

Research Article

Improved Coupled Tensor Factorization with Its Applications in Health Data Analysis

Qing Wu,¹ Jie Wang ,¹ Jin Fan ,¹ Gang Xu ,¹ Jia Wu ,² Blake Johnson,³ Xingfei Li,¹ Quan Do,⁴ and Ruiquan Ge¹

¹School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China

²Department of Computing, Macquarie University, Sydney, Australia

³Department of Cognitive Science, Macquarie University, Sydney, Australia

⁴Advanced Analytics Institute, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, Australia

Correspondence should be addressed to Jin Fan; fanjin@hdu.edu.cn

Received 30 August 2018; Revised 20 December 2018; Accepted 8 January 2019; Published 5 February 2019

Academic Editor: Ludovico Minati

Copyright © 2019 Qing Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Coupled matrix and tensor factorizations have been successfully used in many data fusion scenarios where datasets are assumed to be exactly coupled. However, in the real world, not all the datasets share the same factor matrices, which makes joint analysis of multiple heterogeneous sources challenging. For this reason, approximate coupling or partial coupling is widely used in real-world data fusion, with exact coupling as a special case of these techniques. However, to fully address the challenge of tensor factorization, in this paper, we propose two improved coupled tensor factorization methods: one for approximately coupled datasets and the other for partially coupled datasets. A series of experiments using both simulated data and three real-world datasets demonstrate the improved accuracy of these approaches over existing baselines. In particular, when experiments on MRI data is conducted, the performance of our method is improved even by 12.47% in terms of accuracy compared with traditional methods.

1. Introduction

With the rapid development of cyber physical systems, a soaring amount of data from heterogeneous sources is now easily accessible. Analysing data from multiple sources has been proven to enhance knowledge discovery by capturing its underlying structures, which are otherwise difficult to extract. For instance, in recommendation systems, it is not only possible to rely on past user ratings as additional assistance for joint analysis, but to also consider the supply chain surrounding a product, or the similarity between users and other information [1–3]. Drawing upon additional related information can improve recommendation performance. In metabolomics—an analytical technique used to study biological fluids such as LC-MS (liquid chromatography-mass spectrometry) and NMR (nuclear magnetic resonance)—joint analysis helps to accurately identify the various component chemicals [4, 5]. Electroencephalography (EEG) and functional magnetic resonance imaging (fMRI) are

complementary patterns and when jointly analyzed, can provide “the best of both worlds,” i.e., EEG’s superior time resolution and fMRI’s superior spatial resolution. Hence, fusing models can, for example, provide deeper insights into the activities of the brain or help improve medical treatments for nervous system diseases [6–8].

A common and effective way to deal with multisource data is to represent them as matrices and then use *collective matrix factorization* (CMF) [9] for joint analysis. Matrix-based joint analysis is used extensively in many fields, including bioinformatics [10, 11], social network analysis [12, 13], signal processing [14, 15], and so on. However, this type of analysis only works with two-dimensional data and cannot be applied to datasets of three or more dimensions. However, recent developments in sensor technology now allow more and different aspects of data to be captured, and higher order tensors have become an important tool for representing these multidimensional datasets. Accordingly, tensor decomposition was introduced to accurately extract

the correlations between different dimensions and extensions to joint decomposition for heterogeneous and highly dimensional datasets have naturally followed.

Applying coupled higher-order tensors and matrices to heterogeneous datasets from multiple sources has been a topic of interest in many areas, such as metabolomics [16], blind source separation [17], recommendation systems [18, 19], link prediction [20], and brain imaging [21, 22]. The various problems to be solved with this technique are called *coupled matrix and tensor factorization* (CMTF) problems. Acar et al. proposed an all-at-once optimization approach, called CMTF-OPT [23], which is based on gradients. The advanced version of CMTF-OPT, ACMTF-OPT [16], places additional constraints on the CMTF model to force good behavior when distinguishing between shared and unshared data components. Many researchers have subsequently made improvements to CMTF to allow for joint analysis on large-scale data [24, 25], increase the speed of calculation on large-scale data, and provide for situations with data sparsity. As a result, through the joint decomposition of high-order tensors and matrices, CMTF can extract shared and hidden patterns from most heterogeneous datasets and construct those patterns into a factor matrix. However, multisource datasets hold unique forms of shared relationships, including approximately shared or partially shared data. Models that are solely designed for an exactly shared factor matrix may not be suitable. For example, traffic flow data from upstream and downstream highways are clearly related to each other but may not be exactly coupled. There are many other examples, such as MRI images from the same patient or continuous tensor data streams that hold their own internal relationships. For these types of real-life data fusion tasks, joint analysis is crucial [26, 27].

In this paper, we focus on joint data analysis with datasets that are partially or approximately coupled [28]. We propose two improved coupled tensor factorization methods: one for partially coupled datasets, called CTF-PSF, and one for approximately coupled datasets, called CTF-AC.

A summary of our contributions follows.

- (i) The proposed CTF-AC method is the very first tensor factorization model to address data fusion with approximately coupled datasets. This model is also suitable for multisource datasets that are not coupled but where the data are highly correlated.
- (ii) By combining individual decomposition and coupled decomposition, a new coupled tensor factorization method called CTF-PSF emerges. This method handles data fusion with partially coupled datasets.
- (iii) Extensive experiments on synthetic and real-world datasets verify that the two proposed methods generate more accurate results than the traditional methods.

The rest of this paper is organized as follows. Section 2 introduces some background knowledge on tensor decomposition and provides the problem definition. The details of how our approaches work are introduced in Section 3. Section 4 describes the experimental design of this paper

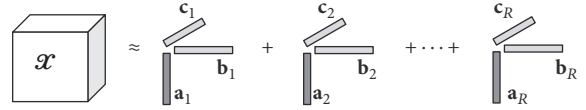


FIGURE 1: A CP decomposition of a third-order tensor. Figure adapted from [29].

and numerical experiments to illustrate the advantages of the proposed methods. Finally, we conclude our work and discuss future research directions in Section 5.

2. Preliminaries and Problem Definition

Following the notations in [29], vectors (tensors of order one) are denoted in boldface lowercase letters, e.g., $\mathbf{a}, \mathbf{b}, \mathbf{c}$. Matrices (tensors of order two) appear as boldface capital letters, e.g., $\mathbf{A}, \mathbf{B}, \mathbf{C}$. The n th column of \mathbf{A} is denoted as \mathbf{a}_n . $\mathbf{A}^{(n)}$ indicates the n th matrix in a sequence. For example, $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}$ represent a sequence of N matrices. The transpose of matrix \mathbf{A} is denoted by \mathbf{A}^T . Higher-order tensors (third-order or higher) appear as boldface Euler script letters, e.g., \mathcal{X}, \mathcal{W} . $\mathbf{X}_{(n)}$ indicates the mode- n matricization of an N^{th} -order tensor \mathcal{X} , which can be obtained by permuting the dimensions of \mathcal{X} and reshaping the permuted tensor into a matrix. $\|\mathbf{a}\|$ and $\|\mathbf{A}\|$ denote the two-norm of \mathbf{a} and the Frobenius norm of \mathbf{A} , respectively. The Hadamard products is indicated by $*$. Table 1 lists all the symbols used in this paper.

2.1. CANDECOMP/PARAFAC Decomposition. CANDECOMP/PARAFAC (CP) is one of the most popular tensor decompositions. The goal of CP decomposition is to factorize a tensor into a sum of rank-one tensors. For instance, given a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, after CP decomposition, \mathcal{X} can be approximately represented as

$$\mathcal{X} \approx \sum_{r=1}^R \mathcal{X}_r \equiv \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \quad (1)$$

where \mathcal{X}_r is a rank-one tensor and the symbol “ \circ ” represents the vector outer product operator [29]. R is a positive integer, which means it approximates \mathcal{X} with R rank-one tensors. This CP model can be concisely described by

$$\mathcal{X} \approx \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \equiv \llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket, \quad (2)$$

where $\llbracket \cdot \rrbracket$ denotes the CP decomposition operator [30]. In this CP decomposition, \mathbf{A}, \mathbf{B} , and \mathbf{C} are the factor matrices of \mathcal{X} , which represent a combination of the vectors from the rank-one components in Figure 1; i.e., $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \dots \ \mathbf{a}_R]$. Later, Acar et al. improved CP algorithm and developed an algorithm named CP-WOPT [31]; it uses a first-order optimization method to solve the weighted least squares problem.

TABLE 1: List of symbols used in this paper.

Symbol	Definition
$\mathcal{X}, \mathbf{X}, \mathbf{x}$	tensor, matrix, column vector
$\mathbf{X}_{(n)}$	mode- n matricization of a tensor
\mathbf{a}_n	the n th column vector of \mathbf{A}
$\mathbf{A}^{(n)}$	the n th matrix in the sequence $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}$
\circ	Outer product
\bigcirc	Outer product of multiple matrices
$*$	Hadamard product
\mathbb{R}	the set of real numbers
\mathbf{A}^T	transpose of \mathbf{A}
$\ \cdot\ $	for matrices and vectors, it refers to the analogous Frobenius and two-norm, respectively
\mathcal{W}	missing index tensor
$size(\mathbf{A})$	number of entries in matrix \mathbf{A}

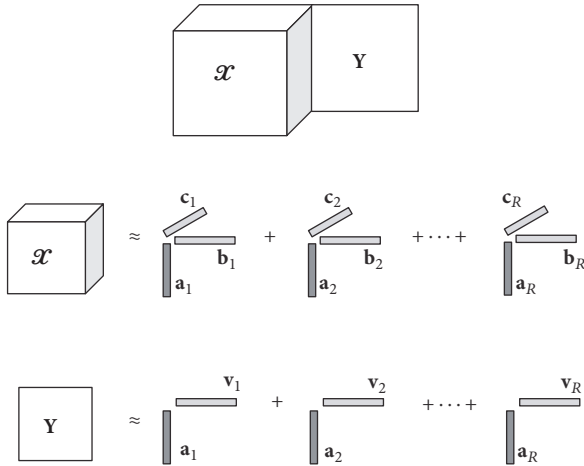


FIGURE 2: An illustration of a joint decomposition with a third-order tensor and a matrix factorization of R components. \mathcal{X} and \mathbf{Y} share one dimension; that is, $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_R]$ as the common factor matrix. Figure adapted from [32].

