All-at-once Optimization for Coupled Matrix and Tensor Factorizations

Evrim Acar
Faculty of Life Sciences,
University of Copenhagen
evrim@life.ku.dk

Tamara G. Kolda
Sandia National Laboratories
Livermore, CA 94551-9159
tgkolda@sandia.gov

Daniel M. Dunlavy
Sandia National Laboratories
Albuquerque, NM 87185-1318
dmdunla@sandia.gov

ABSTRACT

Joint analysis of data from multiple sources has the potential to improve our understanding of the underlying structures in complex data sets. For instance, in restaurant recommendation systems, recommendations can be based on rating histories of customers. In addition to rating histories, customers' social networks (e.g., Facebook friendships) and restaurant categories information (e.g., Thai or Italian) can also be used to make better recommendations. The task of fusing data, however, is challenging since data sets can be incomplete and heterogeneous, i.e., data consist of both matrices, e.g., the *person* by *person* social network matrix or the *restaurant* by *category* matrix, and higher-order tensors, e.g., the "ratings" tensor of the form *restaurant* by *meal* by *person*.

In this paper, we are particularly interested in fusing data sets with the goal of capturing their underlying latent structures. We formulate this problem as a coupled matrix and tensor factorization (CMTF) problem where heterogeneous data sets are modeled by fitting outer-product models to higher-order tensors and matrices in a coupled manner. Unlike traditional approaches solving this problem using alternating algorithms, we propose an all-at-once optimization approach called CMTF-OPT (CMTF-OPTimization), which is a gradient-based optimization approach for joint analysis of matrices and higher-order tensors. We also extend the algorithm to handle coupled incomplete data sets. Using numerical experiments, we demonstrate that the proposed all-at-once approach is more accurate than the alternating least squares approach.

Keywords

data fusion, matrix factorizations, tensor factorizations, CANDECOMP/PARAFAC, missing data

1. INTRODUCTION

With the ability to access massive amounts of data as a result of recent technological advances, e.g., the Internet, communication and multi-media devices, genomic technologies and new medical diagnostic techniques, we are faced with data sets from multiple sources. For instance, in restaurant recommendation systems, online review sites like Yelp have access to shopping histories of customers, friendship networks of those customers, as well as categorizations of the restaurants. Similarly, for medical diagnoses, several types of data are collected from a patient; for example, EEG (electroencephalogram) and ECG (electrocardiogram) monitoring data, fMRI (functional Magnetic Resonance Imaging) scans, and other data gathered from laboratory tests.

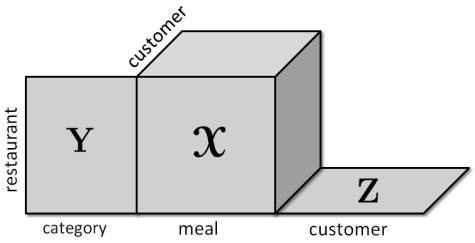


Figure 1: Coupled Data Sets of Different Orders. Each tensor entry indicates the rating of a customer for a specific meal (i.e., breakfast, lunch, dinner) at a particular restaurant. Matrices $\mathbf{Y}$ and $\mathbf{Z}$ show restaurant categories (i.e., Thai, Chinese, Italian) and social network information, respectively. Thus, the third-order tensor $\mathcal{X}$ of type *restaurant* by *meal* by *customer* can be coupled with the *restaurant* by *category* matrix $\mathbf{Y}$ and the *customer* by *customer* matrix $\mathbf{Z}$.

Analysis of data from multiple sources requires handling of data sets of different orders [7, 25, 30], i.e., matrices and/or higher-order tensors. For instance, Banerjee et al. [7], discusses the problem of analyzing heterogeneous data sets with a goal of simultaneously clustering different classes of entities based on multiple relations, where each relation is represented as a matrix (e.g., *movies* by *review words* matrix showing movie reviews) or a higher-order tensor (e.g., *movies* by *viewers* by *actors* tensor showing viewers' ratings). Similarly, coupled analysis of matrices and tensors has been a topic of interest in the areas of community detection [22], collaborative filtering [36], and chemometrics [30].

As an example of data sets from multiple sources, without loss of generality, suppose we have a third-order tensor, $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, and a matrix, $\mathbf{Y} \in \mathbb{R}^{I \times M}$, coupled in the first dimension (mode) of each. The common latent structure in these data sets can be extracted through coupled matrix and tensor factorization (CMTF), where an R-component CMTF model of a tensor $\mathcal{X}$ and a matrix $\mathbf{Y}$ is defined as:

$$f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{V}) = \| \mathcal{X} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \|^2 + \left\| \mathbf{Y} - \mathbf{A}\mathbf{V}^{\mathsf{T}} \right\|^2, \quad (1)$$

where matrices $\mathbf{A} \in \mathbb{R}^{I \times R}, \mathbf{B} \in \mathbb{R}^{J \times R}$ and $\mathbf{C} \in \mathbb{R}^{K \times R}$ are the *factor matrices* of $\mathcal{X}$ extracted using a CANDECOMP/PARAFAC (CP) model [10, 13, 15]. The CP model is one of the most commonly used tensor models in the literature (for a list of CP applications, see reviews [3, 19]).

Here, we use the notation $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ to denote the CP model. Similarly, matrices $\mathbf{A}$ and $\mathbf{V} \in \mathbb{R}^{M \times R}$ are the factor matrices extracted from matrix $\mathbf{Y}$ through matrix factorization. The formulation in (1) easily extends to multiple matrices and tensors, e.g., as shown in Figure 1. We also note that we focus on the least squares error in this paper, but our algorithms can be extended to other loss functions such as Bregman information metric used in, e.g., [7]. We briefly illustrate two motivating applications of coupled matrix and tensor factorizations.

