frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \big(\sum_{(j_1,\cdots,j_N)\in\Omega} 2 s^h_{i_1\cdots i_N, j_1\cdots j_N} \sum_{n=1}^{N} \big(d(\mathbf{y}_{i_n}, \mathbf{y}_{j_n}) \big)^{\alpha} \big)^2 \Big)^{1/2}.
\end{aligned}$$

Now, using Assumption B and (73), we obtain

$$\sup_{A(\zeta)} \frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} T^{(1)}_{i_1\cdots i_N} \le 2\zeta a_1 \phi_1 \sqrt{R_{\max}}. \qquad (78)$$

Using the fact that $\sum_i a_i b_i \le (\sum_i a_i^2)^{\frac{1}{2}} (\sum_i b_i^2)^{\frac{1}{2}}$, for $T^{(2)}_{i_1\cdots i_N}$ defined in (76), we have

$$\begin{aligned}
&\frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} T^{(2)}_{i_1\cdots i_N} \\
\le\ & \frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \Big(\sum_{(j_1,\cdots,j_N)\in\Omega} 2 s^h_{i_1\cdots i_N, j_1\cdots j_N} (z_{i_1\cdots i_N} - z^*_{i_1\cdots i_N}) \varepsilon_{j_1\cdots j_N} \Big) \\
\le\ & \Big(\frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} (z_{i_1\cdots i_N} - z^*_{i_1\cdots i_N})^2 \Big)^{1/2} \\
\cdot\ & \Big(\frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \big(\sum_{(j_1,\cdots,j_N)\in\Omega} 2 s^h_{i_1\cdots i_N, j_1\cdots i_N} \varepsilon_{j_1\cdots j_N} \big)^2 \Big)^{1/2}. \qquad (79)
\end{aligned}$$

We combine (79) with (73) to obtain

$$\begin{aligned}
&\sup_{A(\zeta)} \frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} T^{(2)}_{i_1\cdots i_N} \\
&\le \zeta \Big(\frac{1}{\prod_{n\in[N]} I_n} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \big(\sum_{(j_1,\cdots,j_N)\in\Omega} 2 s^h_{i_1\cdots i_N, j_1\cdots i_N} \varepsilon_{j_1\cdots j_N} \big)^2 \Big)^{1/2}. \qquad (80)
\end{aligned}$$

Substituting (78) and (80) into (77) yields

$$\sup_{A(\zeta)}\Big(F(s^h,\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Z}}^*)-F(s^h,\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Z}})\Big)=2\zeta a_1\phi_1\sqrt{R_{\max}}$$
$$+\zeta\Big(\frac{1}{\prod_{n\in[N]}I_n}\sum_{i_1=1}^{I_1}\cdots\sum_{i_N=1}^{I_N}\big(\sum_{(j_1,\cdots,j_N)\in\Omega}2s^h_{i_1\cdots i_N,j_1\cdots i_N}\varepsilon_{j_1\cdots j_N}\big)^2\Big)^{1/2}-\zeta^2. \tag{81}$$

Now, let

$$Q_\zeta:=\frac{\zeta^2-2\zeta a_1\sqrt{R_{\max}}\phi_1-\lambda J(\boldsymbol{\mathcal{Z}}^*)}{\zeta}. \tag{82}$$

It follows from (81) that

$$\begin{aligned}
&P^*\Big(\sup_{A(\zeta)}(F(s^h,\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Z}}^*)-F(s^h,\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Z}}))\geq\lambda(J(\boldsymbol{\mathcal{Z}})-J(\boldsymbol{\mathcal{Z}}^*))\Big)\\
\leq\;& P^*\Big(\sup_{A(\zeta)}(F(s^h,\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Z}}^*)-F(s^h,\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Z}}))\geq -J(\boldsymbol{\mathcal{Z}}^*))\Big)\\
\leq\;& P^*\Big(\frac{1}{\prod_{n\in[N]}I_n}\sum_{i_1=1}^{I_1}\cdots\sum_{i_N=1}^{I_N}\big(\sum_{(j_1,\cdots,j_N)\in\Omega}2s^h_{i_1\cdots i_N,j_1\cdots i_N}\varepsilon_{j_1\cdots j_N}\big)^2)^{1/2}\geq Q_\zeta\Big)\\
\leq\;& P^*\Big(\max_{i_1,\cdots,i_N}|\sum_{(j_1,\cdots,j_N)\in\Omega}2s^h_{i_1\cdots i_N,j_1\cdots i_N}\varepsilon_{j_1\cdots j_N}|\geq Q_\zeta\Big)\\
\leq\;& \sum_{i_1=1}^{I_1}\cdots\sum_{i_N=1}^{I_N}2\exp\big(-\frac{Q_\zeta^2}{2\sigma^2\max_{i_1,\cdots,i_N}\sum_{(j_1,\cdots,j_N)\in\Omega}(s^h_{i_1\cdots i_N,j_1\cdots i_N})^2}\big)\\
\leq\;& 2\prod_{n\in[N]}I_n\exp\big(-\frac{Q_\zeta^2}{2\sigma^2\phi_2}\big),
\end{aligned}\tag{83}$$

