Tendi: Tensor Disaggregation from Multiple Coarse Views

Faisal M. Almutairi[1], Charilaos I. Kanatsoulis[1], and Nicholas D. Sidiropoulos[2]

[1] ECE dept., University of Minnesota, Minneapolis, MN 55455 USA
(almut012, kanat003)@umn.edu
[2] ECE dept., University of Virginia, Charlottesville, VA 22903, USA
nikos@virginia.edu

Abstract. Multidimensional data appear in various interesting applications, e.g., sales data indexed by stores, items, and time. Oftentimes, data are observed aggregated over multiple data atoms, thus exhibit low resolution. Temporal aggregation is most common, but many datasets are also aggregated over other attributes. Multidimensional data, in particular, are sometimes available in multiple coarse views, aggregated across different dimensions – especially when sourced by different agencies. For instance, item sales can be aggregated temporally, and over groups of stores based on their location or affiliation. However, data in finer granularity significantly benefit forecasting and data analytics, prompting increasing interest in data disaggregation methods. In this paper, we propose Tendi, a principled model that efficiently disaggregates multidimensional (tensor) data from multiple views, aggregated over different dimensions. Tendi employs coupled tensor factorization to fuse the multiple views and provide recovery guarantees under realistic conditions. We also propose a variant of Tendi, called TendiB, which performs the disaggregation task without any knowledge of the aggregation mechanism. Experiments on real data from different domains demonstrate the high effectiveness of the proposed methods.

Keywords: Data disaggregation · tensor decomposition · multiview data

1 Introduction

Low-resolution data, aggregated over multiple data indices, are found in the databases of diverse applications, e.g., economics [8], health care [15], education [5], and smart grid systems [6], to name a few. The most common type of aggregation is *temporal aggregation*, for example, the GDP quarterly national accounts are aggregated over months. Aggregation over other dimensions is also common, such as geographically (e.g., population of New York by county) or according to a defined affiliation (e.g., number of students by majors). The latter is known in economics literature as *contemporaneous aggregation*. The different types of aggregation are often combined. For instance, the number of foreigners who visited different US states in 2019 can be aggregated in time, location (states), and affiliation (nationality). Aggregated data offer data summarization, which serves multiple purposes, including scalability, communication cost, and privacy. On the other hand, a plethora of data mining and machine learning tasks strive for

data in high-resolution (disaggregated). Analysis results can differ substantially when using aggregated versus disaggregated data in many application domains, such as economics [8], education [5], and supply chains [20]. This has motivated numerous works in developing algorithms for data disaggregation.

The task of data disaggregation, in general, boils down to finding a solution to a system of linear equations $\mathbf{U}\mathbf{x} = \mathbf{y}$, where $\mathbf{y}$ is the vector of aggregated observations, $\mathbf{x}$ is the target disaggregated series, and $\mathbf{U}$ is the aggregation matrix that maps the target series to the aggregated measurements. In practical settings, the linear system is under-determined as the number of observations is often significantly smaller than the length of the target series, resulting in an ill-posed problem. In order to tackle the problem, disaggregation techniques exploit side information or domain knowledge [14, 2], in their attempt to over-determine the problem and enhance the disaggregation accuracy. Some common prior models, imposed on the target high-resolution data, involve smoothness, periodicity [14], non-negativity, and sparsity over a given dictionary [2]. The main issue with these approaches is that they impose application-specific constraints and therefore they cannot generalize to different disaggregation tasks in a straightforward manner. Moreover, it is unclear whether the assumed models are identifiable (i.e., an optimal solution of the model is not guaranteed to be the true disaggregated data), especially when the solution does not *exactly* follow the imposed constraints. Note that, identifiability is important, in the sense of assuring correct recovery under certain reasonable conditions. In our present context, identifiability has not received the attention it deserves, likely because guaranteed recovery is considered mission impossible under realistic conditions

An interesting special case of disaggregation arises when data are aggregated over more that one dimension. This is a popular research problem in the area of business and economics going back to the 70's [4]. In this case, temporal *and* contemporaneous aggregated views of the data are available. For instance, we are interested in estimating the quarterly Gross Regional Product (GRP) values for regions of a country, given: 1) the annual GRP per region (temporal aggregates), and 2) the GDP quarterly national accounts (contemporaneous aggregates) [16]. Another notable example appears in healthcare, where data are collected by national, regional, and local government agencies, health and scientific organizations, insurance companies and other entities, and are often aggregated in many dimensions (e.g., temporally, geographically, or group of hospitals), often to preserve privacy [15]—see Section 2.2 for another example. Algorithms have been developed to integrate the multiple aggregates in the disaggregation process [4, 16]. The majority of them leverage linear regression models with priors and require additional information to perform the disaggregation task.