Example 1: Clustering. Joint analysis of data from multiple sources may capture fine-grained clusters that would not be captured by the individual analysis of each data set. Suppose that there is a set of customers and there are two sources of information about these customers, $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ and $\mathbf{Y} \in \mathbb{R}^{I \times M}$, one storing information about which items customers have bought over a period of time, and the other showing where customers live. Within this set of customers, there are 4 groups: $\{G_1, G_2, G_3, G_4\}$ and each group consists of people who live in the same neighborhood and have an interest in similar items. Imagine that the matrix $\mathbf{Y}$ can only discriminate between $(G_1 \cup G_3)$ and $(G_2 \cup G_4)$. A rank-2 matrix SVD factorization of $\mathbf{Y}$ would yield factors that could be used to cluster the customers as in the top plot shown in Figure 2 (SVD), failing to fully separate the four groups. Conversely, imagine that the tensor $\mathbf{X}$ only has enough information to discriminate between $(G_1 \cup G_2)$ and $(G_3 \cup G_4)$. In this case, a rank-2 CP factorization of the tensor would still only separate the data into two groups, albeit two different groups, as illustrated in the middle plot in Figure 2 (CP). If, however, we jointly factor the matrix and tensor simultaneously using CMTF, the four groups are completely separated, as shown in the bottom plot of Figure 2 (CMTF). Details of the data generation for this example are provided in the appendix. □

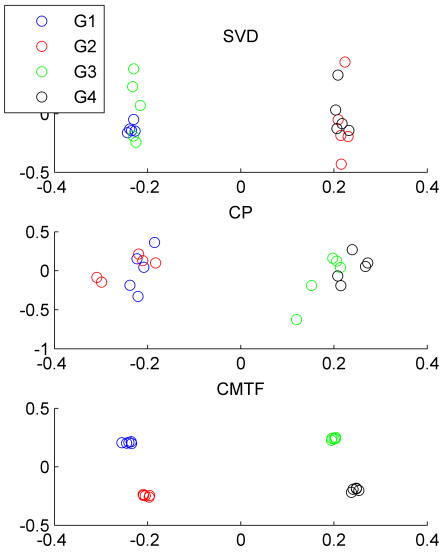


Figure 2: Clustering based on matrix SVD factorization of $\mathbf{Y}$ vs. CP tensor factorization of $\mathcal{X}$ vs. coupled matrix-tensor factorization of $\mathcal{X}$ and $\mathbf{Y}$. The subplots present the scatter plots showing the first factor plotted against the second factor in the first mode.

Example 2: Missing Data Recovery. CMTF can be used for missing data recovery when data from different sources have the same underlying low-rank structure (at least in one mode) but some of the data sets have missing entries (Figure 3). If a matrix or a higher-order tensor has a low-rank structure, it is possible to recover the missing entries using a limited number of data entries [1, 9]. However, if there is a large amount of missing data, then the analysis of a single data set is no longer enough for accurate data recovery. Here, we provide an example illustrating that missing entries can still be recovered accurately using CMTF even when the analysis of a single data set fails to do so.

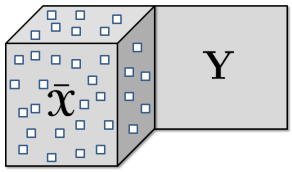


Figure 3: Incomplete tensor $\bar{\mathcal{X}}$ and matrix $\mathbf{Y}$ coupled in the first mode.

Suppose we have a tensor $\mathcal{X}$ and a matrix $\mathbf{Y}$ computed as $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ and $\mathbf{Y} = \mathbf{A}\mathbf{V}^\intercal$, where matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, $\mathbf{C} \in \mathbb{R}^{K \times R}$ and $\mathbf{V} \in \mathbb{R}^{M \times R}$ are generated using random entries drawn from the standard normal distribution. $\bar{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ is constructed by randomly setting $M\%$ of the entries of $\mathcal{X}$ to be missing, i.e., $\bar{\mathcal{X}} = \mathcal{W} * \mathcal{X}$, where the binary tensor $\mathcal{W}$ is the same size as tensor $\mathcal{X}$ and $w_{ijk} = 0$ if we want to set x_{ijk} to missing. In order to recover the missing entries, one approach is to fit an R-component CP model to $\bar{\mathcal{X}}$ and use the extracted factors to recover missing entries. An alternative approach is to fit an R-component CMTF model to $\bar{\mathcal{X}}$ and $\mathbf{Y}$ by extracting a common factor matrix in the first mode and then make use of CMTF factors to recover the missing entries.

Figure 4 illustrates how the recovery error behaves for different amounts of missing data. We observe that CP is accurate in terms of recovering missing entries if less than 80% of the entries are missing. However, there is a sharp in error as we further increase the amount of missing entries. On the other hand, CMTF can compute factors with low recovery error for higher amounts of missing data; only for problems with more that 90% missing data does the recovery error increase[1]. □

In order to solve coupled matrix and tensor factorization, various algorithms have been proposed in the literature using a variety of loss functions [7, 22]. These algorithms all propose an alternating scheme, where the basis for each entity type is determined one at a time. In this paper, we focus solely on the L_2 loss function, which is often solved using alternating least squares (ALS). Especially when fitting tensor models, ALS is the most commonly used algorithm due to its speed and ease of implementation. On the other hand, ALS suffers from several problems: (i) It may fail to find the underlying components accurately if the number of components is not correctly estimated [2, 32]; (ii) In

[1]Note that the amount of missing data where the recovery error makes a sharp increase may change depending on the values of I, J, K, V and R. For example, with small data sizes and large R, dealing with missing data is challenging[1].

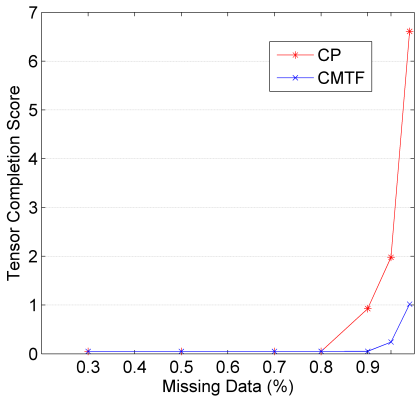


Figure 4: Missing data recovery using CP factorization of $\bar{\mathcal{X}}$ vs. CMTF of $\bar{\mathcal{X}}$ and $\mathbf{Y}$. Tensor Completion Score is the recovery error measure used here and is defined as $\frac{\|\,(1-\mathcal{W})*(\mathcal{X}-\hat{\mathcal{X}})\,\|}{\|\,(1-\mathcal{W})*\mathcal{X}\,\|}$, where $\hat{\mathcal{X}}$ corresponds to the data reconstructed using the extracted factor matrices, i.e., $\hat{\mathbf{A}}, \hat{\mathbf{B}}$, and $\hat{\mathbf{C}}$, as $\hat{\mathcal{X}} = [\![\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}]\!]$.

the presence of missing data, ALS-based imputation techniques may suffer from poor convergence [8] and do not scale to large-scale data sets [1]. Therefore, unlike the previous work using alternating least squares algorithms, we propose an all-at-once optimization approach solving for all variables simultaneously. Our contributions in this paper are

- Developing an algorithm called CMTF-OPT (CMTF-OPTimization) based on first-order optimization to solve coupled matrix and tensor factorization problem for a given number of components (R).[2]
- Extending CMTF-OPT algorithm to handle incomplete data sets, i.e., data with missing or unknown entries.
- Demonstrating that CMTF-OPT is more accurate than an alternating least squares approach using numerical experiments.