where the third to the last inequalities follow from the Chernoff inequality of a weighted sub-Gaussian distribution Chung and Lu (2006) and Assumption C.

Let $\zeta=2^{t-1}\eta$. By our assumptions, we have $\eta\geq 2^4a_1\sqrt{R_{\max}}\phi_1$ and $\lambda J(\boldsymbol{\mathcal{Z}}^*)\leq 2^{-2}\eta^2$. Substituting these bounds into (82) and using (83), we obtain

$$\begin{aligned}
&\sum_{t=1}^{\infty}P^*\Big(\sup_{A(2^{t-1}\eta)}(F(s^h,\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Z}}^*)-F(s^h,\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{Z}}))\geq\lambda J(\boldsymbol{\mathcal{Z}})-\lambda J(\boldsymbol{\mathcal{Z}}^*)\Big)\\
\leq\;& 4\prod_{n\in[N]}I_n\sum_{t=1}^{\infty}\exp(\frac{-2^{2t-8}\eta^2}{2\phi_2\sigma^2})\\
\leq\;& 4\prod_{n\in[N]}I_n\frac{\exp(-a_2\frac{\eta^2}{\phi_2\sigma^2})}{1-\exp(-a_2\frac{\eta^2}{\sigma^2\phi_2})},
\end{aligned}$$

where $a_2>0$ is a constant.

Thus, there exists a positive constant a_3 such that

$$P(\rho(\widehat{\boldsymbol{\mathcal{Z}}},\boldsymbol{\mathcal{Z}}^*)\geq\eta)\;\leq\;a_3\exp(-a_2\frac{\eta^2}{\phi_2\sigma^2}+\log\prod_{n\in[N]}I_n).$$

■

Proof of Corollary 12

Proof Assume Ω is sufficiently large. It follows form the law of large numbers that

$$\sum_{(j_1,\cdots,j_N)\in\Omega} K_h\left(\sum_{n=1}^{N}\|\mathbf{y}_{j_n}-\mathbf{y}_{i_n}\|\right)c_{i_1\cdots i_N,j_1\cdots j_N} \geq \frac{|\Omega|}{2}E\left(K_h\left(\sum_{n=1}^{N}\|\mathbf{y}-\mathbf{y}_{i_n}\|\right)c_{i_1\cdots i_N}\Big|\Delta=1\right). \quad (84)$$

On the other hand, by letting $u=\sum_{n=1}^{N}\|\mathbf{y}-\mathbf{y}_{i_n}\|$, we obtain

$$\begin{aligned} E\left(K_h(u)\,c\Big|\Delta=1\right) &\geq P(c=1,\Delta=1)\,E\left(K_h(u)\Big|c=1,\Delta=1\right)\\ &\geq a_4E\left(K_h(u)\Big|c=1,\Delta=1\right)\\ &\geq a_5\int K_h(u)f_{U\big|c=1,\Delta=1}udu\\ &\geq a_6h\int K_h(u)du\\ &\geq a_7|\Omega|h. \end{aligned} \quad (85)$$

Here, $f_{U\big|c=1,\Delta=1}$ is the conditional density for U defined in (35) and the inequalities follow from Assumption D.