In this paper we study the multiview dissagregation task using a tensor decomposition approach, which provably converts the ill-posed problem to an identifiable one. Our work is inspired by the following question: *Is the disaggregation task possible when the data are: 1) multidimensional, and 2) observed by different agencies via diverse aggregation mechanisms?* This is a well motivated problem due to the ubiquitous presence of data with multiple dimensions (three

or more), also known as tensors, in a large number of applications. It is also very common that aggregation happens in more than one dimensions as in the previously explained examples. The informal definition of the problem is:

Informal Problem 1 (Multidimensional Disaggregation)
- **Given:** *two (or more) observations of a multidimensional dataset, each representing a view of the data aggregated in one (or more) dimension (e.g., temporal and contemporaneous aggregates).*
- **Recover:** *the data in high-resolution (disaggregated) in all the dimensions.*

We propose TENDI: a principled model for fusing the multiple aggregates of multidimensional data. The proposed approach represents the target high-resolution data as a *tensor*, and models them using the *canonical polyadic decomposition* (CPD) to reduce the number of unknowns, while capturing correlations and higher-order statistical dependencies across dimensions. TENDI employs a coupled CPD approach and estimates the low-rank factors of the target data, to perform the disaggregation task. This way the originally ill-posed disaggregation problem is transformed to an over-determined one, by leveraging the uniqueness properties of the CPD. TENDI can disaggregate under the challenging scenario where the views are doubly aggregated, i.e., a view is aggregated in two dimensions. We also propose an algorithm (called TENDIB) that handles the disaggregation task in cases where the aggregation pattern is unknown. As a result, the proposed framework not only provides a disaggregation algorithm, but also gives insights that can be potentially exploited in creating accurately retrievable data summaries for database applications. Along the same lines, our work provides insights on when aggregation does not preserve anonymity. With the aid of another view of aggregated data, estimating the individual-level accurately is possible as we show in this work, even without knowing the aggregation pattern. This leads to privacy violation if data are aggregated to preserve anonymity. Experiments on real data from different applications show that TENDI is very effective and significantly improves the accuracy of the baselines. In summary, the contributions of our work are as follows:

- **Formulation**: we formally define the multidimensional data disaggregation task from multiple views, aggregated across different dimensions, and provide an efficient algorithm.
- **Identifiability:** the considered model can provably transform the original ill-posed disaggregation problem to an identifiable one.
- **Effectiveness:** TENDI recovers real data accurately and reduces the disaggregation error of the best baseline by up to 48%.
- **Blind disaggregation:** the proposed model works very effectively, even when the aggregation mechanism is unknown (TENDIB).

2 Preliminaries & Problem Statement

Notation: $\mathbf{x}$, $\mathbf{X}$, $\boldsymbol{\mathcal{X}}$ denote a vector, a matrix, and a tensor, respectively, $\mathbf{X}^{(n)}$ is mode-n matricization of $\boldsymbol{\mathcal{X}}$, $\|.\|_F$ is the Frobenius norm, and $[\![.]\!]$ denotes the Kruskal operator, e.g., $\boldsymbol{\mathcal{X}} \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$. $\mathbf{X}^T$ is the Transpose of $\mathbf{X}$, and $\text{vec}(\cdot)$ is the vectorization operator for matrix $\mathbf{X}$ or tensor $\boldsymbol{\mathcal{X}}$. Finally, $\circ$, $\odot$, and $\circledast$ denote the outer, Khatri-Rao, and Hadamard (element-wise) products, respectively.

2.1 Tensor Preliminaries

Tensors are multidimensional arrays indexed by three or more indices, $(i, j, k, ...)$. A third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ consists of three *modes*: columns $\boldsymbol{\mathcal{X}}(:, j, k)$, rows $\boldsymbol{\mathcal{X}}(i, :, k)$, and fibers $\boldsymbol{\mathcal{X}}(i, j, :)$. Moreover, $\boldsymbol{\mathcal{X}}(i, :, :)$, $\boldsymbol{\mathcal{X}}(:, j, :)$, and $\boldsymbol{\mathcal{X}}(:, :, k)$ denote the i^{th} horizontal, j^{th} lateral, and k^{th} frontal slabs/slices of $\boldsymbol{\mathcal{X}}$, respectively—refer to [13, 17] for more background on tensors.