2.2. Coupled Tensor Factorization. Coupled factorization methods have become an effective means for jointly analyzing multisource datasets. The simplest form of coupled tensor factorization is *collective matrix factorization* (CMF). For example, in a movie recommendation system, additional information about the movie, such as the movie genre, its actors, or the user's social network, in addition to the user's historical ratings, could be used to improve the accuracy of rating predictions. For example, a user rating matrix for the movie can be expressed as matrix \mathbf{X} , which represents *movie* \times *user*, coupled with matrix \mathbf{Y} , which represents *movie* \times *type*. This CMF model can be defined as

$$f(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \|\mathbf{X} - \mathbf{UV}^T\|^2 + \|\mathbf{Y} - \mathbf{UW}^T\|^2, \quad (3)$$

where \mathbf{U} , \mathbf{V} , and \mathbf{W} are the factor matrices.

As shown in Figure 2, a high-order extension of CMF, i.e., a CMTF model, can be simply defined as

$$f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{V}) = \|\mathcal{X} - \llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket\|^2 + \|\mathbf{Y} - \mathbf{AV}^T\|^2, \quad (4)$$

where \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{V} are the factor matrices. In this problem, CMTF-OPT is used to vectorize all the factor matrices and their partial derivatives so the problem can be solved by any gradient-based optimization algorithm, such as the nonlinear conjugate gradient (NCG) method. More details can be found in [23].

2.3. Problem Definition. Consider a coupled tensor factorization of two third-order tensors \mathcal{X}_1 and \mathcal{X}_2 that are coupled in the first dimension. \mathbf{A} , \mathbf{B} , and \mathbf{C} are the factor matrices of \mathcal{X}_1 and \mathbf{U} , \mathbf{V} , and \mathbf{W} are the factor matrices of \mathcal{X}_2 . To jointly factorize \mathcal{X}_1 and \mathcal{X}_2 , the objective function can be written as

$$f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{U}, \mathbf{V}, \mathbf{W}) = \frac{1}{2} \|\mathcal{X}_1 - \llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket\|^2 + \frac{1}{2} \|\mathcal{X}_2 - \llbracket \mathbf{U}, \mathbf{V}, \mathbf{W} \rrbracket\|^2. \quad (5)$$

Given our focus is on situations that are not exactly coupled, but rather approximately coupled, e.g., $\mathbf{A} \approx \mathbf{U}$, function (5) is no longer applicable. However, like soft constraints, the matrices can be coupled approximately by adding a regularization term. Then, the objective function becomes

$$f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{U}, \mathbf{V}, \mathbf{W}) = \frac{1}{2} \|\mathcal{X}_1 - \llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket\|^2 + \frac{1}{2} \|\mathcal{X}_2 - \llbracket \mathbf{U}, \mathbf{V}, \mathbf{W} \rrbracket\|^2 + \frac{\lambda}{2} \|\mathbf{A} - \mathbf{U}\|^2. \quad (6)$$

However, function (6) has two potential issues. (i) The model loses accuracy when there is a large difference between the number of entries in tensors \mathcal{X}_1 and \mathcal{X}_2 . The errors from approximating \mathcal{X}_1 and \mathcal{X}_2 will have a different impact on

the objective function depending on whether there are many more or many less entries in \mathcal{X}_1 than \mathcal{X}_2 . Therefore, using the same weight ratio will, obviously, result in a loss of accuracy [33]. (ii) Further, this model is only suitable for cases of two-tensor coupling and cannot be applied to multiple tensor scenarios.

A completely shared factor matrix, whether approximate or not, is only one type of exact coupling, e.g., $\mathbf{A} = \mathbf{U}$. There are other types of exact coupling scenarios, such as partial coupling, e.g., $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_3]$, $\mathbf{U} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{u}_3]$, where heterogeneous datasets only share some, but not all, components [34]. The methods based on function (4) may not be applicable to such situations. Hence, we turn our attention to partial coupling with an extension to CTF-AC, called CTF-PSF. CTF-PSF is based on Acar et al.'s [23] CMTF-OPT algorithm, but with some modifications to allow for data reconstruction with heterogeneous data that has both shared and unshared components. More details on this model appear in Section 3.2.

3. The Proposed Models

In real life, many heterogeneous datasets are only approximately coupled, which means that their dimensions are not exactly coupled. The CTF-AC model offers a joint decomposition solution to situations with approximately coupled datasets. Moreover, it is relatively common for multisource datasets to be partially coupled, which means that only some of the factors in a potential matrix are shared, not all, as is the case with exact coupling and its variants. To address these situations, we have extended CTF-AC to incorporate the CMTF-OPT algorithm in a method called CTF-PSF to offer joint decomposition for partially coupled datasets. The CTF-AC model is presented in Section 3.1. The CTF-PSF model is presented in Section 3.2.

3.1. CTF-AC. To address the two potential problems associated with function (7), i.e. unbalanced tensor entries and its inapplicability to multitensor scenarios, we have developed a ‘‘two birds with one stone’’ solution. To overcome potential inaccuracies as a result of unbalanced tensor entry distributions, we have added error weights to the objective function (Section 3.1.1) and to extend traditional models for use with more than two tensors, we have added a soft constraint to the transfer factor matrix (Section 3.1.2).

3.1.1. Adding Error Weights. When a weight is assigned to the fitting error for each tensor, function (6) becomes

$$\begin{aligned} f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{U}, \mathbf{V}, \mathbf{W}) &= \frac{\omega_1}{2} \|\mathcal{X}_1 - \llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket\|^2 \\ &+ \frac{\omega_2}{2} \|\mathcal{X}_2 - \llbracket \mathbf{U}, \mathbf{V}, \mathbf{W} \rrbracket\|^2 \quad (7) \\ &+ \frac{\lambda}{2} \|\mathbf{A} - \mathbf{U}\|^2, \end{aligned}$$

where $1/2$ helps with the derivative calculations and ω_1, ω_2 are the error weights from approximating \mathcal{X}_1 and \mathcal{X}_2 , respectively. λ is the error weight of the Frobenius norm of

\mathbf{A} and \mathbf{U} . To equalize the contribution of errors in each part of the objective function, ω_1 and ω_2 are set to the reciprocals of the number of entries in \mathcal{X}_1 and \mathcal{X}_2 , respectively. λ is set to the reciprocal of number of entries in \mathbf{A} .

$$\begin{aligned} f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{U}, \mathbf{V}, \mathbf{W}) &= \frac{\|\mathcal{X}_1 - \llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket\|^2}{2 \|\mathcal{W}^{(1)}\|^2} \\ &+ \frac{\|\mathcal{X}_2 - \llbracket \mathbf{U}, \mathbf{V}, \mathbf{W} \rrbracket\|^2}{2 \|\mathcal{W}^{(2)}\|^2} \quad (8) \\ &+ \frac{\|\mathbf{A} - \mathbf{U}\|^2}{2 \text{size}(\mathbf{A})}, \end{aligned}$$

where $\mathcal{W}^{(1)}$ and $\mathcal{W}^{(2)}$ are binary tensors of the same size as \mathcal{X}_1 and \mathcal{X}_2 , respectively. Therefore, $\|\mathcal{W}^{(1)}\|^2$ and $\|\mathcal{W}^{(2)}\|^2$ indicate the number of entries of \mathcal{X}_1 and \mathcal{X}_2 , and $\text{size}(\mathbf{A})$ denotes the number of entries in \mathbf{A} . In this way, the model eliminates the influence an imbalanced number of tensor entries has on accuracy.

3.1.2. Adding a Soft Constraint to the Transfer Factor Matrix. To extend traditional models for use with more than two tensors, we have modified function (8) on the assumption that the Frobenius norm may have possible transitivity. Assume that tensor \mathcal{X}_3 is approximately shared with \mathcal{X}_1 and \mathcal{X}_2 . \mathbf{D}, \mathbf{E} , and \mathbf{F} are the factor matrices of \mathcal{X}_3 . Three tensors can then be approximately coupled by

$$\begin{aligned} f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{U}, \mathbf{V}, \mathbf{W}, \mathbf{D}, \mathbf{E}, \mathbf{F}) &= \frac{\|\mathcal{X}_1 - \llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket\|^2}{2 \|\mathcal{W}^{(1)}\|^2} + \frac{\|\mathcal{X}_2 - \llbracket \mathbf{U}, \mathbf{V}, \mathbf{W} \rrbracket\|^2}{2 \|\mathcal{W}^{(2)}\|^2} \\ &+ \frac{\|\mathcal{X}_3 - \llbracket \mathbf{D}, \mathbf{E}, \mathbf{F} \rrbracket\|^2}{2 \|\mathcal{W}^{(3)}\|^2} \quad (9) \\ &+ \frac{\|\mathbf{A} - \mathbf{U}\|^2 + \|\mathbf{U} - \mathbf{D}\|^2 + \|\mathbf{D} - \mathbf{A}\|^2}{2 \text{size}(\mathbf{A})}. \end{aligned}$$

Now consider a more general situation. Suppose there are M tensors $\mathcal{X}^{(m)} \in \mathbb{R}^{I_1^{(m)} \times I_2^{(m)} \times \dots \times I_{N_m}^{(m)}}$ from M sources. The objective function of the joint decomposition of these tensors, based on the CP model, is defined as

$$f = \sum_{m=1}^M \left\| \mathcal{X}^{(m)} - \bigcirc_{n=1}^{N_m} \mathbf{A}_m^{(n)} \right\|^2, \quad (10)$$

where

$$\bigcirc_{n=1}^{N_m} \mathbf{A}_m^{(n)} = \mathbf{A}_m^{(1)} \circ \mathbf{A}_m^{(2)} \circ \dots \circ \mathbf{A}_m^{(N_m)}. \quad (11)$$