This paper is organized as follows. In §2, we introduce the notation for tensors and tensor operations. After discussing the related work on coupled data analysis in §3, we introduce our CMTF-OPT algorithm and its extension to data with missing entries in §4. Section 5 describes numerical experiments and demonstrates how CMTF-OPT compares with CMTF-ALS in terms of capturing the underlying factors in coupled data sets. Finally, we conclude with future research directions in §6.

2. NOTATION AND BACKGROUND

Tensors of order $N \geq 3$ are denoted by Euler script letters ($\mathcal{X}, \mathcal{Y}, \mathcal{Z}$), matrices are denoted by boldface capital letters ($\mathbf{A}, \mathbf{B}, \mathbf{C}$), vectors are denoted by boldface lowercase letters ($\mathbf{a}, \mathbf{b}, \mathbf{c}$), and scalars are denoted by lowercase letters (a, b, c). Columns of a matrix are denoted by boldface lower letters with a subscript, e.g., $\mathbf{a}_r$ is the rth column of matrix $\mathbf{A}$. Entries of a matrix or a tensor are denoted by lowercase letters with subscripts, i.e., the $(i_1, i_2, \ldots, i_N)$ entry of an N-way tensor $\mathcal{X}$ is denoted by $x_{i_1 i_2 \cdots i_N}$.

Given a matrix $\mathbf{A}$ of size $I \times J$, $\mathsf{vec}(\mathbf{A})$ stacks the columns of the matrix and forms a vector of length IJ:

$$\mathsf{vec}(\mathbf{A}) = \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_J \end{bmatrix} \in \mathbb{R}^{IJ}.$$

Given two matrices $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$, their Khatri-Rao product is denoted by $\mathbf{A} \odot \mathbf{B}$ and defined as columnwise Kronecker product. The result is a matrix of size $(IJ) \times K$ and defined by

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_K \otimes \mathbf{b}_K \end{bmatrix},$$

where $\otimes$ denotes Kronecker product. For more details on properties of Kronecker and Khatri-Rao products, see [19].

An N-way tensor can be rearranged as a matrix; this is called *matricization*. The mode-n matricization of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\mathbf{X}_{(n)}$ and arranges the mode-n one-dimensional "fibers" to be the columns of the resulting matrix.

Given two tensors $\mathcal{X}$ and $\mathcal{Y}$ of equal size $I_1 \times I_2 \times \cdots \times I_N$, their Hadamard (elementwise) product is denoted by $\mathcal{X} * \mathcal{Y}$ and defined as

$$(\mathcal{X} * \mathcal{Y})_{i_1 i_2 \cdots i_N} = x_{i_1 i_2 \cdots i_N} y_{i_1 i_2 \cdots i_N}$$

for all $i_n \in \{1, \ldots, I_n\}$ and $n \in \{1, \ldots, N\}$. Their inner product, denoted by $\langle \mathcal{X}, \mathcal{Y} \rangle$, is the sum of the products of their entries, i.e.,

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \cdots i_N} y_{i_1 i_2 \cdots i_N}.$$

For a tensor $\mathcal{X}$ of size $I_1 \times I_2 \times \cdots \times I_N$, its *norm* is $\|\,\mathcal{X}\,\| = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$. For matrices and vectors, $\|\cdot\|$ refers to the analogous Frobenius and two-norm, respectively.

Given a sequence of matrices $\mathbf{A}^{(n)}$ of size $I_n \times R$ for $n = 1, \ldots, N$, the notation $[\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!]$ defines an $I_1 \times I_2 \times \cdots \times I_N$ tensor whose elements are given by

$$\left([\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!]\right)_{i_1 i_2 \cdots i_N} = \sum_{r=1}^{R} \prod_{n=1}^{N} a_{i_n r}^{(n)},$$

for $i_n \in \{1, \ldots, I_n\}, n \in \{1, \ldots, N\}$. For just two matrices, this reduces to $[\![\mathbf{A}, \mathbf{B}]\!] = \mathbf{A}\mathbf{B}^{\mathsf{T}}$.

3. RELATED WORK IN DATA FUSION

Data fusion, also called collective data analysis, multiblock, multi-view or multi-set data analysis, has been a topic of interest in different fields for decades. First, we briefly discuss data fusion techniques for multiple data sets each represented as a matrix and then focus on techniques proposed for coupled analysis of heterogenous data sets.

3.1 Collective Factorization of Matrices

The analysis of data from multiple sources attracted considerable attention in the data mining community during the Netflix Prize competition [20], where the goal was to make accurate predictions about movie ratings. In order to achieve better rating predictions, additional data sources complementing user ratings such as tagging information have been exploited; e.g., users tag movies [33] as well as features

[2]Determining the number of components R in CMTF remains as a challenge just like computing the tensor rank, which is NP-hard [14].

of movies such as movie types or movie players . Singh and Gordon [28] proposed Collective Matrix Factorization (CMF) to take advantage of correlations between different data sets and simultaneously factorize coupled matrices. Given two matrices $\mathbf{X}$ and $\mathbf{Y}$ of size $I \times M$ and $I \times L$, respectively, CMF can be formulated as

$$f(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \left\| \mathbf{X} - \mathbf{U}\mathbf{V}^\mathsf{T} \right\|^2 + \left\| \mathbf{Y} - \mathbf{U}\mathbf{W}^\mathsf{T} \right\|^2, \qquad (2)$$

where $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$ are factor matrices of size $I \times R$, $M \times R$ and $L \times R$, respectively and R is the number of factors. This formulation is a special case of the general approach introduced in [28], which extends to different loss functions. Earlier, Long et al. [24, 23] had also studied collective matrix factorization using a different matrix factorization scheme than $\mathbf{X} = \mathbf{U}\mathbf{V}^\mathsf{T}$. The proposed approaches in those studies are based on alternating algorithms solving the collective factorization problem for one factor matrix at a time.

Analysis of multiple matrices dates back to one of the earliest models aiming to capture the common variation in two data sets, i.e., Canonical Correlation Analysis (CCA) [16]. Later, other studies followed CCA by extending it to more than two data sets [18], focusing on simultaneous factorization of Gramian matrices [21] and working on PCA of multiple matrices [17, 34]. Moreover, several approaches for simultaneous factor analysis have been developed for specific applications as well; e.g., population differentiation in biology [31], blind source separation [37], multimicrophone speech filtering [11], and microarray data analysis [4, 5, 6].

Tensor factorizations [3, 19, 29] can also be considered as one way of analyzing multiple matrices. For instance, when a tensor model is fit to a third-order tensor, multiple coupled matrices are analyzed simultaneously. Nevertheless, neither CMF nor tensor factorizations can handle coupled analysis of heterogeneous data, which we address next.