Tensor decomposition (CPD): A rank-one third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ results from the outer product of three vectors, i.e., $\boldsymbol{\mathcal{X}} = (\mathbf{a} \circ \mathbf{b} \circ \mathbf{c})$, where $\mathbf{a} \in \mathbb{R}^I$, $\mathbf{b} \in \mathbb{R}^J$, and $\mathbf{c} \in \mathbb{R}^K$. The Canonical Polyadic Decomposition (CPD) decomposes a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ into a sum of R rank-one tensors, i.e., $\boldsymbol{\mathcal{X}} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$, where the minimal $R \in \mathbb{N}$ for which the approximation is exact is the *rank* of $\boldsymbol{\mathcal{X}}$, $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, and $\mathbf{c}_r \in \mathbb{R}^K$. The CPD can be stated in terms of the *factor matrices* as $\boldsymbol{\mathcal{X}} \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, where $\mathbf{A} = [\mathbf{a}_1 \dots \mathbf{a}_R] \in \mathbb{R}^{I \times R}$, $\mathbf{B} = [\mathbf{b}_1 \dots \mathbf{b}_R] \in \mathbb{R}^{J \times R}$ and $\mathbf{C} = [\mathbf{c}_1 \dots \mathbf{c}_R] \in \mathbb{R}^{K \times R}$. A striking property of CDP is that it is essentially unique (the rank-one components $\mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ are unique; or, equivalently, $\mathbf{A}, \mathbf{B}, \mathbf{C}$ can be identified up to common column permutation and scaling) under mild conditions [3]. The CPD can also be expressed using the matricized (unfolded) tensors as $\mathbf{X}^{(1)} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T$, $\mathbf{X}^{(2)} = \mathbf{B}(\mathbf{C} \odot \mathbf{A})^T$, and $\mathbf{X}^{(3)} = \mathbf{C}(\mathbf{B} \odot \mathbf{A})^T$, where $\mathbf{X}^{(1)} \in \mathbb{R}^{I \times JK}$, $\mathbf{X}^{(2)} \in \mathbb{R}^{J \times IK}$, and $\mathbf{X}^{(3)} \in \mathbb{R}^{K \times IJ}$ are mode-1, mode-2, and mode-3 unfolding of $\boldsymbol{\mathcal{X}}$, respectively.

Mode product: is the multiplication of a matrix by a tensor in one particular mode, e.g., mode-1 product of matrix $\mathbf{U} \in \mathbb{R}^{I_u \times I}$ and tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ corresponds to multiplying every column $\boldsymbol{\mathcal{X}}(:, j, k)$ of the tensor by $\mathbf{U}$. Similarly, mode-2 (mode-3) product corresponds to multiplying every row (fiber) of $\boldsymbol{\mathcal{X}}$ by a matrix. Mode products can also be expressed in terms of unfolded tensors. Multiplying a matrix $\mathbf{U}$ in the n^{th} mode can be denoted as: $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_n \mathbf{U} \iff \mathbf{Y}^{(n)} = \mathbf{U}\mathbf{X}^{(n)}$, where "$\times_n$" is the product over the n^{th} mode—see Fig. 1 for an illustration. An important observation is that mode products can be absorbed in the CPD of the tensor, i.e., in Fig. 1, if $\boldsymbol{\mathcal{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, then $\boldsymbol{\mathcal{Y}} = [\![\mathbf{UA}, \mathbf{VB}, \mathbf{WC}]\!]$

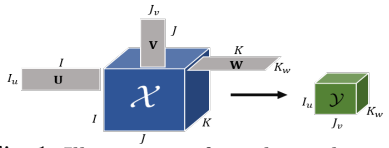


Fig. 1: Illustration of mode products

2.2 Disaggregation Problem

Given a set of low-resolution observations $\mathbf{y} \in \mathbb{R}^{I_u}$ (e.g., monthly) about a time series $\mathbf{x} \in \mathbb{R}^I$, the goal of the time series disaggregation problem is to estimate the series $\mathbf{x}$ in a higher resolution (e.g., weekly). This can be cast as a linear inverse problem $\mathbf{y} = \mathbf{U}\mathbf{x}$, where $\mathbf{U} \in \mathbb{R}^{I_u \times I}$ is a 'fat' *aggregation matrix* that maps the observations in $\mathbf{y}$ with the variables in $\mathbf{x}$. In this work, we consider the case where the target high-resolution data are multidimensional (tensor). The different dimensions represent the physical dimensions of the data, e.g., time stamps, locations, etc. For the sake of simplicity of exposition, we focus on three-dimensional data in our formulation and algorithm. However, the proposed method can handle more general cases with data of higher order. Specifically, let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ be the target high-resolution third-order tensor. In the considered problem, we are given two sets of observations, each aggregated