Further, assume that there are C related factors in these M relevant tensors; i.e.,

$$I_c^{(1)} = I_c^{(2)} = \dots = I_c^{(M)}. \quad (12)$$

Thus, functions (9) and (10) can be modified as

$$\begin{aligned}
& f(\mathbf{A}_1^{(1)}, \dots, \mathbf{A}_1^{(N)}, \mathbf{A}_2^{(1)}, \dots, \mathbf{A}_2^{(N)}, \dots, \mathbf{A}_M^{(1)}, \dots, \mathbf{A}_M^{(N)}) \\
&= \underbrace{\frac{\|\mathcal{X}^{(1)} - \bigcirc_{n=1}^{N_1} \mathbf{A}_1^{(n)}\|^2}{\|\mathcal{W}^{(1)}\|^2} + \frac{\|\mathcal{X}^{(2)} - \bigcirc_{n=1}^{N_2} \mathbf{A}_2^{(n)}\|^2}{\|\mathcal{W}^{(2)}\|^2} + \dots + \frac{\|\mathcal{X}^{(M)} - \bigcirc_{n=1}^{N_M} \mathbf{A}_M^{(n)}\|^2}{\|\mathcal{W}^{(M)}\|^2}}_{\text{tensors}} \\
&+ \underbrace{\frac{\sum_{m=1}^{M-1} \|\mathbf{A}_m^{(1)} - \mathbf{A}_{m+1}^{(1)}\|^2}{\text{size}(\mathbf{A}_1^{(1)})} + \frac{\|\mathbf{A}_M^{(1)} - \mathbf{A}_1^{(1)}\|^2}{\text{size}(\mathbf{A}_1^{(1)})}}_{1^{\text{st}} \text{ shared factor constraint}} + \underbrace{\frac{\sum_{m=1}^{M-1} \|\mathbf{A}_m^{(2)} - \mathbf{A}_{m+1}^{(2)}\|^2}{\text{size}(\mathbf{A}_1^{(2)})} + \frac{\|\mathbf{A}_M^{(2)} - \mathbf{A}_1^{(2)}\|^2}{\text{size}(\mathbf{A}_1^{(2)})}}_{2^{\text{st}} \text{ shared factor constraint}} + \dots \\
&+ \underbrace{\frac{\sum_{m=1}^{M-1} \|\mathbf{A}_m^{(C)} - \mathbf{A}_{m+1}^{(C)}\|^2}{\text{size}(\mathbf{A}_1^{(C)})} + \frac{\|\mathbf{A}_M^{(C)} - \mathbf{A}_1^{(C)}\|^2}{\text{size}(\mathbf{A}_1^{(C)})}}_{C^{\text{st}} \text{ shared factor constraint}} = \sum_{m=1}^M \frac{\|\mathcal{X}^{(m)} - \bigcirc_{n=1}^{N_m} \mathbf{A}_m^{(n)}\|^2}{\|\mathcal{W}^{(m)}\|^2} + \sum_{c=1}^C \frac{\sum_{m=1}^{M-1} \|\mathbf{A}_m^{(c)} - \mathbf{A}_{m+1}^{(c)}\|^2}{\text{size}(\mathbf{A}_1^{(c)})} \\
&+ \sum_{c=1}^C \frac{\|\mathbf{A}_M^{(c)} - \mathbf{A}_1^{(c)}\|^2}{\text{size}(\mathbf{A}_1^{(c)})}.
\end{aligned} \tag{13}$$

Let $\partial \|\mathcal{X}^{(m)} - \bigcirc_{n=1}^{N_m} \mathbf{A}_m^{(n)}\|^2 / \partial \mathbf{A}_m^{(n)} = g(\mathbf{A}_m^{(n)})$. Then, the partial derivatives of f with respect to $\mathbf{A}_m^{(n)}$ can be calculated with

function (14) as follows:

$$\frac{\partial f}{\partial \mathbf{A}_m^{(n)}} = \begin{cases} \left\{ \begin{array}{l} \frac{2 * g(\mathbf{A}_m^{(n)})}{\|\mathcal{W}^{(m)}\|^2} + \frac{2 * (2 * \mathbf{A}_m^{(n)} - \mathbf{A}_{m-1}^{(n)} - \mathbf{A}_{m+1}^{(n)})}{\text{size}(\mathbf{A}_1^{(n)}), m \neq 1, m \neq M \\ \frac{2 * g(\mathbf{A}_m^{(n)})}{\|\mathcal{W}^{(m)}\|^2} + \frac{2 * (2 * \mathbf{A}_m^{(n)} - \mathbf{A}_M^{(n)} - \mathbf{A}_{m+1}^{(n)})}{\text{size}(\mathbf{A}_1^{(n)}), m = 1 \\ \frac{2 * g(\mathbf{A}_m^{(n)})}{\|\mathcal{W}^{(m)}\|^2} + \frac{2 * (2 * \mathbf{A}_m^{(n)} - \mathbf{A}_{m-1}^{(n)} - \mathbf{A}_1^{(n)})}{\text{size}(\mathbf{A}_1^{(n)}), m = M} \end{array} \right\} & n \leq C \\ \\ \frac{2 * g(\mathbf{A}_m^{(n)})}{\|\mathcal{W}^{(m)}\|^2}, & C < n \leq N. \end{cases} \tag{14}$$

The factor matrix between multiple tensors is constrained using the soft constraint transfer method. When taking partial derivatives of the shared factor matrix, the solution of the factor matrix for the first tensor and the last tensor is somewhat different from that in the middle. Hence, the shared factor matrix is divided into these three different types of partial derivatives, i.e., the first, last, and middle.

With all the gradients of the factor matrix derived, the problem can be solved with any gradient-based method. The algorithm flow of the joint-filled nonlinear conjugate gradient with C related factors of these M relevant tensors is shown in Algorithm 1. Convergence is achieved when the relative change of the objective function is less than the set threshold. The algorithm terminates when the number of iterations reaches its maximum.

3.2. CTF-PSF. The CTF-AC model outlined above can further be evolved into a new model that deals with partially coupled datasets, i.e., CTF-PSF [35].

As shown in Figure 3, when heterogeneous datasets only share some components rather than all, methods based on objective function (4) may not be applicable. Without loss of generality, we take the coupled datasets of a tensor and a matrix as an example. Figure 3 shows that a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ and a matrix $\mathbf{Y} \in \mathbb{R}^{I \times M}$ are coupled in the first dimension. However, suppose they have the same low-rank structures, i.e., the same number of R . Let $\mathbf{A}_1 \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$ be the factor matrices of \mathcal{X} extracted through an individual decomposition with R components. Similarly, $\mathbf{A}_2 \in \mathbb{R}^{I \times R}$ and $\mathbf{V} \in \mathbb{R}^{M \times R}$ are the factor matrices extracted from matrix \mathbf{Y} using a matrix factorization with


```

Input:  $\mathcal{X}^{(m)} \in \mathbb{R}^{I_1^{(m)} \times I_2^{(m)} \times \dots \times I_{N_m}^{(m)}}$ ,  $m \in [1, M]$ ,  $R, C$ 
Output: factor matrices  $\mathbf{A}_m^{(n)}$ ,  $m \in [1, M]$ ,  $n \in [1, N_m]$ 
1 iter = 0;
2 termination = 0;
3 convergence = 0;
4 initialize  $\mathbf{A}_m^{(n)}$ ,  $m \in [1, M]$ ,  $n \in [1, N_m]$ ;
5 while convergence  $\neq$  1 or termination  $\neq$  1 do
6    $\mathbf{x} \leftarrow \mathbf{A}_m^{(n)}$ , vectorized factor matrices;
7    $f \leftarrow$  calculate objective function using (13);
8    $\partial f / \partial \mathbf{A}_m^{(n)} \leftarrow$  calculate the partial derivatives of  $f$  with respect to  $\mathbf{A}_m^{(n)}$  using (14);
9    $\mathbf{g} \leftarrow$  vectorized combination of  $\partial f / \partial \mathbf{A}_m^{(n)}$ ;
10   $\mathbf{x} \leftarrow \mathbf{x}, f, \mathbf{g}$ , update with nonlinear conjugate gradient and linear search;
11   $\mathbf{A}_m^{(n)} \leftarrow \mathbf{x}$ , convert the vector to the factor matrices;
12  convergence  $\leftarrow$  convergence analysis;
13  termination  $\leftarrow$  termination analysis;
14  iter  $\leftarrow$  iter + 1;
15 end

```

ALGORITHM 1: The pseudocode of CTF-AC.

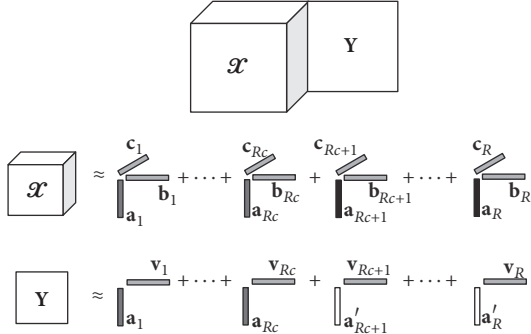


FIGURE 3: Illustration of a joint decomposition for a third-order \mathcal{X} and a \mathbf{Y} coupled in one dimension. They have R_c shared components and $R - R_c$ unshared components rather than sharing all the components.

R components. In partially coupled multisource datasets, the factor matrix derived from each source will not match exactly; i.e., \mathbf{A}_1 and \mathbf{A}_2 are likely to only share some columns. Further suppose that tensor \mathcal{X} and matrix \mathbf{Y} have R_c shared components and R_s ($R - R_c$) unshared components. Then, the objective function (4) can be modified to

$$f(\mathbf{A}_1, \mathbf{B}, \mathbf{C}, \mathbf{A}_2, \mathbf{V}) = \frac{1}{2} \|\mathcal{X} - \llbracket \mathbf{A}_1, \mathbf{B}, \mathbf{C} \rrbracket\|^2 + \frac{1}{2} \|\mathbf{Y} - \mathbf{A}_2 \mathbf{V}^T\|^2, \quad (15)$$

where

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{R_c}, \mathbf{a}_{R_c+1}, \dots, \mathbf{a}_R \\ \mathbf{A}_c \quad \mathbf{A}_{s1} \end{bmatrix}, \quad (16)$$

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{R_c}, \mathbf{a}'_{R_c+1}, \dots, \mathbf{a}'_R \\ \mathbf{A}_c \quad \mathbf{A}_{s2} \end{bmatrix},$$

and where $\mathbf{A}_{s1} \in \mathbb{R}^{I \times R_s}$ and $\mathbf{A}_{s2} \in \mathbb{R}^{I \times R_s}$ are the unshared columns of \mathbf{A}_1 and \mathbf{A}_2 , respectively, and $\mathbf{A}_c \in \mathbb{R}^{I \times R_c}$ are the shared R_c columns. Thus, the objective function (15) can be further modified to

$$f(\mathbf{A}_c, \mathbf{A}_{s1}, \mathbf{A}_{s2}, \mathbf{B}, \mathbf{C}, \mathbf{V}) = \frac{1}{2} \underbrace{\|\mathcal{X} - \llbracket \mathbf{A}_c, \mathbf{A}_{s1} \rrbracket, \mathbf{B}, \mathbf{C} \rrbracket\|^2}_{f_1} + \frac{1}{2} \underbrace{\|\mathbf{Y} - [\mathbf{A}_c, \mathbf{A}_{s2}] \mathbf{V}^T\|^2}_{f_2}. \quad (17)$$

Here, the shared and unshared components are optimized separately. The unshared components of the tensors and matrix are updated using individual decompositions, and the shared components of the tensor and the matrix are updated using joint decompositions. Specific details of this optimization can be found in [35].