Similarly, for some positive constants a_8, we have

$$\begin{aligned} &\sum_{(j_1,\cdots,j_N)\in\Omega} K_h\left(\sum_{n=1}^{N}\|\mathbf{y}_{j_n}-\mathbf{y}_{i_n}\|\right)\left(\sum_{n=1}^{N}\|\mathbf{y}_{j_n}-\mathbf{y}_{i_n}\|\right)^{\alpha}c_{i_1\cdots i_N,j_1\cdots j_N}\\ \geq\ & 2|\Omega|E\left(K_h(U_{i_1\cdots i_N})U^{\alpha}_{i_1\cdots i_N}\Big|c_{i_1\cdots i_N}=1,\Delta=1)\right)\\ \geq\ & 2|\Omega|\int K_h(u)u^{\alpha}f_{U_{i_1\cdots i_N}|c_{i_1\cdots i_N}=1,\Delta=1}udu\\ \geq\ & a_8|\Omega|h^{\alpha+1}, \end{aligned} \quad (86)$$

where the last inequality uses Assumption D.

Further,

$$\begin{aligned} &\sum_{(j_1,\cdots,j_N)\in\Omega} K_h^2(\|y_{i_1,\cdots,i_N}-y_{j_1,\cdots,j_N}\|)c_{i_1\cdots i_N,j_1\cdots j_N}\\ \geq\ & 2|\Omega|E\left(K_h(U_{i_1\cdots i_N})U_{i_1\cdots i_N}\big|c_{i_1\cdots i_N}=1,\Delta=1)\right)\\ \leq\ & a_9|\Omega|h. \end{aligned} \quad (87)$$

We combine the inequalities (84)–(87) to get

$$\begin{aligned} \max_{i_1\cdots i_N}\sum_{(j_1,\cdots,j_N)\in\Omega}(s^h_{i_1\cdots i_N,j_1\cdots j_N})^2 &\leq (|\Omega|h)^{-1}, \quad \text{and}\\ \max_{i_1\cdots i_N}\sum_{(j_1,\cdots,j_N)\in\Omega}s^h_{i_1\cdots i_N,j_1j_2\cdots j_N}(\sum_{n=1}^{N}d(\mathbf{y}_{i_n},\mathbf{y}_{j_n}))^{\alpha} &\leq h^{\alpha}. \end{aligned}$$

Thus, $\phi_1 = h^\alpha$ and $\phi_2 = (|\Omega|h)^{-1}$, then the desired result immediately follows. ■

References

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

Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.

Boris Alexeev, Michael A Forbes, and Jacob Tsimerman. Tensor rank: Some lower and upper bounds. In *Computational Complexity (CCC), 2011 IEEE 26th Annual Conference on*, pages 283–291. IEEE, 2011.

Elizabeth S Allman, Peter D Jarvis, John A Rhodes, and Jeremy G Sumner. Tensor rank, invariants, inequalities, and applications. *SIAM Journal on Matrix Analysis and Applications*, 34(3):1014–1045, 2013.

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15:2773–2832, 2014.

Mohammad Taha Bahadori, Qi Rose Yu, and Yan Liu. Fast multivariate spatio-temporal analysis via low rank tensor learning. *Advances in neural information processing systems*, pages 3491–3499, 2014.

Mehdi Bahri, Yannis Panagakis, and Stefanos P Zafeiriou. Robust kronecker component analysis. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Arindam Banerjee, Sugato Basu, and Srujana Merugu. Multi-way clustering on relation graphs. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 145–156. SIAM, 2007.

Heinz H Bauschke, Patrick L Combettes, et al. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

Peter N. Belhumeur, João P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

Xuan Bi, Annie Qu, Junhui Wang, and Xiaotong Shen. A group-specific recommender system. *Journal of the American Statistical Association*, 112(519):1344–1353, 2017.

Xuan Bi, Annie Qu, Xiaotong Shen, et al. Multilayer tensor factorization with applications to recommender systems. *The Annals of Statistics*, 46(6B):3308–3333, 2018.

Xuan Bi, Xiwei Tang, Yubai Yuan, Yanqing Zhang, and Annie Qu. Tensors in statistics. *Annual Review of Statistics and Its Application*, 8, 2020.

Patrick Billingsley. *Convergence of probability measures*. John Wiley & Sons, 2013.