3.2 Collective Factorization of Mixed Data

As described in §1, heterogeneous data consists of data sets of different orders, i.e., both matrices and higher-order tensors. The formulation in (2) can be extended to heterogeneous data sets: Given a tensor $\mathcal{X}$ and a matrix $\mathbf{Y}$ of sizes $I \times J \times K$ and $I \times M$, respectively, we can formulate their factorization coupled in the first mode as shown in (1). Note that (1) can be considered as a special case of the approach introduced by Smilde et al. [30] for multi-way multi-block data analysis, where different tensor models can be fit to higher-order data sets and the factor matrices corresponding to the coupled modes do not necessarily match. The same formulation as in (1) has recently been studied in psychometrics as Linked-Mode PARAFAC-PCA [35]. As a more general framework, not restricted to squared Euclidean distance, Banerjee et al. [7] introduced a multi-way clustering approach for relational and multi-relational data where coupled analysis of multiple data sets including higher-order data sets were studied using minimum Bregman information. The paper [22] also discussed coupled analysis of multiple tensors and matrices using nonnegative factorization by formulating the problem using KL-divergence. All these studies propose algorithms that are based on alternating approaches.

In this paper we focus on the squared Euclidean distance as the loss function as in (1). Rewriting $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ as $\mathbf{X}_{(1)} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})^\mathsf{T}$, $\mathbf{X}_{(2)} = \mathbf{B}(\mathbf{C} \odot \mathbf{A})^\mathsf{T}$, etc., for the different

while not "converged" **do**
  rescale all factors to unit Frobenious norm
  solve for $\mathbf{A}$ (for fixed $\mathbf{B}, \mathbf{C}, \mathbf{V}$)
    $\min_{\mathbf{A}} \left\| \begin{bmatrix} \mathbf{X}_{(1)} & \mathbf{Y} \end{bmatrix} - \mathbf{A} \begin{bmatrix} (\mathbf{C} \odot \mathbf{B})^\mathsf{T} & \mathbf{V}^\mathsf{T} \end{bmatrix} \right\|^2$
  solve for $\mathbf{B}$ (for fixed $\mathbf{A}, \mathbf{C}, \mathbf{V}$)
    $\min_{\mathbf{B}} \left\| \mathbf{X}_{(2)} - \mathbf{B}(\mathbf{C} \odot \mathbf{A})^\mathsf{T} \right\|^2$
  solve for $\mathbf{C}$ (for fixed $\mathbf{A}, \mathbf{B}, \mathbf{V}$)
    $\min_{\mathbf{C}} \left\| \mathbf{X}_{(3)} - \mathbf{C}(\mathbf{B} \odot \mathbf{A})^\mathsf{T} \right\|^2$
  solve for $\mathbf{V}$ (for fixed $\mathbf{A}, \mathbf{B}, \mathbf{C}$)
    $\min_{\mathbf{V}} \left\| \mathbf{Y} - \mathbf{A}\mathbf{V}^\mathsf{T} \right\|^2$
end while

Figure 5: CMTF-ALS: Alternating Least Squares Algorithm for coupled matrix and tensor factorization of a third-order tensor $\mathcal{X}$ and matrix $\mathbf{Y}$ coupled in the first mode.

modes of $\mathcal{X}$, we summarize the steps of an alternating least squares algorithm for solving (1) in Figure 5. The main loop in the algorithm is often terminated as a function of the relative change in function value (e.g., as defined below in (5)), a function of the relative change in factor matrices, and/or after a prescribed number of iterations.

ALS-based algorithms are simple to implement and computationally efficient; however, ALS has shown to be error-prone when fitting a CP model in the case of overfactoring, i.e., the number of extracted components is more than the true number of underlying components [2, 32]. Furthermore, in the presence of missing data, ALS-based techniques may have poor convergence [8] and do not scale to very large data sets [1]. On the other hand, all-at-once optimization, in other words, solving for all CP factor matrices simultaneously has shown to be more robust to overfactoring [2, 32] and easily extends to handle data with missing entries even for very large data sets [1]. Therefore, in order to deal with these issues, we develop an algorithm called CMTF-OPT, which formulates coupled analysis of heterogeneous data sets just like [30, 35] but solves for all factor matrices simultaneously using a gradient-based optimization approach.

4. CMTF-OPT ALGORITHM

In this section, we consider joint analysis of a matrix and an Nth-order tensor with one mode in common, where the tensor is factorized using an R-component CP model and the matrix is factorized by extracting R factors using matrix factorization. Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\mathbf{Y} \in \mathbb{R}^{I_1 \times M}$ have the nth mode in common, where $n \in \{1, \ldots, N\}$. The objective function for coupled analysis of these two data sets is defined by

$$\begin{aligned} &f(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}, \mathbf{V}) \\ &= \frac{1}{2} \left\| \mathcal{X} - [\![\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!] \right\|^2 + \frac{1}{2} \left\| \mathbf{Y} - \mathbf{A}^{(n)}\mathbf{V}^\mathsf{T} \right\|^2 \end{aligned} \qquad (3)$$

Our goal is to find the matrices $\mathbf{A}^{(i)} \in \mathbb{R}^{I_i \times R}$ for $i = 1, 2, ...N$ and matrix $\mathbf{V} \in \mathbb{R}^{M \times R}$ that minimize the objective in (3). In order to solve this optimization problem, we can compute the gradient and then use any first-order optimization algorithm [27]. Next, we discuss the computation of the gradient for (3).

We can rewrite (3) as two components, f_1 and f_2:

$$f = \frac{1}{2}\underbrace{\left\| \mathcal{X} - [\![\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!] \right\|^2}_{f_1(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)})} + \frac{1}{2}\underbrace{\left\| \mathbf{Y} - \mathbf{A}^{(n)}\mathbf{V}^\mathsf{T} \right\|^2}_{f_2(\mathbf{A}^{(n)}, \mathbf{V})}$$

The partial derivative of f_1 with respect to $\mathbf{A}^{(i)}$ has been derived in [2] so we just present the results here.

Let $\mathcal{Z} = [\![\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$, then

$$\frac{\partial f_1}{\partial \mathbf{A}^{(i)}} = (\mathbf{Z}_{(i)} - \mathbf{X}_{(i)})\mathbf{A}^{(-i)}$$

where

$$\mathbf{A}^{(-i)} = \mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(i+1)} \odot \mathbf{A}^{(i-1)} \odot \cdots \odot \mathbf{A}^{(1)},$$

for $i = 1, \ldots, N$.