over one or more different dimension(s), which is common when data are reported by different agencies, resulting in multiple views of the same information. The key insight is that the given aggregates can be modeled as mode product(s) of $\boldsymbol{\mathcal{X}}$ by an aggregation matrix in a particular mode(s). To see this, consider tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{4\times2\times2}$, a simple example of a set of observations aggregated over the first mode can be expressed as

$$\underbrace{\begin{bmatrix}1&1&0&0\\0&0&1&1\end{bmatrix}}_{\mathbf{U}\in\mathbb{R}^{2\times4}} \times \underbrace{\begin{bmatrix}x_{111}&x_{121}&x_{112}&x_{122}\\x_{211}&x_{221}&x_{212}&x_{222}\\x_{311}&x_{321}&x_{312}&x_{322}\\x_{411}&x_{421}&x_{412}&x_{422}\end{bmatrix}}_{\mathbf{X}^{(1)}\in\mathbb{R}^{4\times(2\times2)}} = \underbrace{\begin{bmatrix}y_{111}&y_{121}&y_{112}&y_{122}\\y_{211}&y_{221}&y_{212}&y_{222}\end{bmatrix}}_{\mathbf{Y}^{(1)}\in\mathbb{R}^{2\times(2\times2)}} \tag{1}$$

The same idea applies when the aggregation is over the second (third) mode using mode-2 (mode-3) product. The major challenge in data disaggregation is that the number of available aggregated observations is much smaller than the number of variables, resulting in an under-determined ill-posed problem. This is the case even when more than one set of aggregates are available.

Before defining the problem formally, we explain the concept with an example of retail sales. There are two sources of data used to forecast future demand in retail sales: 1) store-level data, commonly aggregated in time (temporal aggregate $\boldsymbol{\mathcal{Y}}_t$); and 2) historical *orders* by the retailers' Distribution Centers (DC orders), aggregated over their multiple stores (contemporaneous aggregate $\boldsymbol{\mathcal{Y}}_c$). Note that both store-level and DC orders data are used for demand forecasting, and especially store-level data are vital in predicting future orders [20]. Hence, many retailers share data with their suppliers to assist in the forecasting task and avoid shortage or excess in inventory [9]. In a more restricted scenario, the second source collects sales of each category of items rather than each item individually. The question that arises is whether we can fuse these sources to reconstruct high-resolution data in stores, items, and time dimensions. Formally, we are interested in:

*Problem 1 (***Tensor Disaggregation).**

- **Given** two aggregated views of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$: $\boldsymbol{\mathcal{Y}}_t \in \mathbb{R}^{I\times J\times K_w}$, and $\boldsymbol{\mathcal{Y}}_c \in \mathbb{R}^{I_u\times J\times K}$ (or $\boldsymbol{\mathcal{Y}}_c \in \mathbb{R}^{I_u\times J_v\times K}$), where $I_u < I$, $J_v < J$, and $K_w < K$.
- **Recover** the original disaggregated tensor data $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$.

We tackle this problem using a coupled low-rank factorization model as we explain next. Coupled factorization techniques are commonly used to fuse information when data share common dimension(s) for different tasks, e.g., link prediction [7], demand forecasting [21], context-aware recommendation [1], medical imaging [10], and remote sensing [11]. Closest to our work is the approach in [11], which employs a coupled CPD to fuse a hyperspectral image with a multispectral image, to produce a high spatial and spectral resolution image. To our knowledge, this work is the first to propose a coupled tensor factorization to tackle data disaggregation applications.

3 Proposed Method: Tendi

TENDI builds upon two basic principles. The first is that the target tensor, $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$, admits a CPD model ($\boldsymbol{\mathcal{X}} \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$). The second notes that

the available aggregates, $\mathcal{Y}_t$ and $\mathcal{Y}_c$, are resulting from the mode product of an aggregation matrix (matrices) by $\mathcal{X}$ in a particular mode(s). In particular, $\mathcal{Y}_t = \mathcal{X} \times_3 \mathbf{W}$, and $\mathcal{Y}_c = \mathcal{X} \times_1 \mathbf{U}$ (or $\mathcal{Y}_c = \mathcal{X} \times_1 \mathbf{U} \times_2 \mathbf{V}$), where $\mathbf{W} \in \mathbb{R}^{K_w \times K}$, $\mathbf{U} \in \mathbb{R}^{I_u \times I}$, and $\mathbf{V} \in \mathbb{R}^{J_v \times J}$ are aggregation matrices with $K_w < K$, $I_u < I$, and $J_v < J$. As a result, the aggregated views admit CPD models: $\mathcal{Y}_t \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{WC}]\!]$ and $\mathcal{Y}_c \approx [\![\mathbf{UA}, \mathbf{VB}, \mathbf{C}]\!]$.