The pseudocode for CTF-PSF is shown in Algorithm 2. The algorithm terminates when the number of iterations and the number of function evaluations reach their respective maximums. The algorithm converges based on the relative change value and the two-norm of the gradient of all factor matrices divided by the number of entries in the gradient. The functions of *InD* and *JoD* in Algorithm 2 denote individual and joint decompositions, respectively. First, each single dataset is decomposed individually to update the unshared columns ($\mathbf{A}_{s1}, \mathbf{A}_{s2}$) in the matrix of shared dimension (line 7 to 17 in Algorithm 2). The others factor matrices (\mathbf{B}, \mathbf{C} , and \mathbf{V}) and the shared columns (\mathbf{A}_c) in the matrix of the shared dimension are updated through joint decomposition (line 19 in Algorithm 2). However, this does mean that the number of shared components needs to be determined in advance. To ensure proper modeling (line 20 Algorithm 2), there is a necessary adjustment step to combine $\mathbf{A}_c, \mathbf{A}_{s1}$, and \mathbf{A}_{s2} into

```

Input:  $\mathcal{X}, \mathbf{Y}, R_c, \widehat{R}$ 
Output:  $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{C}, \mathbf{V}$ 
1 initialize  $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{C}, \mathbf{V}$  where  $\mathbf{A}_1 = \mathbf{A}_2$ ;
2  $iter = 0$ ;
3  $termination = 0$ ;
4  $convergence = 0$ ;
5 while  $convergence \neq 1$  or  $termination \neq 1$  do
6   individual decompositions;
7   if  $R_c < \widehat{R}$  then
8     if  $iter = 0$  then
9       update by initialization;
10       $\mathbf{A}_{s1} \leftarrow InD(\mathcal{X}, \mathbf{A}_1, \mathbf{B}, \mathbf{C}, \widehat{R})$ ;
11       $\mathbf{A}_{s2} \leftarrow InD(\mathbf{Y}, \mathbf{A}_2, \mathbf{V}, \widehat{R})$ ;
12    else
13      update by the result of last iteration;
14       $\mathbf{A}_{s1} \leftarrow InD(\mathcal{X}, \mathbf{A}_1, \mathbf{B}, \mathbf{C}, \widehat{R})$ ;
15       $\mathbf{A}_{s2} \leftarrow InD(\mathbf{Y}, \mathbf{A}_2, \mathbf{V}, \widehat{R})$ ;
16    end
17  end
18  joint decomposition;
19   $\mathbf{A}_c, \mathbf{B}, \mathbf{C}, \mathbf{V} \leftarrow JoD(\mathcal{X}, \mathbf{Y}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{C}, \mathbf{V}, \widehat{R})$ ;
20   $\mathbf{A}_1, \mathbf{A}_2 \leftarrow Adjust(\mathbf{A}_c, \mathbf{A}_{s1}, \mathbf{A}_{s2})$ ;
21   $convergence \leftarrow$  convergence analysis;
22   $termination \leftarrow$  termination analysis;
23   $iter \leftarrow iter + 1$ ;
24 end

```

ALGORITHM 2: The pseudocode of CTF-PSF.

$\mathbf{A}_1, \mathbf{A}_2$. It is worth noting here that CMTF-OPT is a special case of CTF-PSF when $R_c = \widehat{R}$.

4. Experiments

4.1. Experimental Design. We tested and verified the advantages of both CTF-AC and CTF-PSF through a series of comparative experiments with several baselines on both simulated data and three real-world datasets.

4.1.1. Baselines. The baselines roughly fall into two different categories: methods that jointly decompose multiple tensors, such as CMTF-OPT and ACMTF-OPT, and methods that individually decompose a single tensor, such as CP-WOPT. A description of each follows.

- (i) CMTF-OPT first vectorizes all the factor matrices and their partial derivatives so that problems can be solved using any gradient-based optimization algorithm.
- (ii) ACMTF-OPT is an advanced version of CMTF-OPT that includes additional constraints to allow analysis of more complex coupled data. It can also be used for missing completion. More details can be found in [16].
- (iii) CP-WOPT is an individual decomposition method for single tensors, it uses a first-order optimization method to solve the weighted least squares problem.

4.1.2. Performance Metrics. The performance of all methods, including CTF-AC and CTF-PSF, was evaluated according to the accuracy of missing completions. Hence, missing index tensors in the models were added to deal with incomplete data. The assessment metric is defined as the difference between the original and the estimated entries for missing values, known as a tensor completion score (TCS). TCS is defined as

$$TCS = \frac{\|(1 - \mathcal{W}) * (\mathcal{X} - \widehat{\mathcal{X}})\|}{\|(1 - \mathcal{W}) * \mathcal{X}\|}, \quad (18)$$

where \mathcal{X} is the initial tensor and $\widehat{\mathcal{X}}$ indicates the datasets estimated by different methods. \mathcal{W} is a binary tensor of the same size as \mathcal{X} and represents the missing entries in \mathcal{X} with zeros to represent the missing data and ones to represent valid data. Obviously, the smaller the TCS value, the better the result.

In the CTF-AC experiments, we also used RMSE to measure the fitness of the observable values for each method, defined as

$$RMSE = \frac{\|\mathcal{W} * (\mathcal{X} - \widehat{\mathcal{X}})\|}{\|\mathcal{W}\|}. \quad (19)$$

4.1.3. Real-World Datasets. *Dataset1* (*Dataset1* is available at <http://www.models.life.ku.dk/3Dnosedata>) contains data from an electronic nose sensor. It comprises structural data from readings on the smell of licorice given 18 licorice samples \times 241 times \times 12 sensors [36], including 6 good licorice samples, 6 bad licorice samples, and 6 fabricated bad licorice samples. These were mixed into three tensors, each with a dimension of $6 \times 241 \times 12$ to test approximately coupled datasets with CTF-AC.

Dataset2 (*Dataset2* is available at <http://www.med-info.cs.ucy.ac.cy>) consists of 38 patients who underwent two MRI scans of different parts of the brain within the same year. One patient was randomly selected from 38 patients, and then we randomly selected two MRI brain scans of the same site as the two sources of data. Each scan has a size of 378×378 . The two MRI images are shown in Figure 4. *Dataset2* was also used to assess CTF-AC.

Dataset3 (*Dataset3* is available at <http://www.models.life.ku.dk/joda/prototype>) contains 29 chemical mixtures, each comprising five chemicals measured using LC-MS (liquid chromatography-mass spectrometry) and NMR (nuclear magnetic resonance). NMR was able to detect all five component chemicals and the results can be formulated as tensors $\mathcal{X} \in \mathbb{R}^{28 \times 13324 \times 8}$. LC-MS, however, only detected four components and therefore the results were formulated as a matrix $\mathbf{Y} \in \mathbb{R}^{28 \times 168}$. More details about these coupled datasets can be found in [16]. This dataset was used to assess CTF-PSF.

4.1.4. General Experimental Parameters. For all comparative experiments, each method was given the same termination conditions. The maximum number of iterations was set to (10^4) , and the maximum number of function evaluations was set to (10^5) . Additionally, the relative change in loss function

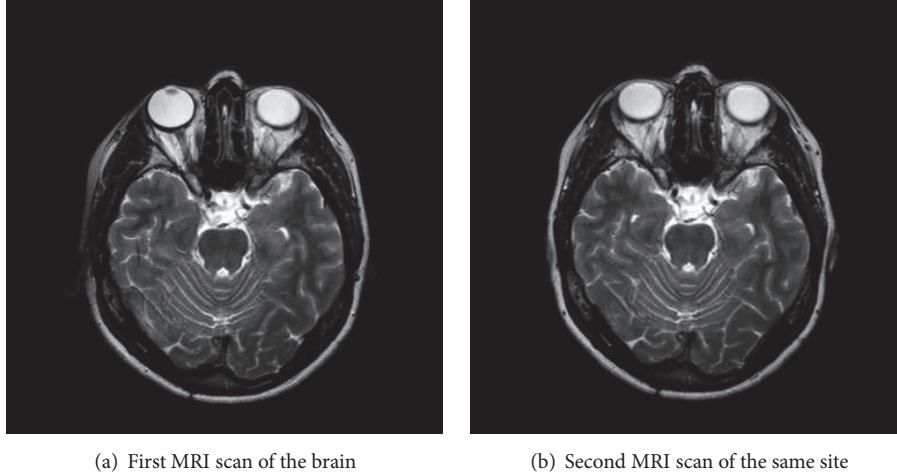


FIGURE 4: The results of two brain examinations of the same patient on the same site within the same year.

values was set to 10^{-6} and the two-norm of the gradient divided by the number of entries in the gradient was set to 10^{-7} . The sparsity penalty parameters for ACMTF-OPT were set to 10^{-3} ; i.e., $\beta = 10^{-3}$.

4.2. Partially Coupled Data

4.2.1. Simulated Data

Experimental Set-Up. Tensor data $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ and matrix data $\mathbf{Y} \in \mathbb{R}^{I_n \times M}$ were generated according to the same technique in [32] using the following formulation:

$$\mathcal{X} = \left[\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)} \right] + \left[\mathbf{B}^{(1)}, \dots, \mathbf{B}^{(N)} \right], \quad (20)$$

$$\mathbf{Y} = \mathbf{A}^{(n)} \mathbf{V}^T + \mathbf{C}^{(1)} \mathbf{C}^{(2)T},$$

where $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_c}$ is the factor matrix shared by both datasets and the other matrices $\mathbf{A}^{(i)} \in \mathbb{R}^{I_i \times R_c}$ ($i = 1, 2, \dots, N$ and $i \neq n$) are the factor matrices for other dimensions of the tensor. $\mathbf{V} \in \mathbb{R}^{I_n \times R_c}$ denotes the factor matrix corresponding to the second dimension of matrix \mathbf{Y} . R_c denotes the number of shared components. The factor matrices $\mathbf{B}^{(i)} \in \mathbb{R}^{I_n \times R_s}$, $i \in [1, N]$, are the unshared factors for each dimension of the tensor. $\mathbf{C}^{(1)} \in \mathbb{R}^{I_n \times R_s}$ and $\mathbf{C}^{(2)} \in \mathbb{R}^{M \times R_s}$ are the unshared factors of the matrix \mathbf{Y} , and R_s represents the number of the unshared components in each dataset. $\mathbf{B}^{(n)}$, $\mathbf{C}^{(1)}$, and $\mathbf{A}^{(n)}$ correspond to \mathbf{A}_{s1} , \mathbf{A}_{s2} , and \mathbf{A}_c in Section 3.2, respectively.