Jan Bogaert, Steven Dymarkowski, Andrew M Taylor, and Vivek Muthurangu. *Clinical cardiac MRI*. Springer Science & Business Media, 2012.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization or nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2): 459–494, 2014.

Jason Brownlee. Train-test split for evaluating machine learning algorithms. *Machine learning mastery*, 2020, 2020.

Emmanuel J Candes, Carlos A Sing-Long, and Joshua D Trzasko. Unbiased risk estimates for singular value thresholding and spectral estimators. *IEEE transactions on signal processing*, 61(19):4643–4657, 2013.

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

Yi-Lei Chen and Chiou-Ting Hsu. Multilinear graph embedding: Representation and regularization for images. *IEEE Transactions on Image Processing*, 23(2):741–754, 2014.

Yi-Lei Chen, Chiou-Ting Hsu, and Hong-Yuan Mark Liao. Simultaneous tensor decomposition and completion using factor priors. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):577–591, 2013.

Eric C Chi and Tamara G Kolda. On tensors, sparsity, and nonnegative factorizations. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1272–1299, 2012.

Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: a survey. *Internet Mathematics*, 3(1):79–127, 2006.

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

Ben Dai, Junhui Wang, Xiaotong Shen, and Annie Qu. Smooth neighborhood recommender systems. *Journal of machine learning research*, 20, 2019.

Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

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

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

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

Ehsan Elhamifar, Guillermo Sapiro, and S Shankar Sastry. Dissimilarity-based sparse subset selection. *IEEE transactions on pattern analysis and machine intelligence*, 38(11): 2182–2197, 2015.

Jianqing Fan. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*. Routledge, 2018.

Yixin Fang and Junhui Wang. Selection of the number of clusters via the bootstrap method. *Computational Statistics & Data Analysis*, 56(3):468–477, 2012.

Evgeny Frolov and Ivan Oseledets. Tensor methods and recommender systems. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(3):e1201, 2017.

Hancheng Ge, James Caverlee, and Haokai Lu. Taper: A contextual tensor-based approach for personalized expert recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 261–268. ACM, 2016.

Rungang Han, Rebecca Willett, and Anru Zhang. An optimal statistical and computational framework for generalized tensor estimation. *arXiv preprint arXiv:2002.11255*, 2020.

Johan Håstad. Tensor rank is np-complete. *Journal of Algorithms*, 11(4):644–654, 1990.

Simon Hawe, Matthias Seibert, and Martin Kleinsteuber. Separable dictionary learning. *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 438–445, 2013.

Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):328–340, 2005.

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

David Hong, Tamara G. Kolda, and Jed A. Duersch. Generalized canonical polyadic tensor decomposition. *SIAM Review*, 2019. in press.

Mingyi Hong, Zhi-Quan Luo, and Meisam Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

Victoria Hore, Ana Viñuela, Alfonso Buil, Julian Knight, Mark I McCarthy, Kerrin Small, and Jonathan Marchini. Tensor decomposition for multiple-tissue gene expression experiments. *Nature genetics*, 48(9):1094, 2016.

Zhuolin Jiang, Zhe Lin, and Larry S Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2651–2664, 2013.

Misha E Kilmer, Karen Braman, Ning Hao, and Randy C Hoover. 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):148–172, 2013.

Yejin Kim, Robert El-Kareh, Jimeng Sun, Hwanjo Yu, and Xiaoqian Jiang. Discriminative and distinct phenotyping by constrained tensor factorization. *Scientific reports*, 7(1):1114, 2017.

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

Tamara G Kolda and David Hong. Stochastic gradients for large-scale tensor decomposition. *SIAM Journal on Mathematics of Data Science*, 2(4):1066–1095, 2020.

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

Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th International Conference on Computer Vision*, pages 365–372. IEEE, 2009.

Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier, 1995.

Joonseok Lee, Seungyeon Kim, Guy Lebanon, and Yoram Singer. Local low-rank matrix approximation. In *International conference on machine learning*, pages 82–90, 2013.

Jungwoo Lee, Sejoon Oh, and Lee Sael. Gift: Guided and interpretable factorization for tensors with an application to large-scale multi-platform cancer analysis. *Bioinformatics*, 34(24):4151–4158, 2018.

Arthur Liberzon, Chet Birger, Helga Thorvaldsdóttir, Mahmoud Ghandi, Jill P Mesirov, and Pablo Tamayo. The molecular signatures database hallmark gene set collection. *Cell systems*, 1(6):417–425, 2015.