TENDI learns the factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ by applying a coupled CPD model on the available aggregates—Fig. 2 illustrates the high level picture of TENDI. Specifically, we propose the following formulation:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \quad \mathcal{L}(\mathbf{A}, \mathbf{B}, \mathbf{C}) := \|\mathcal{Y}_t - [\![\mathbf{A}, \mathbf{B}, \mathbf{WC}]\!]\|_F^2 + \|\mathcal{Y}_c - [\![\mathbf{UA}, \mathbf{VB}, \mathbf{C}]\!]\|_F^2 \tag{2}$$

Note that additional aggregated views can be handled in a similar fashion.

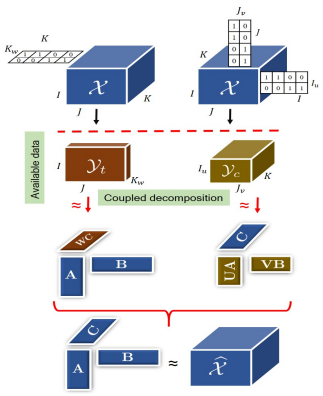


Fig. 2: Overview of TENDI.

3.1 Algorithm

Problem (2) is non-convex, and NP-hard in general. To tackle it we employ a *block coordinate descent* (BCD) approach and update the three factors in an alternating fashion as summarized in Algorithm 1. The gradient of the loss function $\mathcal{L}$ w.r.t. $\mathbf{A}$ is

$$\begin{aligned}\nabla_{\mathbf{A}}\mathcal{L} = {} & 2\mathbf{A}((\mathbf{WC}) \odot \mathbf{B})^T((\mathbf{WC}) \odot \mathbf{B}) + 2\mathbf{U}^T\mathbf{UA}(\mathbf{C} \odot (\mathbf{VB}))^T(\mathbf{C} \odot (\mathbf{VB})) \\ & - 2\mathbf{Y}_t^{(1)}(\mathbf{WC} \odot \mathbf{B}) - 2\mathbf{U}^T\mathbf{Y}_c^{(1)}(\mathbf{C} \odot (\mathbf{VB}))\end{aligned} \tag{3}$$

Using the properties of the Khatri-Rao product, the space and time computational complexity of the products $(\mathbf{C} \odot (\mathbf{VB}))^T(\mathbf{C} \odot (\mathbf{VB}))$ can be reduced using the following element-wise Hadamard product $\left(\mathbf{C}^T\mathbf{C}\right) \circledast \left(\mathbf{B}^T\mathbf{V}^T\mathbf{VB}\right)$ (similarly for $((\mathbf{WC}) \odot \mathbf{B})^T((\mathbf{WC}) \odot \mathbf{B})$) [19]. The updates of the factors $\mathbf{B}$ and $\mathbf{C}$ can be derived similarly using mode-2 and mode-3 unfolding of the tensors, respectively. The step size parameters α, β, and γ in Algorithm 1 are chosen by the *exact line search* method—see steps 1,3, and 5 in Algorithm 1.

The initialization step in Algorithm 1 is crucial to the disaggregation accuracy. Thus, we propose to initialize as follows: if $\mathcal{Y}_c$ is aggregated in two modes, then we initialize by:

$$\mathbf{A}^{(0)}, \mathbf{B}^{(0)}, \widetilde{\mathbf{C}} \leftarrow \texttt{CPD}(\mathcal{Y}_t), \ \widetilde{\mathbf{A}} = \mathbf{UA}^{(0)}, \ \widetilde{\mathbf{B}} = \mathbf{VB}^{(0)}, \ \mathbf{C}^{(0)} \leftarrow \mathbf{Y}_c^{(3)} = \mathbf{C}^{(0)}(\widetilde{\mathbf{B}} \odot \widetilde{\mathbf{A}})^T \tag{4}$$

if $\mathcal{Y}_c$ is aggregated only in one mode, i.e., $\mathbf{V} = \mathbf{I}$, then $\mathbf{B}$ is common in the two aggregated tensors and we can use the CPD of either to get two "disaggregated" factors. In this case, if $I > K$, then we initialize with (4), otherwise, we use:

$$\widetilde{\mathbf{A}}, \mathbf{B}^{(0)}, \mathbf{C}^{(0)} \leftarrow \texttt{CPD}(\mathcal{Y}_c), \quad \widetilde{\mathbf{C}} = \mathbf{WC}^{(0)}, \quad \mathbf{A}^{(0)} \leftarrow \mathbf{Y}_t^{(3)} = \mathbf{A}^{(0)}(\widetilde{\mathbf{C}} \odot \mathbf{B}^{(0)})^T \tag{5}$$

This way we have obtained an initial guess for all the factors. We use the Matlab-based package `Tensorlab` to compute the CPD in the initialization step.