All matrices, except for $\mathbf{B}^{(n)}$ and $\mathbf{C}^{(1)}$, were generated randomly with entries drawn from a standard normal distribution. All matrix columns were normalized to a unit norm. Then, Gaussian noise was added to the tensor and matrix using $\mathcal{X}_n = \mathcal{X} + \eta \mathcal{N}(\|\mathcal{X}\|/\|\mathcal{N}\|)$, $\mathbf{Y}_n = \mathbf{Y} + \eta \mathbf{N}(\|\mathbf{Y}\|/\|\mathbf{N}\|)$, respectively, where η indicates that the noise levels, tensors \mathcal{N} , and matrix \mathbf{N} are the same size. All entries had a standard normal distribution. Finally, the simulated missing values were added to tensor \mathcal{X}_n according to a sampling ratio, denoted as SR.

In the first dimension, we used a tensor size of $50 \times 30 \times 20$ coupled with a matrix of 50×100 . The factor matrices were generated and constructed as coupled datasets using (20). Let \hat{R} indicate the estimation rank of the coupled datasets. In individual decompositions, \hat{R} denotes the estimation rank of the individual datasets. In the experiments with CTF-PSF, the total number of shared and unshared components were set to four unless otherwise specified; i.e., the rank of each individual dataset was set to 4 ($R = R_c + R_s = 4$). The noise level η and the sampling ratio of missing values (SR) were set independently for each comparative test. As previously mentioned, performance was evaluated according to the estimation accuracy of missing values as calculated by (18).

Numerical Results. Figures 5 and 6 shows the TCSs of all four methods for different SR at $R_c = 3$, $R_s = 1$, and $\eta = 0.1$ when $\hat{R} = 3$ and $\hat{R} = 4$, respectively. Figure 5(a) shows similar performance by the three joint decompositions based methods when $R_c = \hat{R}$. In Figure 5(b), we see that CP-WOPT, which is based on individual decomposition was relatively stable and ostensibly equivalent to the joint decomposition methods with a missing value ratio of less than 90%. However, Figure 5(c) shows that the joint decomposition methods performed well with many missing values, while individual decomposition produced completely inaccurate results. CTF-PSF considers the shared components and does not take the unshared components into account when $\hat{R} = R_c$. In other words, CMTF-OPT is a special case of CTF-PSF when the coupled datasets do not have any unshared components. Figure 6(a) shows that CMTF-OPT and ACMTF-OPT gave almost the same performance when $R_c \neq \hat{R}$. However, in contrast to the previous experiment, individual decomposition had certain advantages with a missing value ratio below 90%, as shown in Figure 6(b). It is worth noting that since CTF-PSF considers both the unshared components and the shared components, this method performed almost as well as the methods based on individual decomposition.

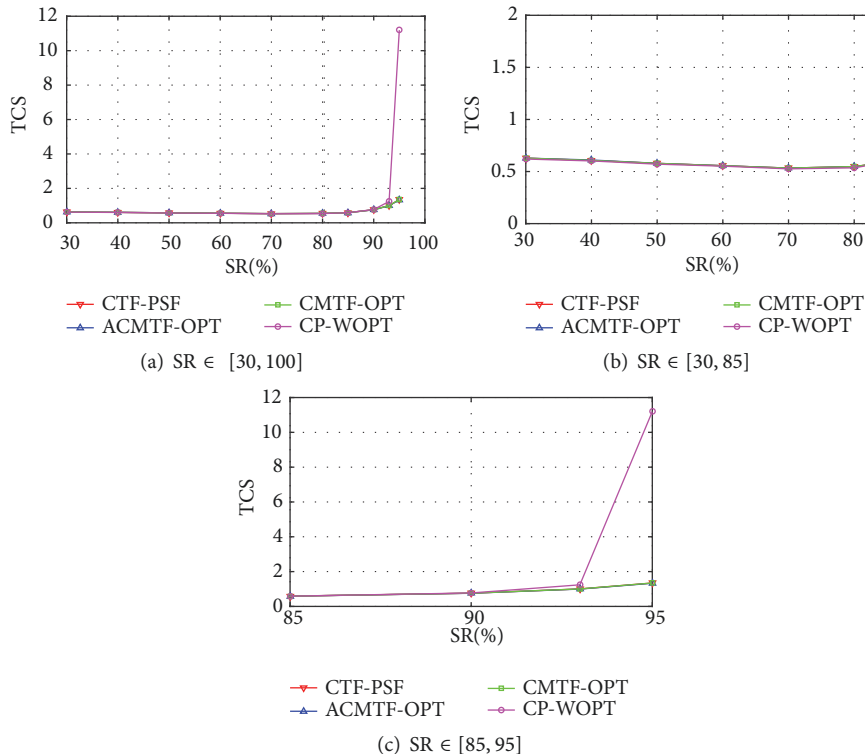


FIGURE 5: The TCS for each baseline with different amounts of missing data ($\hat{R} = 3$ when $R_c = 3, R_s = 1$). The performance of joint decompositions methods was almost the same because CMTF-OPT is a special case of CTF-PSF when $\hat{R} = R_c$. Figures 5(b) and 5(c) are local amplifications of Figure 5(a). The sampling ratio of missing values is represented by SR.

However, as can be seen in Figure 6(c), once the proportion of missing values reached 90%, the TCS for the individual decomposition method rapidly increased, while CTF-PSF continued to provide good performance.

The influence of different numbers of shared components with the methods based on joint decomposition is shown in Figure 7. CTF-PSF and CMTF-OPT were tested with $\hat{R} = R = 6$, $R_c = \{1, 3, 5\}$, and $\eta = 0.1$ at different missing value ratios. As shown, increasing the number of shared components helped to improve completion accuracy. This is mainly because the factor matrix provides more auxiliary information as the number of shared components increases. In addition, the advantages of CTF-PSF became more obvious as the number of shared components increased compared to CMTF-OPT.

Figure 8 shows the TCSs for the joint decomposition methods when \mathcal{X} and \mathcal{Y} were simultaneously sampled at $R_c = 3, \eta = 0.1$ and $\text{SR}(\%) = [90, 40]$. Here, $\hat{R} = R = 4$. S-1 and S-2 represent \mathcal{X} and \mathcal{Y} , respectively. The results show that our method still achieved good completion accuracy when every tensor and matrix contained at least some missing values.

4.2.2. Real-World Data. Recall that \mathcal{X} and \mathcal{Y} in *Dataset3* (described in Section 4.1.3) are partially coupled in terms of the constituent chemicals. These datasets have four shared components and \mathcal{X} has an unshared component. LC-MS data are often noisy and contain many irrelevant features.

TABLE 2: The TCSs for methods based on joint decomposition with the LC-MS and NMR dataset at $\hat{R} = 5, \text{SR}(\%) = \{30, 50, 80\}$ at $\eta = 0.1$.

SR(%)	CTF-PSF	ACMTF-OPT	CMTF-OPT	CTF-AC
[30 0]	0.2669	0.3363	0.2550	4.0155
[50 0]	0.4517	0.5184	0.4470	2.0650
[80 0]	1.2403	1.2336	1.2424	3.5604

Therefore, noise can also be regarded as an unshared component of \mathcal{Y} .

To compare the performance of different methods, \mathcal{X} was simulated with different proportions of missing values, and TCSs were evaluated for all baselines using joint decomposition, as shown in Table 2. CMTF-OPT preformed better than CTF-PSF with lower amounts of missing values (SR(%)), and ACMTF-OPT was superior to CTF-PSF when the missing value ration reached 80%. The reason for this is that CTF-PSF does not consider the weight of shared and unshared components. Unsurprisingly, CTF-AC did not perform as well as the other methods, including CTF-PSF, when faced with partially coupled datasets.

4.2.3. Discussion. Figure 9 shows the TCSs for the methods based on joint decomposition at $\eta = \{0.1, 0.3\}$, $\text{SR}(\%) = \{20, 40, 60, 80\}$, $R_c = \{1, 2, 3, 4\}$, and $\hat{R} = R_c + R_s$. As indicated

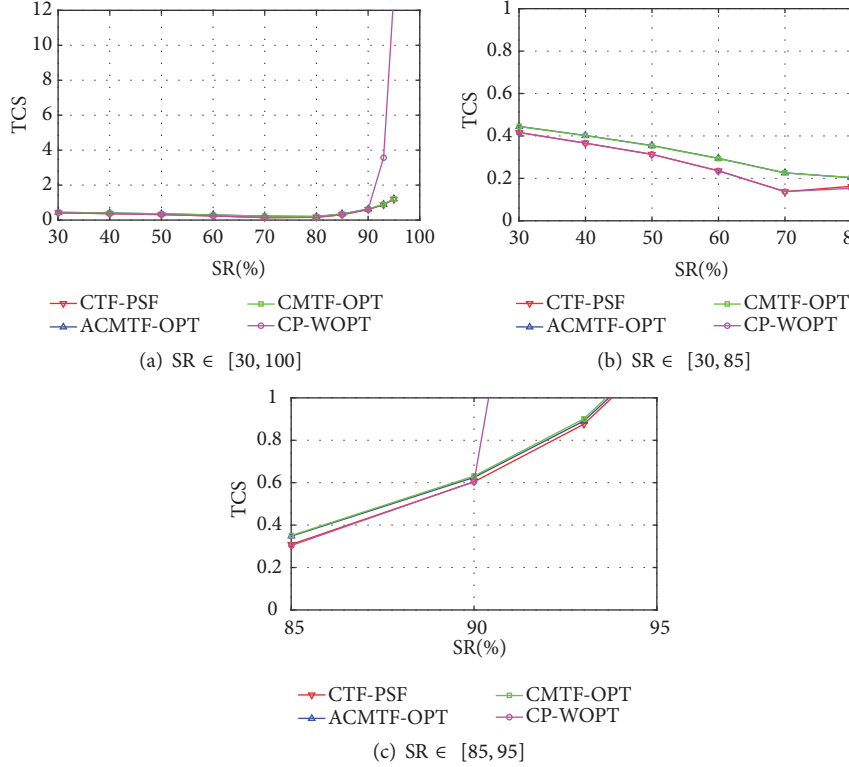


FIGURE 6: The TCS for each baseline with different amounts of missing data ($\hat{R} = 4$ when $R_c = 3, R_s = 1$). CTF-PSF performed better than the baseline joint decomposition methods, indicating that CTF-PSF does not lose accuracy with high amounts of missing data like CP-WOPT. Figures 6(b) and 6(c) are local amplifications of Figure 6(a).