The partial derivatives of the second component, f_2, with respect to $\mathbf{A}^{(i)}$ and $\mathbf{V}$ can be computed as

$$\frac{\partial f_2}{\partial \mathbf{A}^{(i)}} = \begin{cases} -\mathbf{Y}\mathbf{V} + \mathbf{A}^{(-i)}\mathbf{V}^\mathsf{T}\mathbf{V}, & \text{for } i = n, \\ 0 & \text{for } i \neq n, \end{cases}$$

$$\frac{\partial f_2}{\partial \mathbf{V}} = -\mathbf{Y}^\mathsf{T}\mathbf{A}^{(i)} + \mathbf{V}{\mathbf{A}^{(i)}}^\mathsf{T}\mathbf{A}^{(i)}.$$

Combining the above results, we can compute the partial derivative of f with respect to factor matrix $\mathbf{A}^{(i)}$, for $i = 1, 2, ..., N$, and $\mathbf{V}$ as:

$$\frac{\partial f}{\partial \mathbf{A}^{(i)}} = \frac{\partial f_1}{\partial \mathbf{A}^{(i)}} + \frac{\partial f_2}{\partial \mathbf{A}^{(i)}}$$
$$\frac{\partial f}{\partial \mathbf{V}} = \frac{\partial f_2}{\partial \mathbf{V}}$$

Finally, the gradient of f, which is a vector of size $P = R(\sum_{n=1}^{N} I_n + M)$, can be formed by vectorizing the partial derivatives with respect to each factor matrix and concatenating them all, i.e.,

$$\nabla f = \begin{bmatrix} \mathsf{vec}\left(\frac{\partial f}{\partial \mathbf{A}^{(1)}}\right) \\ \vdots \\ \mathsf{vec}\left(\frac{\partial f}{\partial \mathbf{A}^{(N)}}\right) \\ \mathsf{vec}\left(\frac{\partial f}{\partial \mathbf{V}}\right) \end{bmatrix}$$

4.1 CMTF-OPT for Incomplete Data

In the presence of missing data, it is still possible to do coupled analysis by ignoring the missing entries and fitting the tensor and/or the matrix model to the known data entries. Here we study the case where tensor $\mathcal{X}$ has missing entries as in Figure 3. Let $\mathcal{W} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ indicate the missing entries of $\mathcal{X}$ such that

$$w_{i_1 i_2 \cdots i_N} = \begin{cases} 1 & \text{if } x_{i_1 i_2 \cdots i_N} \text{ is known,} \\ 0 & \text{if } x_{i_1 i_2 \cdots i_N} \text{ is missing,} \end{cases}$$

for all $i_n \in \{1, \ldots, I_n\}$ and $n \in \{1, \ldots, N\}$. We can then modify the objective function (3) as

$$f_{\mathcal{W}}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}, \mathbf{V}) = \frac{1}{2}\underbrace{\left\| \mathcal{W} * \left(\mathcal{X} - [\![\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]\right) \right\|^2}_{f_{\mathcal{W}_1}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)})} + \frac{1}{2}\left\| \mathbf{Y} - \mathbf{A}^{(n)}\mathbf{V}^\mathsf{T} \right\|^2$$

The first term, $f_{\mathcal{W}_1}$, corresponds to the weighted least squares problem for fitting a CP model while the second term stays the same as in (3). The partial derivative of $f_{\mathcal{W}_1}$ with respect to $\mathbf{A}^{(i)}$ can be computed as in [1] as

$$\frac{\partial f_{\mathcal{W}_1}}{\partial \mathbf{A}^{(i)}} = \left(\mathbf{W}_{(i)} * \mathbf{Z}_{(i)} - \mathbf{W}_{(i)} * \mathbf{X}_{(i)}\right)\mathbf{A}^{(-i)},$$

for $i = 1, \ldots, N$. Since the partial derivatives of the second term do not change, we can write the partials of f with respect to $\mathbf{A}^{(i)}$ for $i = 1, 2, ..., N$, and $\mathbf{V}$ as:

$$\frac{\partial f_{\mathcal{W}}}{\partial \mathbf{A}^{(i)}} = \begin{cases} \frac{\partial f_{\mathcal{W}_1}}{\partial \mathbf{A}^{(i)}} & \text{for } i \in \{1, \ldots, N\}/\{n\}, \\ \frac{\partial f_{\mathcal{W}_1}}{\partial \mathbf{A}^{(i)}} + \frac{\partial f_2}{\partial \mathbf{A}^{(i)}} & \text{for } i = n. \end{cases}$$

$$\frac{\partial f_{\mathcal{W}}}{\partial \mathbf{V}} = \frac{\partial f_2}{\partial \mathbf{V}}$$

The gradient of $f_{\mathcal{W}}$, $\nabla f_{\mathcal{W}}$, is also a vector of size $P = R(\sum_{n=1}^{N} I_n + M)$ and can be formed in the same way as ∇f.

Once we have the function, f (or $f_{\mathcal{W}}$), and gradient, ∇f (or $\nabla f_{\mathcal{W}}$), values, we can use any gradient-based optimization algorithm to compute the factor matrices. For the results presented in this paper, we use the Nonlinear Conjugate Gradient (NCG) with Hestenes-Steifel updates [27] and the Moré-Thuente line search [26] as implemented in the Poblano Toolbox [12].

5. EXPERIMENTS

We compare the performance of the proposed CMTF-OPT algorithm with the ALS-based approach (i.e., CMTF-ALS as shown in Figure 5) in terms of accuracy using randomly generated matrices and tensors. Our goal is to see whether the algorithms can capture the underlying factors in the data (i) when $\bar{R} = R$ factors are extracted, and (ii) in the case of overfactoring, e.g., when $\bar{R} = R + 1$ factors are extracted, where $\bar{R}$ and R denote the extracted and true number of factors.

5.1 Data Generation

In our experiments, three different scenarios are used (see Table 1). In the first case, we have a tensor $\mathcal{X}$ and a matrix $\mathbf{Y}$ coupled in the first mode. In the second case, we have two third-order tensors $\mathcal{X}$ and $\mathcal{Y}$ coupled in one dimension. The third case has a third-order tensor $\mathcal{X}$ and two matrices $\mathbf{Y}$ and $\mathbf{Z}$ such that the tensor shares one mode with each one of the matrices.