The computational complexity of each step in Algorithm 1 boils down to matrix multiplications that are dominated by $\mathcal{O}\big((I_u J_v K + IJK_w)R\big)$. Since R is very small relative to the size of the tensors with many real data, the complexity is linear in the number of observations in $\mathcal{Y}_t$ and $\mathcal{Y}_c$.

3.2 TendiB: Tendi with Blind Disaggregation

In most practical applications, the aggregation details are known. However, there exists cases with limited or no information on how data are aggregated (i.e., $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ are unknown). This happens in privacy sensitive domains such as healthcare [15], where hospital records are aggregated to protect the privacy of patients. For such cases, we propose TENDIB (TENDI with Blind disaggregation) to get the factors of the disaggregated tensor ($\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$):

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C},\widetilde{\mathbf{A}},\widetilde{\mathbf{C}}} \quad \mathcal{F} := \|\mathcal{Y}_t - [\![\mathbf{A},\mathbf{B},\widetilde{\mathbf{C}}]\!]\|_F^2 + \|\mathcal{Y}_c - [\![\widetilde{\mathbf{A}},\mathbf{B},\mathbf{C}]\!]\|_F^2 + \mu\|\mathbf{1}^T\widetilde{\mathbf{C}} - \mathbf{1}^T\mathbf{C}\|_2^2 \tag{6}$$

Where $\widetilde{\mathbf{A}} = \mathbf{U}\mathbf{A}$, and $\widetilde{\mathbf{C}} = \mathbf{W}\mathbf{C}$ are treated as separate variables since we do not know $\mathbf{U}$ and $\mathbf{W}$. This results in a more challenging problem than (2) as the number of variables is increased, with the same number of equations. Another challenge is that there is a scaling ambiguity between the factors of the two tensors, if we omit the third term in (6). To overcome this, we observe that temporal aggregation $\mathbf{W}$ in most aggregated data is non-overlapping and includes all the time ticks[3]. This means that the respective column sums of $\mathbf{C}$ and $\widetilde{\mathbf{C}}$ should be equal. We exploit this observation by adding the last term in (6), thereby reconciling the scaling ambiguity.

In order to solve (6), we adopt an Alternating Optimization (AO) procedure described in Algorithm 2. The updates of $\mathbf{A}$, $\widetilde{\mathbf{A}}$, and $\mathbf{B}$ are solving overdetermined linear systems, and those for $\mathbf{C}$, $\widetilde{\mathbf{C}}$ boil down to solving a *Sylvester equation*. The Sylvester equation is a special form of a linear system of equations, which can be handled efficiently [12]. To initialize the variables in Algorithm 2, we compute the ($\texttt{CPD}(\mathcal{Y}_c)$) to get $\widetilde{\mathbf{A}}^{(0)}$, $\mathbf{B}^{(0)}$, and $\mathbf{C}^{(0)}$. To get an initial estimate of $\widetilde{\mathbf{C}}$, we exploit the fact that the temporal aggregates are the summation over consecutive time stamps in most real data. As such, we sum every consecutive $w = \text{round}(\frac{K}{K_W})$ rows in $\mathbf{C}^{(0)}$ (In the experiments, we make sure that the true and estimated temporal aggregation do not align).

Algorithm 1 : TENDI (2)

input: $\mathcal{Y}_t$, $\mathcal{Y}_c$, $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$, R
output: $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$
Initialize: $\mathbf{A}^{(0)}, \mathbf{B}^{(0)}, \mathbf{C}^{(0)}$ by (4) or (5)
Repeat
1. $\alpha \leftarrow \text{argmin}_\alpha \ \mathcal{L}(\mathbf{A} - \alpha\nabla_{\mathbf{A}})$
2. $\mathbf{A} = \mathbf{A} - \alpha\nabla_{\mathbf{A}}\mathcal{L}$
3. $\beta \leftarrow \text{argmin}_\beta \ \mathcal{L}(\mathbf{A} - \beta\nabla_{\mathbf{A}})$
4. $\mathbf{B} = \mathbf{B} - \beta\nabla_{\mathbf{B}}\mathcal{L}$
5. $\gamma \leftarrow \text{argmin}_\gamma \ \mathcal{L}(\mathbf{A} - \gamma\nabla_{\mathbf{A}})$
6. $\mathbf{C} = \mathbf{C} - \gamma\nabla_{\mathbf{C}}\mathcal{L}$