Tianyi Lin, Shiqian Ma, and Shuzhong Zhang. Iteration complexity analysis of multi-block admm for a family of convex minimization without strong convexity. *Journal of Scientific Computing*, 69(1):52–81, 2016.

Sajan Goud Lingala, Yue Hu, Edward DiBella, and Mathews Jacob. Accelerated dynamic mri exploiting sparsity and low-rank structure: kt slr. *IEEE transactions on medical imaging*, 30(5):1042–1054, 2011.

Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220, 2012.

Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

Eric F Lock and Gen Li. Supervised multiway factorization. *Electronic journal of statistics*, 12(1):1150, 2018.

Eric F Lock, Katherine A Hoadley, James Stephen Marron, and Andrew B Nobel. Joint and individual variation explained (jive) for integrated analysis of multiple data types. *The annals of applied statistics*, 7(1):523, 2013.

Haiping Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, 44(7):1540–1551, 2011.

Ju Man and Bir Bhanu. Individual recognition using gait energy image. *IEEE transactions on pattern analysis and machine intelligence*, 28(2):316–322, 2006.

JS Marron, WJ Padgett, et al. Asymptotically optimal bandwidth selection for kernel density estimators from randomly right-censored samples. *The Annals of Statistics*, 15(4): 1520–1535, 1987.

Atsuhiro Narita, Kohei Hayashi, Ryota Tomioka, and Hisashi Kashima. Tensor factorization using auxiliary information. *Data Mining and Knowledge Discovery*, 25(2):298–324, 2012.

Parvin Nazari, Davoud Ataee Tarzanagh, and George Michailidis. Dadam: A consensus-based distributed adaptive gradient method for online optimization. *arXiv preprint arXiv:1901.09109*, 2019.

Parvin Nazari, Davoud Ataee Tarzanagh, and George Michailidis. Adaptive first-and zeroth-order methods for weakly convex stochastic optimization problems. *arXiv preprint arXiv:2005.09261*, 2020.

Trung V Nguyen, Alexandros Karatzoglou, and Linas Baltrunas. Gaussian process factorization machines for context-aware recommendations. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 63–72. ACM, 2014.

Larsson Omberg, Kyle Ellrott, Yuan Yuan, Cyriac Kandoth, Chris Wong, Michael R Kellen, Stephen H Friend, Josh Stuart, Han Liang, and Adam A Margolin. Enabling transparent and collaborative computational analysis of 12 tumor types within the cancer genome atlas. *Nature genetics*, 45(10):1121, 2013.

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

Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

Na Qi, Yunhui Shi, Xiaoyan Sun, Jingdong Wang, Baocai Yin, and Junbin Gao. Multi-dimensional sparse models. *IEEE transactions on pattern analysis and machine intelligence*, 40(1):163–178, 2018.

Garvesh Raskutti, Ming Yuan, Han Chen, et al. Convex regularization for high-dimensional multiresponse tensor regression. *The Annals of Statistics*, 47(3):1554–1584, 2019.

Steffen Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

Emile Richard and Andrea Montanari. A statistical model for tensor pca. In *Advances in Neural Information Processing Systems*, pages 2897–2905, 2014.

R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798, 2007.

Zahra Shakeri, Waheed U Bajwa, and Anand D Sarwate. Minimax lower bounds on dictionary learning for tensor data. *IEEE Transactions on Information Theory*, 2018.

Xiaotong Shen. On the method of penalization. *Statistica Sinica*, 8(2):337–357, 1998.

Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression (pie) database. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 53–58. IEEE, 2002.

Age K Smilde, Johan A Westerhuis, and Ricard Boque. Multiway multiblock component and covariates regression models. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 14(3):301–331, 2000.

Qingquan Song, Hancheng Ge, James Caverlee, and Xia Hu. Tensor completion algorithms in big data analytics. *arXiv preprint arXiv:1711.10105*, 2017.

Charles J Stone et al. An asymptotically optimal window selection rule for kernel density estimates. *The Annals of Statistics*, 12(4):1285–1297, 1984.