To construct the tensors and matrices in each scenario, random factor matrices with entries following standard normal distribution are generated and a tensor (or a matrix) is formed based on a CP (or a matrix) model. For example, for the first scenario, we generate random factor matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$ and $\mathbf{C} \in \mathbb{R}^{K \times R}$, and form a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ based on a CP model. Gaussian noise is later added to the tensor, i.e., $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] + \eta\mathcal{N}\frac{\|[\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]\|}{\|\mathcal{N}\|}$, 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. Similarly, we generate a factor matrix $\mathbf{V} \in \mathbb{R}^{I \times R}$ and form matrix $\mathbf{Y} = \mathbf{A}\mathbf{V}^\mathsf{T} + \eta\mathbf{N}\frac{\|\mathbf{A}\mathbf{V}^\mathsf{T}\|}{\|\mathbf{N}\|}$. The matrix $\mathbf{N} \in \mathbb{R}^{I \times R}$ is a matrix with entries drawn from the standard normal distribution and is used to introduce differing amounts of noise in the data. In our experiments, we

use three different noise levels, i.e., $\eta = 0.1, 0.25$ and 0.35, and the true number of factors is $R = 3$.

5.2 Performance Metric

We fit an $\bar{R}$-component CMTF model using CMTF-OPT and CMTF-ALS algorithms, where $\bar{R} = R$ and $\bar{R} = R + 1$, and compare the algorithms in terms of accuracy; in other words, how well the factors extracted by the algorithms match with the original factors used to generate the data. For instance, for the first scenario described above, let $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{V}$ be the original factor matrices and $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}$ and $\hat{\mathbf{V}}$ be the extracted factor matrices. We quantify how well the extracted factors match with the original ones using the factor match score (FMS) defined as follows:

$$\text{FMS} = \min_r (1 - \frac{|\xi_r - \hat{\xi}_r|}{\max(\xi_r, \hat{\xi}_r)}) |\mathbf{a}_r^\mathsf{T} \hat{\mathbf{a}}_r \mathbf{b}_r^\mathsf{T} \hat{\mathbf{b}}_r \mathbf{c}_r^\mathsf{T} \hat{\mathbf{c}}_r \mathbf{v}_r^\mathsf{T} \hat{\mathbf{v}}_r|. \quad (4)$$

Here, the columns of factor matrices are normalized to unit norm and ξ_r denotes the weight for each factor r, for $r = 1, 2, ...R$. The weight for each factor is computed as follows: We can rewrite $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ as $\mathcal{X} = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$, where $\circ$ denotes n-mode vector outer product; the columns of the factor matrices are normalized to unit norm and λ_r is the product of the vector norms in each mode. Similarly, we can rewrite $\mathbf{Y} = \mathbf{A}\mathbf{V}^\mathsf{T}$ as $\mathbf{Y} = \sum_{r=1}^{R} \alpha_r \mathbf{a}_r \circ \mathbf{v}_r$ by normalizing the matrix columns and using α_r to denote the product of vector norms. We define the norm of each CMTF component, i.e., ξ_r, as $\xi_r = \lambda_r + \alpha_r$. In (4), ξ_r and $\hat{\xi}_r$ indicate the weights corresponding to the original and extracted rth CMTF component, respectively. If the extracted and original factors perfectly match, FMS value will be 1. It is also considered a success if FMS is above a certain threshold, i.e., $(0.99)^N$, where N is the number of factor matrices.

5.3 Stopping Conditions

As a stopping condition, both algorithms use the relative change in function value and stop when

$$\frac{|f_{\text{current}} - f_{\text{previous}}|}{f_{\text{previous}}} \leq 10^{-8}, \quad (5)$$

where f is as given in (3). Additionally, for CMTF-ALS, the maximum number of iterations is set to 10^4. For CMTF-OPT, the maximum number of function values (which corresponds the number of iterations in an ALS algorithm) is set to 10^4 and the maximum number of iterations is set to 10^3. CMTF-OPT also uses the two-norm of the gradient divided by the number of entries in the gradient and the tolerance for that is set to 10^{-8}. In the experiments, the algorithms have generally stopped due to the relative change in function value criterion. However, we also rarely observe that both CMTF-OPT and CMTF-ALS have stopped since the algorithms have reached the maximum number of iterations, i.e., approximately 2% of all runs for CMTF-OPT and 0.3% of all runs for CMTF-ALS. Furthermore, for around 2% of all runs, CMTF-OPT has stopped by satisfying the condition on the gradient.

5.4 Results

We demonstrate the performance of the algorithms in terms of accuracy in Table 1 and Table 2. Table 1 presents the results for the experiments where factor matrices are generated with columns normalized to unit norm. For instance, for the example discussed above, this corresponds to generating tensor $\mathcal{X}$ and matrix $\mathbf{Y}$ using $\lambda_r = \alpha_r = 1$, for $r = 1, 2, ..R$. In Table 1, for all different scenarios, we observe that when the correct number of underlying factors are extracted, i.e., $\bar{R} = R$, the success ratios of both algorithms are compatible. Since we repeat our experiments with 30 different sets of factor matrices for each set of parameters, we report the average factor match score for 30 runs. The average scores for both algorithms are quite close and p-values computed for paired-sample t-tests indicate that differences in the scores are not statistically significant. On the other hand, for all scenarios, when we look at the cases where the data is overfactored, i.e., $\bar{R} = R + 1$, CMTF-OPT significantly outperforms CMTF-ALS. While CMTF-OPT algorithm is quite accurate in terms of recovering the underlying factors in the case of overfactoring, the accuracy of CMTF-ALS is quite low. We also observe that the increase in the noise level barely affects the accuracies of the algorithms.

In Table 2, we present the results for a harder set of experiments, where the norm of each tensor and matrix component, i.e., λ_r, α_r for $r = 1, 2, ..R$, is a randomly assigned integer greater than or equal to 1.[3] Similar to the previous case, we observe that CMTF-OPT is more robust to overfactoring compared to CMTF-ALS. However, the accuracies reported in this table are lower compared to Table 1. In particular, as the noise level increases, it becomes harder to find the underlying components and accuracies drop even if the true number of underlying factors are extracted from the data. For instance, for the first scenario, when the noise level is 0.35, the accuracy of CMTF-OPT is 60% for the case we extract the correct number of components. This is in part due to not being able to find ξ_r, for $r = 1, 2, ..R$, accurately. If we slightly change the FMS such that we only require $\min_r |\mathbf{a}_r^\mathsf{T} \hat{\mathbf{a}}_r \mathbf{b}_r^\mathsf{T} \hat{\mathbf{b}}_r \mathbf{c}_r^\mathsf{T} \hat{\mathbf{c}}_r \mathbf{v}_r^\mathsf{T} \hat{\mathbf{v}}_r|$ to be greater than the threshold, then the average accuracy goes up to around 73%. The rest of the failing runs is due to the fact that extracted factors are distorted compared to the original factors used to generate the data. Note that factor match scores are still high which indicates that the factors are only slightly distorted.