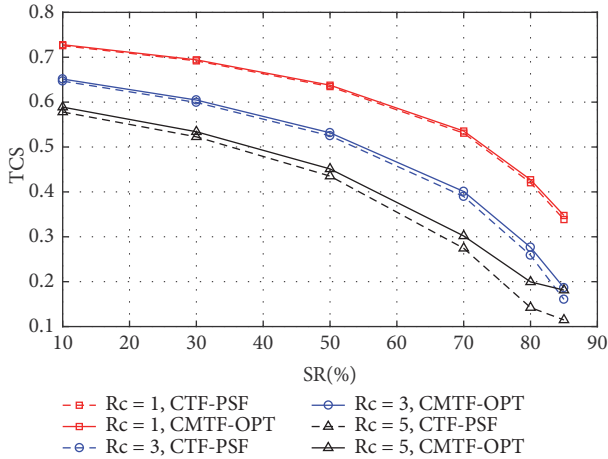


FIGURE 7: The TCS for CTF-PSF at different missing value ratios (SR% where $R_c = \{1, 3, 5\}$, $\hat{R} = R = 6$, $\eta = 0.1$). As the number of shared components increases, the factor matrix provides more auxiliary information, which helps to improve completion accuracy.

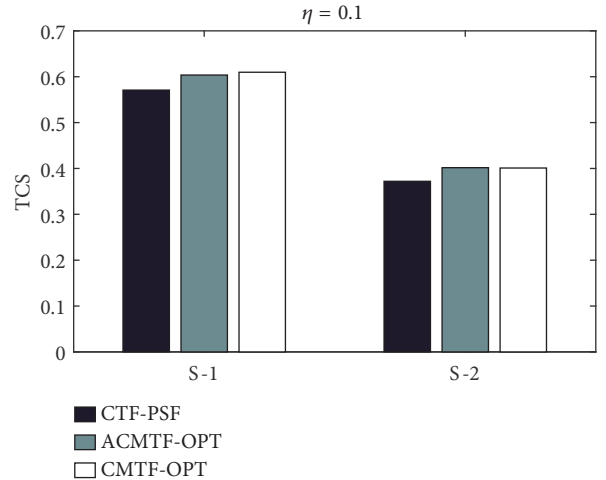


FIGURE 8: An illustration of the joint decomposition methods with a third-order tensor and a matrix with R components. \mathcal{X} and \mathbf{Y} share one dimension (i.e., $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_R]$) as the common factor matrix. Figure adapted from [32].

in the figure, the accuracy of these methods deteriorated as noise increased, particularly with higher proportions of missing values. CTF-PSF performed better than the other methods when $R_c < \hat{R}$, but not obviously so with high levels

of noise ($\eta = 0.3$). This is because unshared components can be very helpful with data reconstruction when $R_c < \hat{R}$. The accuracy of these methods improved as the number of shared components (R_c) increased. And their performance

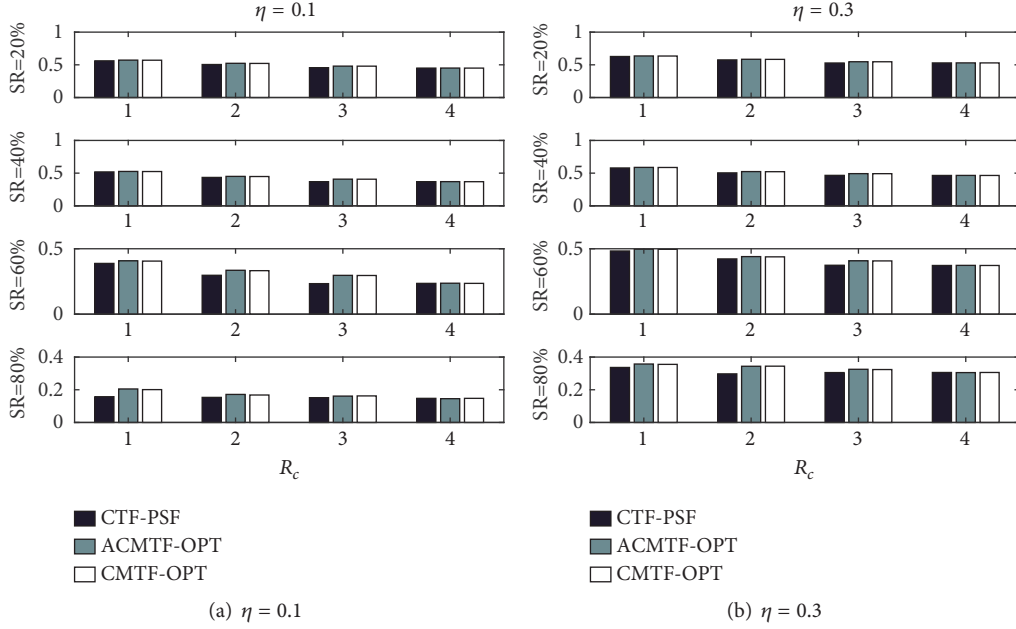


FIGURE 9: The TCSs for coupled factorization methods at $\eta = 0.1$ and $\eta = 0.3$ and $\hat{R} = 4$. The TCSs for these methods deteriorated as noise increased. CTF-PSF was superior to CMTF-OPT and ACMTF-OPT, although their performance gradually diminished as noise increased.

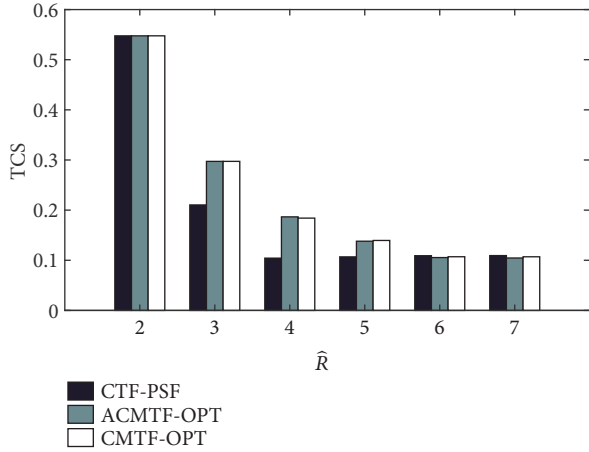


FIGURE 10: The TCSs for coupled factorization methods with different numbers of estimated components (\hat{R}) (at SR = 80%, $\eta = 0.1$ and $R_c = 2, R_s = 2$). CTF-PSF showed its best performance at $\hat{R} = 4$, while CMTF-OPT and ACMTF-OPT peaked at $\hat{R} = 6$.

was almost the same when $R_c = R = 4$, i.e., when all components are shared, where CMTF-OPT becomes a special case of CTF-PSF.

Figure 10 shows the TCSs for the joint decomposition-based methods for different numbers of estimated components. \hat{R} (i.e., $\hat{R} = \{2, 3, 4, 5, 6, 7\}$, at SR(%) = 80, $\eta = 0.1$, and $R_c = 2, R_s = 2$). The TCSs for CMTF-OPT and ACMTF-OPT improved as the number of estimated components increased, peaking at $\hat{R} = 6$. However, despite CTF-PSF's performance improvement until $\hat{R} = 4$, there were no significant changes when $\hat{R} > 4$. These results demonstrate that CTF-PSF is able

to improve accuracy at relatively low estimation ranks over CMTF-OPT and ACMTF-OPT.

4.3. Approximately Coupled Data

4.3.1. Simulated Data

Experimental Set-Up. The multisource datasets we generated contained two types of shared data to simulate the different kinds of shared relationships found in reality. Without loss of generality, two third-order tensors are used as an example to explain the way the data was generated. Suppose the tensors $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ and $\mathcal{Y} \in \mathbb{R}^{I \times M \times N}$ are two related tensors. The factor matrices for tensor \mathcal{X} derived through individual decomposition are \mathbf{A} , \mathbf{B} , and \mathbf{C} , and the factor matrices for tensor \mathcal{Y} derived through individual decomposition are \mathbf{U} , \mathbf{V} , and \mathbf{W} . From these matrices, two simulated datasets were generated in two different ways:

Case 1. The unshared factor matrices \mathbf{B} , \mathbf{C} , \mathbf{V} , and \mathbf{W} and the shared factor matrix \mathbf{A} were randomly generated from the normal distribution. $\mathbf{U} = \text{random}(0, 1) * \mathbf{A} + \text{random}(0, 1)$, where the *random* function represents the random arrays generated from a specified distribution. Two third-order tensors were then formed based on the generated factor matrix and normalized to arrive at $\overline{\mathcal{X}}$ and $\overline{\mathcal{Y}}$. Then, Gaussian noise was added to the tensor, i.e., $\mathcal{X}_n = \overline{\mathcal{X}} + \eta \mathcal{N}(\|\overline{\mathcal{X}}\|_F / \|\mathcal{N}\|_F)$, where $\mathcal{N} \in \mathbb{R}^{I \times J \times K}$ corresponds to the random noise tensor and η is used to adjust the noise level. The same process applies to \mathcal{Y}_n .

Case 2. The only difference between this case and Case 1 is that $\mathbf{U} = \text{normal}(0.5 * \mathbf{n}, 1)$, where \mathbf{n} indicates the parameter

TABLE 3: TCSs and RMSEs for three methods on two datasets at SR(%)= [0, 95].

SR(%)	CTF-AC			MTF			CP-WOPT
	RMSE1	RMSE2	TCS2	RMSE1	RMSE2	TCS2	TCS2
Case 1	2.8155e-4	2.5189e-4	0.1155	2.8108e-4	4.3111e-4	0.2455	0.2841
Case 2	2.8270e-4	3.7283e-4	0.2137	2.8111e-4	3.7471e-4	0.2257	0.3040

related to the number of tensors and the *normal* function produces random arrays from a normal distribution.

The difference between these two datasets is the relationship between the factor matrices \mathbf{A} and \mathbf{U} . In Case 1, \mathbf{A} has a linear relationship to \mathbf{U} . In Case 2, the elements in \mathbf{A} and \mathbf{U} have normal distributions that satisfy the same variance with different mean values. Both these datasets approximate real-world data.

We use CMTF-OPT to realize the traditional multi-source tensor decomposition, i.e., MTF. In the experiments with simulated datasets, we compared CTF-AC with a joint decomposition method (MTF) and an individual decomposition method (CP-WOPT). We set the dimensions of the tensor to 50 and the estimation rank to 5. The estimated rank for joint decomposition methods was set to $\hat{R} = 10$. \hat{R} is the estimated rank for CP-WOPT. All noise levels were set to $\eta = 0.1$ and, again, the sampling ratio of missing values is denoted as SR.