Y-H Taguchi. Identification of candidate drugs using tensor-decomposition-based unsupervised feature extraction in integrated analysis of gene expression between diseases and drugmatrix datasets. *Scientific reports*, 7(1):13733, 2017.

Xiwei Tang, Xuan Bi, and Annie Qu. Individualized multilayer tensor learning with an application in imaging analysis. *Journal of the American Statistical Association*, 115(530): 836–851, 2020.

Dacheng Tao, Xuelong Li, Weiming Hu, Stephen Maybank, and Xindong Wu. Supervised tensor learning. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

Davoud Ataee Tarzanagh and George Michailidis. Estimation of graphical models through structured norm minimization. *Journal of Machine Learning Research*, 18(209):1–48, 2018a.

Davoud Ataee Tarzanagh and George Michailidis. Fast randomized algorithms for t-product based tensor operations and decompositions with applications to imaging data. *SIAM Journal on Imaging Sciences*, 11(4):2629–2664, 2018b.

Robert Tibshirani and Trevor Hastie. Local likelihood estimation. *Journal of the American Statistical Association*, 82(398):559–567, 1987.

Ryota Tomioka and Taiji Suzuki. Convex tensor decomposition via structured schatten norm regularization. In *Advances in neural information processing systems*, pages 1331–1339, 2013.

L. R. Tucker. The extension of factor analysis to three-dimensional matrices. In H. Gulliksen and N. Frederiksen, editors, *Contributions to mathematical psychology.*, pages 110–127. Holt, Rinehart and Winston, New York, 1964.

Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

Katrien Verbert, Nikos Manouselis, Xavier Ochoa, Martin Wolpers, Hendrik Drachsler, Ivana Bosnic, and Erik Duval. Context-aware recommender systems for learning: a survey and future challenges. *IEEE Transactions on Learning Technologies*, 5(4):318–335, 2012.

Philippe Vieu. Nonparametric regression: optimal local bandwidth choice. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(2):453–464, 1991.

Matt P Wand and M Chris Jones. *Kernel smoothing*. Chapman and Hall/CRC, 1994.

Junhui Wang. Consistent selection of the number of clusters via crossvalidation. *Biometrika*, 97(4):893–904, 2010.

Yichen Wang, Robert Chen, Joydeep Ghosh, Joshua C Denny, Abel Kho, You Chen, Bradley A Malin, and Jimeng Sun. Rubik: Knowledge guided tensor factorization and completion for health data analytics. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1265–1274. ACM, 2015.

Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1):29–63, 2019.

Larry Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006.

John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M Stuart, Cancer Genome Atlas Research Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113, 2013.

Wing Hung Wong, Xiaotong Shen, et al. Probability inequalities for likelihood ratios and convergence rates of sieve mles. *The Annals of Statistics*, 23(2):339–362, 1995.

Fei Wu, Xu Tan, Yi Yang, Dacheng Tao, Siliang Tang, and Yueting Zhuang. Supervised nonnegative tensor factorization with maximum-margin constraint. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 962–968, 2013.

Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 521–528, 2003.

Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM international conference on data mining*, pages 211–222. SIAM, 2010.

Yangyang Xu, Ruru Hao, Wotao Yin, and Zhixun Su. Parallel matrix factorization for low-rank tensor completion. *arXiv preprint arXiv:1312.1254*, 2013.

Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE transactions on pattern analysis and machine intelligence*, 29(1):40–51, 2007.

Kenan Y Yılmaz, Ali T Cemgil, and Umut Simsekli. Generalised coupled tensor factorisation. In *Advances in neural information processing systems*, pages 2151–2159, 2011.

Shiqi Yu, Daoliang Tan, and Tieniu Tan. A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 441–444. IEEE, 2006.

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

Yanqing Zhang, Xuan Bi, Niansheng Tang, and Annie Qu. Dynamic tensor recommender systems. *arXiv preprint arXiv:2003.05568*, 2020.

Zemin Zhang and Shuchin Aeron. Exact tensor completion using t-svd. *IEEE Transactions on Signal Processing*, 2016.

Zemin Zhang, Gregory Ely, Shuchin Aeron, Ning Hao, and Misha Kilmer. 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*, pages 3842–3849, 2014.

Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. Bayesian cp factorization of incomplete tensors with automatic rank determination. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1751–1763, 2015.

Hua Zhou, Lexin Li, and Hongtu Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.