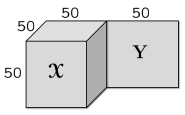
6. CONCLUSIONS

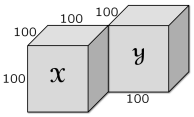
We have seen a shift in data mining in recent years: from models focusing on matrices to those studying higher-order tensors and now we are in need of models to explore and extract the underlying structures in data from multiple sources. One approach is to formulate this problem as a coupled matrix and tensor factorization problem. In this paper, we address the problem of solving coupled matrix and tensor factorizations when we have squared Euclidean distance as the loss function and introduce a first-order optimization algorithm called CMTF-OPT, which solves for all factor matrices in all data sets simultaneously. We have also extended our algorithm to data with missing entries by introducing CMTF-WOPT for the case where we have an incomplete higher-order tensor coupled with matrices. The algorithm can easily be extended to multiple incomplete data sets.

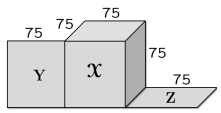
To the best of our knowledge, the algorithms proposed so far for fitting a coupled matrix and tensor factorization model have all been alternating algorithms. We have compared our CMTF-OPT algorithm with a traditional alter-

[3]Each norm is chosen as the absolute value of a number randomly chosen from $\mathcal{N}(0, 25)$ rounded to the nearest integer plus 1.

Table 1: Comparison of CMTF-OPT and CMTF-ALS for the cases where $\lambda_r = \alpha_r = ... = 1$, for $r = 1, 2, ...R$.

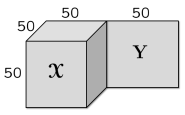


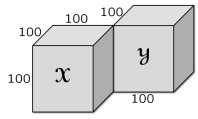


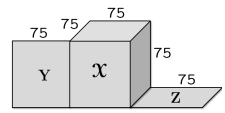


Noise	$\bar{R}$	ALG.	Success (%)	Mean FMS	p-val	Success (%)	Mean FMS	p-val	Success (%)	Mean FMS	p-val
$\eta = 0.10$	R	**OPT**	100.0	1.00	9.3e-1	96.7	0.97	1.0	100.0	1.00	3.3e-1
		ALS	100.0	1.00		96.7	0.97		96.7	0.96	
	$R+1$	**OPT**	96.7	0.97	3.3e-13	100.0	1.00	1.6e-8	96.7	0.97	3.8e-24
		ALS	3.3	0.29		6.7	0.71		0.0	0.06	
$\eta = 0.25$	R	**OPT**	100.0	0.99	4.5e-1	100.0	1.00	0.2e-1	96.7	0.96	3.3e-1
		ALS	100.0	0.99		100.0	1.00		100.0	0.99	
	$R+1$	**OPT**	100.0	0.99	2.6e-11	100.0	1.00	1.8e-7	100.0	0.99	7.0e-16
		ALS	6.7	0.34		16.7	0.70		10.0	0.14	
$\eta = 0.35$	R	**OPT**	100.0	0.99	5.8e-1	100.0	1.00	3.3e-1	100.0	0.99	3.6e-1
		ALS	100.0	0.99		96.7	0.97		100.0	0.99	
	$R+1$	**OPT**	90.0	0.92	1.8e-5	100.0	1.00	1.0e-7	86.7	0.88	2.9e-5
		ALS	33.3	0.55		13.3	0.71		36.7	0.39	

Table 2: Comparison of CMTF-OPT and CMTF-ALS for the cases where $\lambda_r, \alpha_r, ... \geq 1$ for $r = 1, 2, ...R$.







Noise	$\bar{R}$	ALG.	Success) (%)	Mean FMS	p-val	Success (%)	Mean FMS	p-val	Success (%)	Mean FMS	p-val
$\eta = 0.10$	R	**OPT**	96.7	0.96	3.3e-1	100.0	1.00	9.1e-1	96.7	0.96	3.3e-1
		ALS	100.0	0.99		100.0	1.00		90.0	0.90	
	$R+1$	**OPT**	90.0	0.96	1.5e-13	100.0	1.00	6.8e-9	83.3	0.89	1.6e-11
		ALS	3.3	0.24		13.3	0.52		10.0	0.13	
$\eta = 0.25$	R	**OPT**	76.7	0.97	3.3e-1	96.7	0.97	1.0	83.3	0.97	0.4e-1
		ALS	73.3	0.94		96.7	0.97		70.0	0.84	
	$R+1$	**OPT**	86.7	0.97	6.4e-9	100.0	1.00	7.8e-5	76.7	0.90	1.4e-8
		ALS	13.3	0.40		46.7	0.72		10.0	0.23	
$\eta = 0.35$	R	**OPT**	60.0	0.95	3.6e-1	100.0	1.00	3.3e-1	53.3	0.92	7.4e-1
		ALS	56.7	0.95		96.7	0.97		53.3	0.92	
	$R+1$	**OPT**	46.7	0.87	1.7e-9	100.0	1.00	4.0e-6	50.0	0.83	4.9e-7
		ALS	6.7	0.35		40.0	0.62		10.0	0.30	

nating least squares approach and the numerical results show that all-at-once optimization is more robust to overfactoring as it has also been the case for fitting tensor models [2, 32].

Note that our current formulation of coupled matrix and tensor factorization has one drawback, which is encountered when we have a data set whose factor matrices are shared by all other data sets. For instance, we may have a tensor $\mathcal{X}$, where $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ and two matrices $\mathbf{Y} = \mathbf{A}\mathbf{V}^\mathsf{T} = \sum_{r=1}^{R} \alpha_r \mathbf{a}_r \circ \mathbf{v}_r$ and $\mathbf{Z} = \mathbf{B}\mathbf{V}^\mathsf{T} = \sum_{r=1}^{R} \beta_r \mathbf{b}_r \circ \mathbf{v}_r$. If we formulate a coupled factorization for these data sets as $f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{V}) = \| \mathcal{X} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \|^2 + \left\| \mathbf{Y} - \mathbf{A}\mathbf{V}^\mathsf{T} \right\|^2 + \left\| \mathbf{Z} - \mathbf{B}\mathbf{V}^\mathsf{T} \right\|^2$, we do not take into account the cases for $\lambda_r \neq \alpha_r \neq \beta_r$. For such scenarios, scaling ambiguities should be taken into consideration by introducing a set of parameters for the scalars in the formulation.

Another issue we have not addressed in this paper is how to weigh different parts of the objective function modeling different data sets as discussed in [35]. This is an area of future research where a Bayesian framework for coupled matrix and tensor factorizations may be a promising approach. As another area of future research, we plan to extend our approach to different loss functions in order to be able to deal with different types of noise and incorporate nonnegativity constraints on the factors as nonnegativity often improves the interpretability of the model.