Numerical Results. Table 3 shows the completion degree of the observed values and the completion accuracy of the missing values for both simulated datasets with a missing value ratio of SR(%) = [0, 95]. A missing value ratio of SR(%) = [0, 95] means that the first tensor has no missing values, while 95% of the values in the second tensor are missing. Since the two tensors are the same size, but the amount of missing values is very different, the completions for the first tensor and the second differ greatly, i.e., 100 : 5. Table 3 shows the root mean square errors for the observable data in the first tensor (RMSE1) and the second tensor (RMSE2). TCS1 represents the missing value results for the first tensor, with TCS2 representing the second tensor. The results in Table 3 show that RMSE1 for MTF was similar to CTF-AC. In Case 2, CTF-AC’s RMSE1 (2.8270e-4) was actually larger than MTF (2.8111e-4). However, regardless of the type of dataset, the RMSE2 for MTF was larger than that for CTF-AC, especially in Case 1. This indicates that traditional models, like MTF, sacrifice completion accuracy with tensors that have a smaller number of observables to compensate for completion accuracy with tensors that have a larger number of observables. Therefore, the CTF-AC tensor completion score for the second tensor (TCS2) is better than the MTF under this circumstance. CTF-PSF was not included in the experiments with these simulated datasets given that both are approximately coupled. However, CTF-PSF is included in the following experiments with real-world datasets.

4.3.2. Real-World Datasets. Table 4 shows the TCSs for the four methods with the electronic nose dataset and missing value sampling ratios of SR (%) = [5, 50, 90]. Here, MTF had

TABLE 4: The TCSs for all four methods using the electronic nose dataset sampled at different proportions of missing values (SR(%) = [5, 50, 90]). OVERALL denotes the sum of all TCSs at different proportions.

METHODS	TCS1	TCS2	TCS3	OVERALL
CTF-AC	0.0049	0.0070	0.0858	0.0977
CTF-PSF	0.0040	0.0107	0.1699	0.1846
MTF	0.0029	0.0053	0.1635	0.1717
CP-WOPT	0.0037	0.0034	0.1284	0.1355

better completion accuracy with small missing value ratios (TCS=0.0029 at SR(%)=5), but this accuracy was significantly reduced on tensors with large ratios (TCS=0.1635 at SR(%)=90). CTF-AC shows better TCSs with more missing values because, again, traditional methods, like MTF, sacrifice accuracy with fewer observables in favor of better accuracy with more. With a missing value ratio of 90%, CTF-AC’s TCS was half that of MTF. Adding the TCSs for all three missing value ratios, CTF-AC achieved better overall accuracy. CTF-PSF was less effective than other methods because it is specifically designed to address partial coupling and is not well-suited to approximately coupled datasets.

Table 5 lists the TCSs for the four methods on the brain MRI dataset with missing value sampling ratios of SR(%) = [x, 0] and x = [10, 20, 30, 40, 50, 60, 70]. The bold results denote the best scores. Here, CTF-AC shows a significant advantage over MTF. Although there was no significant difference between the completion accuracy for CTF-AC and CP-WOPT with small missing value ratios, CTF-AC’s completion accuracy on the first image improved in comparison as the ratio increased because it borrows auxiliary information from the second image. Hence, CTF-AC’s TCS was superior to CP-WOPT with a high missing value ratio. Figure 11 shows the completion accuracy of the first MRI image for all methods when SR (%) = [60, 0].

4.3.3. Discussion. Figures 12(a) and 12(b) show the completion accuracy comparison plots for each method on the two simulated multi-source datasets; one with balanced missing value ratios (SR(%) = [50, 50]), the other with imbalanced ratios (SR(%) = [5, 95]). With a balanced ratio, the number of observable and missing values in the tensors is roughly equivalent, while with imbalanced ratios, there is great disparity.

From Figure 12, we see that the TCSs for the weighted model, CTF-AC, and the traditional model, MTF, are similar with balanced observable and missing values. However, as the disparity increases, the disadvantages of traditional models becomes very obvious. Figure 12(a) shows an even higher

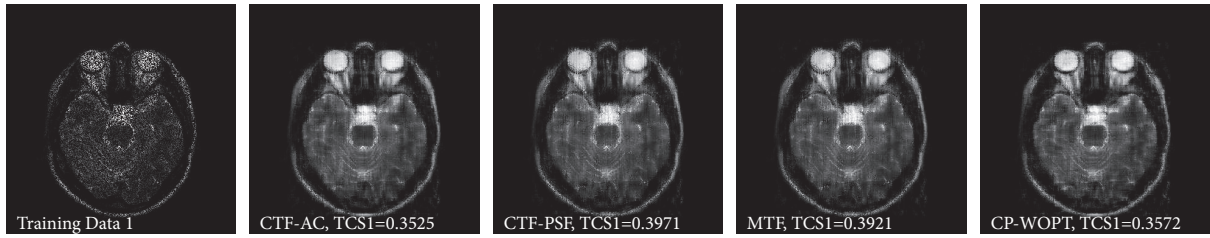
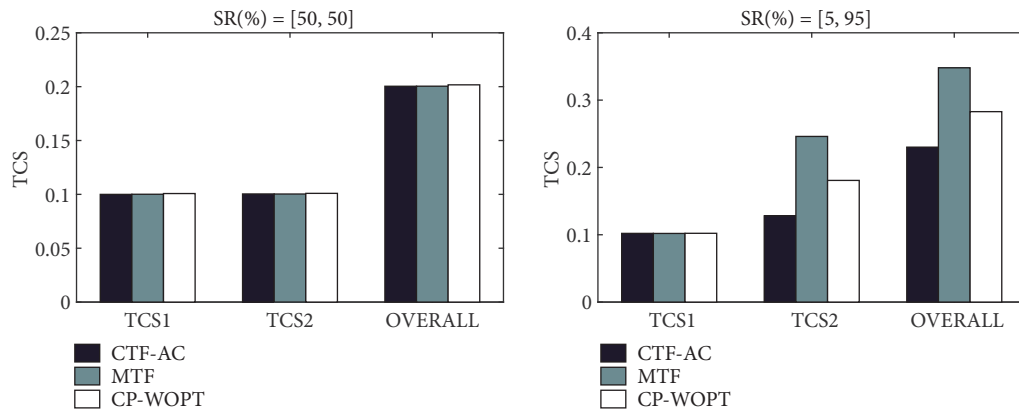
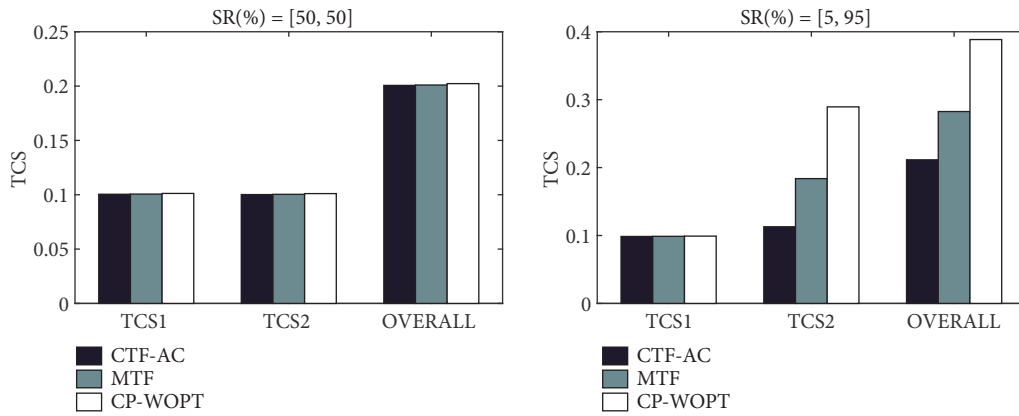


FIGURE 11: MRI data of the brain at $SR(\%) = [60, 0]$, the completion effect of all methods on the first image.



(a) Case 1



(b) Case 2

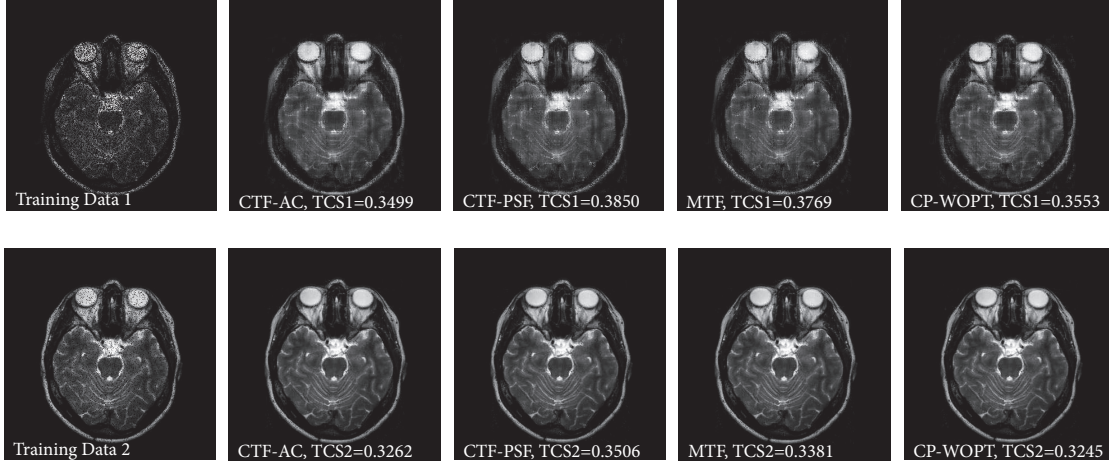
FIGURE 12: The TCSs for the three methods on Cases 1 and 2 with balanced and unbalanced missing values.

TABLE 5: The TCSs for all four methods on MRI brain scans with different ratios of missing values.

SR(%)	CTF-AC	CTF-PSF	MTF	CP-WOPT
[10 0]	0.3427	0.3729	0.3631	0.3408
[20 0]	0.3325	0.3642	0.3596	0.3345
[30 0]	0.3379	0.3779	0.3694	0.3390
[40 0]	0.3413	0.3808	0.3712	0.3436
[50 0]	0.3446	0.3874	0.3797	0.3464
[60 0]	0.3525	0.3971	0.3921	0.3572
[70 0]	0.3641	0.4132	0.4095	0.3783

TABLE 6: The TCSs for all methods on the brain MRI dataset at $SR(\%) = [x, 10]$ and $x = [10, 30, 50, 70]$.