Until convergence (max. #iteration)

Algorithm 2 : TENDIB (6)

input: $\mathcal{Y}_t$, $\mathcal{Y}_c$, R, $\mu \geq 0$
output: $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$
Initialize: $\widetilde{\mathbf{A}}^{(0)}, \mathbf{B}^{(0)}, \mathbf{C}^{(0)} \leftarrow \texttt{CPD}(\mathcal{Y}_c)$
$\widetilde{\mathbf{C}}^{(0)}(k_w,:) \leftarrow \sum_{k=w(k_w-1)+1}^{w\times k_w} \mathbf{C}^{(0)}(k,:)$
Repeat
- $\widetilde{\mathbf{A}} \leftarrow \text{argmin}_{\widetilde{\mathbf{A}}} \quad \mathcal{F}(\mathbf{A},\mathbf{B},\mathbf{C},\widetilde{\mathbf{A}},\widetilde{\mathbf{C}})$
- $\mathbf{C} \leftarrow \text{argmin}_{\mathbf{C}} \quad \mathcal{F}(\mathbf{A},\mathbf{B},\mathbf{C},\widetilde{\mathbf{A}},\widetilde{\mathbf{C}})$
- $\mathbf{A} \leftarrow \text{argmin}_{\mathbf{A}} \quad \mathcal{F}(\mathbf{A},\mathbf{B},\mathbf{C},\widetilde{\mathbf{A}},\widetilde{\mathbf{C}})$
- $\widetilde{\mathbf{C}} \leftarrow \text{argmin}_{\widetilde{\mathbf{C}}} \quad \mathcal{F}(\mathbf{A},\mathbf{B},\mathbf{C},\widetilde{\mathbf{A}},\widetilde{\mathbf{C}})$
- $\mathbf{B} \leftarrow \text{argmin}_{\mathbf{B}} \quad \mathcal{F}(\mathbf{A},\mathbf{B},\mathbf{C},\widetilde{\mathbf{A}},\widetilde{\mathbf{C}})$

Until convergence (max. #iteration)

3.3 Identifiability Analysis

As mentioned earlier, the disaggregation task is an inverse ill-posed problem. Modeling the data with CPD allows to provably transform the ill-posed disaggregation problem to an identifiable one. In other words, the optimal solution

[3] *Known*, e.g., 50% overlap can be treated similarly – as in this case every atom is counted twice.

of (2) and (6) are guaranteed to be unique, under mild conditions and identify the original high-resolution tensor almost surely. Formally identifiability is established in Proposition 1.

Proposition 1. *Let* $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$ *be the target high-resolution tensor data that admits a CPD* $\boldsymbol{\mathcal{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ *with rank R. Also let* $\boldsymbol{\mathcal{Y}}_t \in \mathbb{R}^{I\times J\times K_w} = \boldsymbol{\mathcal{X}} \times_3 \mathbf{W}$ *and* $\boldsymbol{\mathcal{Y}}_c \in \mathbb{R}^{I_u\times J_v\times K} = \boldsymbol{\mathcal{X}} \times_1 \mathbf{U} \times_2 \mathbf{V}$ *be the two aggregated observations. Assume that* $\mathbf{A}$*,* $\mathbf{B}$ *and* $\mathbf{C}$ *are drawn from some absolutely continuous distribution, and that* $(\mathbf{A}^\star, \mathbf{B}^\star, \mathbf{C}^\star)$ *is an optimal solutions to problem* (2) *or* (6)*. Then,* $\hat{\boldsymbol{\mathcal{X}}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ *disaggregates* $\boldsymbol{\mathcal{Y}}_t$*,* $\boldsymbol{\mathcal{Y}}_c$ *to* $\boldsymbol{\mathcal{X}}$ *almost surely if* $R \leq \frac{1}{16}\min\{IJ, IK_w, JK_w, 16I_uJ_v\}$.

The proof is relegated to a journal version of this work due to space limitation, and it leverages the uniqueness properties of the CPD. From our experiments, we observed the tested data approximately exhibit a low-rank structure and therefore our identifiability conditions are satisfied.