7. ACKNOWLEDGMENTS

This work was supported in part by the Laboratory Directed Research and Development program at Sandia National Laboratories. Sandia National Laboratories is a multi-

program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

8. REFERENCES

[1] E. Acar, D. Dunlavy, T. Kolda, and M. Morup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106:41–56, 2011.

[2] E. Acar, D. M. Dunlavy, and T. G. Kolda. A scalable optimization approach for fitting canonical tensor decompositions. *J. Chemometrics*, 25:67–86, 2011.

[3] E. Acar and B. Yener. Unsupervised multiway data analysis: A literature survey. *IEEE Trans. Knowledge and Data Engineering*, 21(1):6–20, 2009.

[4] O. Alter, P. O. Brown, and D. Botstein. Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. *PNAS*, 100(6):3351–3356, 2003.

[5] L. Badea. Combining gene expression and transcription factor regulation data using simultaneous nonnegative matrix factorization. In *Proc. BIOCOMP-2007*, pages 127–131, 2007.

[6] L. Badea. Extracting gene expression profiles common to colon and pancreatic adenocarcinoma using simultaneous nonnegative matrix factorization. In *Pacific Symposium on Biocomputing*, pages 267–278, 2008.

[7] A. Banerjee, S. Basu, and S. Merugu. Multi-way clustering on relation graphs. In *SDM'07*, pages 145–156, 2007.

[8] A. M. Buchanan and A. W. Fitzgibbon. Damped Newton algorithms for matrix factorization with missing data. In *CVPR'05*, pages 316–322, 2005.

[9] E. J. Candes and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Information Theory*, 56:2053–2080, 2009.

[10] J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35:283–319, 1970.

[11] S. Doclo and M. Moonen. Gsvd-based optimal filtering for single and multimicrophone speech enhancement. *IEEE Transactions on Signal Processing*, 50(9):2230–2244, 2002.

[12] D. M. Dunlavy, T. G. Kolda, and E. Acar. Poblano v1.0: A Matlab toolbox for gradient-based optimization. Technical Report SAND2010-1422, Sandia National Laboratories, Mar. 2010.

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

[14] J. Håstad. Tensor rank is NP-complete. *J. Algorithms*, 11(4):644–654, 1990.

[15] F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *J. Mathematics and Physics*, 6(1):164–189, 1927.

[16] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

[17] H. Kargupta, W. Huang, K. Sivakumar, B.-H. Park, and S. Wang. Collective principal component analysis from distributed, heterogeneous data. In *PKDD'00*, pages 452–457, 2000.

[18] J. R. Kettenring. *Canonical Analysis of Several Sets of Variables*. PhD thesis, University of North Carolina at Chapel Hill, 1969.

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

[20] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[21] J. Levin. Simultaneous factor analysis of several gramian matrices. *Psychometrika*, 31:413–419, 1966.

[22] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. Metafac: community discovery via relational hypergraph factorization. In *KDD'09*, pages 527–536, 2009.

[23] B. Long, X. Wu, Z. M. Zhang, and P. S. Yu. Unsupervised learning on k-partite graphs. In *KDD'06*, pages 317–326, 2006.

[24] B. Long, Z. M. Zhang, X. Wu, and P. S. Yu. Spectral clustering for multi-type relational data. In *ICML'06*, pages 585–592, 2006.

[25] I. V. Mechelen and A. K. Smilde. A generic linked-mode decomposition model for data fusion. *Chemometrics and Intelligent Laboratory Systems*, 104(1):83–94, 2010.

[26] J. J. Moré and D. J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Mathematical Software*, 20(3):286–307, 1994.

[27] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2006.

[28] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *KDD'08*, 2008.

[29] A. Smilde, R. Bro, and P. Geladi. *Multi-Way Analysis: Applications in the Chemical Sciences*. Wiley, West Sussex, England, 2004.

[30] A. Smilde, J. A. Westerhuis, and R. Boque. Multiway multiblock component and covariates regression models. *J. Chemometrics*, 14:301–331, 2000.

[31] R. S. Thorpe. Multiple group principal component analysis and population differentiation. *J. Zoology*, 216(1):37–40, 1988.

[32] G. Tomasi and R. Bro. A comparison of algorithms for fitting the PARAFAC model. *Computational Statistics and Data Analysis*, 50(7):1700–1734, 2006.

[33] Z. Wang, Y. Wang, and H. Wu. Tags meet ratings: Improving collaborative filtering with tag-based neighborhood method. In *IUI'10: Workshop on Social Recommender Systems*, 2010.

[34] J. A. Westerhuis, T. Kourti, and J. F. Macgregor. Analysis of multiblock and hierarchical PCA and PLS models. *J. Chemometrics*, 12:301–321, 1998.

[35] T. Wilderjans, E. Ceulemans, and I. V. Mechelen. Simultaneous analysis of coupled data blocks differing in size: A comparison of two weighting schemes. *Computational Statistics and Data Analysis*, 53:1086–1098, 2009.

[36] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang. Collaborative filtering meets mobile recommendation:

A user-centered approach. In *AAAI'10*, pages 236–241, 2010.

[37] A. Ziehe, P. Laskov, G. Nolte, and K.-R. Müller. A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *Journal of Machine Learning Research*, 5:777–800, 2004.

APPENDIX

Here we briefly describe the data generation for Example 1 in §1. Tensor $\mathcal{X}$ and matrix $\mathbf{Y}$ coupled in the first mode are constructed as follows:

- **Step 1:** We form factor matrices $\mathbf{A}_1 \in \mathbb{R}^{I \times R}$ and $\mathbf{A}_2 \in \mathbb{R}^{I \times R}$, where $R = 2$, such that in the first column of $\mathbf{A}_1$, entries corresponding to the members of G_1 and G_2 are assigned 1 (plus noise) while entries corresponding to the members of G_3 and G_4 are assigned -1 (plus noise). It is the vice versa for the second column; in other words, G_3 and G_4 members have 1 + noise values. The columns of $\mathbf{A}_2$ are generated similarly, except that in this case G_1 and G_3 form one cluster while G_2 and G_4 form another.
- **Step 2:** Factor matrices $\mathbf{B} \in \mathbb{R}^{J \times R}, \mathbf{C} \in \mathbb{R}^{K \times R}$ and $\mathbf{V} \in \mathbb{R}^{M \times R}$ are generated using random entries following standard normal distribution.
- **Step 3:** All columns of factor matrices are normalized to unit norm. $\mathcal{X}$ and $\mathbf{Y}$ are constructed as $\mathcal{X} = [\![\mathbf{A}_1, \mathbf{B}, \mathbf{C}]\!]$ and $\mathbf{Y} = \mathbf{A}_2\mathbf{V}^\mathsf{T}$.