SR(%)	CTF-AC		CTF-PSF		MTF		CP-WOPT	
	TCS1	TCS2	TCS1	TCS2	TCS1	TCS2	TCS1	TCS2
[10 10]	0.3341	0.3229	0.3690	0.3580	0.3595	0.3452	0.3341	0.3228
[30 10]	0.3363	0.3196	0.3723	0.3557	0.3629	0.3387	0.3375	0.3197
[50 10]	0.3499	0.3262	0.3850	0.3506	0.3769	0.3381	0.3553	0.3245
[70 10]	0.3608	0.3205	0.4091	0.3395	0.4087	0.3334	0.3765	0.3267

FIGURE 13: MRI data of the brain at $SR(\%) = [50, 10]$, the completion effect of all methods on the first image and second image.

TCS for CP-WOPT, while the completion accuracy for CTF-AC with tensors that contain a great many missing values remains high. To further investigate the impact of missing values with each method, we sampled the MRI dataset with missing value ratios of $SR(\%) = [x, 10]$ and $x = [10, 30, 50, 70]$ and list the results in Table 6. With a balanced ratio ($SR(\%) = [10, 10]$), CTF-AC still performed better than the other methods, including CTF-PSF. Therefore, using a model that is completely shared or partially shared based on factor matrices may introduce some errors. Overall, these experimental results prove the validity and accuracy of the CTF-AC model.

Figure 13 shows the completion accuracy of CTF-AC, CTF-PSF, MTF, and CP-WOPT on the two images with an imbalanced ratio of $SR(\%) = [50, 10]$. Unlike traditional joint decomposition methods (such as MTF), CTF-AC adds error weights to the objective function and sets a discriminant factor for both datasets. The discriminant factor reflects the correlations in the data to fit each tensor. Therefore, CTF-AC provides better completion accuracy than MTF with greater levels of missing values.

5. Conclusions

Jointly analyzing data from multiple sources has the potential to extract underlying data structures to enhance knowledge discovery. However, in data fusion, traditional coupled tensor factorization has been unable to deal with the diverse relationships found between multi-source datasets, such as approximate or partial couplings. Existing techniques are only appropriate for modeling exact couplings. Therefore, to

address this challenge, we propose two improved coupled tensor factorization methods: one for approximately coupled datasets, CTF-AC, and the other for partially coupled datasets, CTF-PSF. CTF-AC is also suitable for multisource datasets with no dimension couplings if the data is highly correlated. CTF-PSF is an extension of CTF-AC, based on the CMTF-OPT algorithm, which factorizes datasets with both shared and unshared components by combining individual and coupled decompositions. Through numerical experiments, we demonstrate that the tensor completion accuracy of the proposed methods outperforms traditional coupled tensor factorization methods on datasets with approximate and partial couplings. However, there are some disadvantages to the proposed methods. These are highlighted because they provide opportunities for future research.

In future, our work will pursue several research directions: (i) overcoming CTF-AC's increase in computational costs as a result of calculating the factor matrix constraint term in the objective function. We intend to rewrite this calculation into a parallel algorithm to improve overall operational efficiency. (ii) CTF-PSF is based on a predetermined number of shared components and it does not consider the weights of those shared and unshared components. Therefore, in future work, we will strive to make our framework more accurate and robust by adding more constraints.

Data Availability

The data sources are available from <http://www.models.life.ku.dk/3Dnosedata>, <http://www.medinfo.cs.ucy.ac.cy> and <http://www.models.life.ku.dk/joda/prototype>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (with nos. 61702146, 61772163, and 61761136010) and Science and Technology Program of Zhejiang Province (no. 2018C04001). Previous work has accepted as a regular paper and published in International Joint Conference on Neural Networks (IJCNN 2018), and we have cited it as [35].

References

- [1] Q. Jiang, X. Shi, G. Li, B. Liu, L. Wu, and S. Liu, "Integrating supply and demand chains in personalized recommendation via chain-coupled tensor factorization," in *Proceedings of the 2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 420–425, IEEE, 2017.
- [2] Q. Do, W. Liu, and F. Chen, "Discovering both explicit and implicit similarities for cross-domain recommendation," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 618–630, Springer, 2017.
- [3] P. Bhargava, T. Phan, J. Zhou, and J. Lee, "Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data," in *Proceedings of the 24th International Conference on World Wide Web, WWW 2015*, pp. 130–140, May 2015.
- [4] E. Acar, R. Bro, and A. K. Smilde, "Data Fusion in Metabolomics Using Coupled Matrix and Tensor Factorizations," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1602–1620, 2015.
- [5] E. Acar, A. J. Lawaetz, M. A. Rasmussen, and R. Bro, "Structure-revealing data fusion model with applications in metabolomics," in *Proceedings of the 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2013*, pp. 6023–6026, July 2013.
- [6] G. Zhou, Q. Zhao, Y. Zhang, T. Adali, S. Xie, and A. Cichocki, "Linked Component Analysis from Matrices to High-Order Tensors: Applications to Biomedical Data," *Proceedings of the IEEE*, vol. 104, no. 2, pp. 310–331, 2016.
- [7] E. Acar, Y. Levin-Schwartz, V. D. Calhoun, and T. Adal, "Tensor-based fusion of EEG and fMRI to understand neurological changes in schizophrenia," 2016, <https://arxiv.org/abs/1612.02189>.
- [8] B. Hunyadi, W. Van Paesschen, M. De Vos, and S. Van Huffel, "Fusion of electroencephalography and functional magnetic resonance imaging to explore epileptic network activity," in *Proceedings of the 24th European Signal Processing Conference, EUSIPCO 2016*, pp. 240–244, September 2016.
- [9] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 650–658, ACM, August 2008.
- [10] L. Badea, "Extracting gene expression profiles common to colon and pancreatic adenocarcinoma using simultaneous nonnegative matrix factorization," *Pacific Symposium on Biocomputing*, vol. 290, no. 13, pp. 279–290, 2008.
- [11] C. H. Lee, B. O. Alpert, P. Sankaranarayanan, and O. Alter, "GSVD comparison of patient-matched normal and tumor aCGH profiles reveals global copy-number alterations predicting glioblastoma multiforme survival," *PLoS ONE*, vol. 7, no. 1, p. e30098, 2012.
- [12] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM'08*, pp. 931–940, ACM, October 2008.
- [13] M. Jiang, P. Cui, R. Liu et al., "Social contextual recommendation," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp. 45–54, ACM, October 2012.
- [14] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 1545–1553, 2002.
- [15] K. Naskovska, A. A. Korobkov, M. Haardt, and J. Haueisen, "Analysis of the photic driving effect via joint EEG and MEG data processing based on the coupled CP decomposition," in *Proceedings of the 25th European Signal Processing Conference, EUSIPCO 2017*, pp. 1285–1289, September 2017.
- [16] E. Acar, E. E. Papalexakis, G. Gürdeniz et al., "Structure-revealing data fusion," *BMC Bioinformatics*, vol. 15, no. 1, article no. 239, 2014.
- [17] L. Zou, X. Chen, X. Ji, and Z. J. Wang, "Underdetermined Joint Blind Source Separation of Multiple Datasets," *IEEE Access*, vol. 5, pp. 7474–7487, 2017.
- [18] Q. Jiang, X. Shi, G. Li, B. Liu, L. Wu, and S. Liu, "Integrating supply and demand chains in personalized recommendation via chain-coupled tensor factorization," in *Proceedings of the 2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 420–425, 2017.
- [19] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang, "Collaborative filtering meets mobile recommendation: A user-centered approach," *Association for the Advancement of Artificial Intelligence*, vol. 10, pp. 236–241, 2010.
- [20] B. Ermis, E. Acar, and A. T. Cemgil, "Link prediction via generalized coupled tensor factorisation," 2012, <https://arxiv.org/abs/1208.6231>.
- [21] E. Acar, Y. Levin-Schwartz, V. D. Calhoun, and T. Adali, "Tensor-based fusion of EEG and fMRI to understand neurological changes in schizophrenia," in *Proceedings of the 50th IEEE International Symposium on Circuits and Systems, ISCAS 2017*, May 2017.
- [22] E. Acar, Y. Levin-Schwartz, V. D. Calhoun, and T. Adali, "ACMTF for fusion of multi-modal neuroimaging data and identification of biomarkers," in *Proceedings of the 25th European Signal Processing Conference, EUSIPCO 2017*, pp. 643–647, IEEE, September 2017.
- [23] E. Acar, T. G. Kolda, and D. M. Dunlavy, "All-at-once optimization for coupled matrix and tensor factorizations," 2011, <https://arxiv.org/abs/1105.3422>.
- [24] E. E. Papalexakis, C. Faloutsos, T. M. Mitchell, P. P. Talukdar, N. D. Sidiropoulos, and B. Murphy, "Turbo-SMT: Accelerating coupled sparse matrix-tensor factorizations by 200x," in *Proceedings of the 2014 SIAM International Conference on Data Mining*, pp. 118–126, SIAM, 2014.
- [25] E. E. Papalexakis, T. M. Mitchell, N. D. Sidiropoulos, C. Faloutsos, P. P. Talukdar, and B. Murphy, "Scoup-SMT: Scalable coupled sparse matrix-tensor factorization," 2013, <https://arxiv.org/abs/1302.7043>.

- [26] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 2, p. 16, 2016.
- [27] D. Lahat, T. Adali, and C. Jutten, "Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–1477, 2015.
- [28] N. Vervliet and L. De Lathauwer, *Numerical Optimization Based Algorithms for Data Fusion*, 2018.
- [29] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [30] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra and Its Applications*, vol. 18, no. 2, pp. 95–138, 1977.
- [31] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011.
- [32] E. Acar, M. A. Rasmussen, F. Savorani, T. Næs, and R. Bro, "Understanding data fusion within the framework of coupled matrix and tensor factorizations," *Chemometrics and Intelligent Laboratory Systems*, vol. 129, pp. 53–63, 2013.
- [33] Q. Do and W. Liu, "ASTEN: An accurate and scalable approach to Coupled Tensor Factorization," in *Proceedings of the 2016 International Joint Conference on Neural Networks, IJCNN 2016*, pp. 99–106, IEEE, July 2016.
- [34] L. De Lathauwer and E. Kofidis, "Coupled matrix-tensor factorizations the case of partially shared factors," ESAT-STADIUS 17-171, Katholieke Universiteit Leuven, Leuven, Belgium, Accepted for publication in 2017 51st Asilomar Conference on Signals, Systems and Computers, 2017.
- [35] X. L. Q. Wu, J. F. Q. Do, and J. W. R. Ge, "CTF-PSF: Coupled tensor factorization with partially shared factors," in *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2018.
- [36] T. Skov and R. Bro, "A new approach for modelling sensor based data," *Sensors and Actuators B: Chemical*, vol. 106, no. 2, pp. 719–729, 2005.




Hindawi

Submit your manuscripts at
www.hindawi.com