4 Experiments

4.1 Datasets

We evaluate TENDI using the following publicly online available datasets:

DFF: retail sales data from Dominick's Finer Foods (DFF), which used to be a grocery store chain in Chicago until it closed. DFF data were collected by the James M. Kilts Center, University of Chicago Booth School of Business. We create 2 ground-truth category-specific (stores × items × weeks) tensors $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$ containing the number of sold items of 50 different types of Cheese (CHE) and fabric softeners (FSF). We choose these two categories because they have different statistics, i.e., different sparsity and standard deviation (SD), to thoroughly examine the disaggregation performance. In addition, we form a (stores × items × weeks) tensor containing items from 10 different categories combined, 50 items from each (namely DFF in Table 1). DFF data contain the geographical locations of stores, which we use to aggregate stores into groups.

Crime: number of crime incidents in the City of Chicago from 2001 to present, marked with beats (police geographical areas), and codes indicating the crime types. We form a (locations (by beat) × crime types × months) tensor.

Walmart: weekly sales for a number of *departments* in 45 Walmart stores. A (stores × departments × weeks) tensor is created from this data. The information on square feet size of stores is available and we use it to aggregate the stores.

Weather: daily weather observations from 49 stations in Australia. These data include 17 different variables, e.g., min/max temperature, humidity, etc. We form a (station × variables × days) tensor of daily observations for one year.

The data, we aim to disaggregate, are created using the datasets summarized with their statistics in in Table 1 and represented by $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$. We examine the performance on two different scenarios: 1) **Scenario A**: we are given temporally aggregated tensor $\boldsymbol{\mathcal{Y}}_t = \boldsymbol{\mathcal{X}} \times_3 \mathbf{W}$ (i.e., aggregated in the third dimension), and contemporaneously aggregated tensor $\boldsymbol{\mathcal{Y}}_c = \boldsymbol{\mathcal{X}} \times_1 \mathbf{U}$ aggregated in the first mode (stores/locations dimension); and 2) **Scenario B**: where we observe $\boldsymbol{\mathcal{Y}}_t$ similar to scenario A, however, the contemporaneous aggregate is

aggregated in the first *and* second dimensions in this scenario (double aggregation), i.e., $\boldsymbol{\mathcal{Y}}_c = \boldsymbol{\mathcal{X}} \times_1 \mathbf{U} \times_2 \mathbf{V}$. The difficulty of the problem also depends on the *aggregation level*, i.e., the number of data points (e.g., weeks) in one sum. Fewer aggregated samples result in more challenging problems, and we test the performance using different aggregation levels.

4.2 Evaluation Baselines & Metrics

Table 1: Summary of datasets.

Dataset	Size	Max	Avg	SD	% (zeros)
CHE	$93 \times 50 \times 393$	18176	24.36	84.75	14.10
FSF	$93 \times 50 \times 397$	7168	4.62	17.13	46.13
DFF	$93 \times 500 \times 230$	17610	16.10	65.98	33.13
Crime	$304 \times 388 \times 221$	325	0.26	1.47	8.44
Walmart	$45 \times 99 \times 143$	6.93e+05	1.05e+04	1.99e+04	66.16
Weather	$49 \times 17 \times 365$	1038	10.23	95.65	93.30

We evaluate the performance of TENDI using the Normalized Disaggregation Error (NDE $= \|\boldsymbol{\mathcal{X}} - \hat{\boldsymbol{\mathcal{X}}}\|_F^2 / \|\boldsymbol{\mathcal{X}}\|_F^2$), where $\hat{\boldsymbol{\mathcal{X}}}$ is the estimated data. We compare the performance to state-of-art approaches in time series disaggregation literature as well as methods developed to fuse multiple views of multidimensional data, but for different tasks (CMTF baseline). To the best of our knowledge our work is the first to perform disaggregation on multidimensional data from multiple views.

Mean: assumes that the constituents data atoms (entries in $\boldsymbol{\mathcal{X}}$) have equal contribution in their aggregated samples. The final estimate of Mean is the average of the estimation from the temporal and contemporaneous aggregates.

LS: baseline is inspired by [16]. However, this work uses additional information that is not available in our context. Therefore, we find the minimum-norm solution to the least squares criterion on the linear relationship between vec($\boldsymbol{\mathcal{X}}$), and vec($\boldsymbol{\mathcal{Y}}_t$) and vec($\boldsymbol{\mathcal{Y}}_c$).

H-Fuse:[14] constrains the solution of LS baseline above to be smooth, i.e., it penalizes the large differences between adjacent time ticks.

HomeRun:[2] solves for vec($\boldsymbol{\mathcal{X}}$) in the frequency domain. Specifically, it searches for the vector $\mathbf{s}$ such that $\mathbf{s} = \mathbf{D}$vec($\boldsymbol{\mathcal{X}}$), where $\mathbf{D}$ is a matrix containing the Discrete Cosine Transform basis.

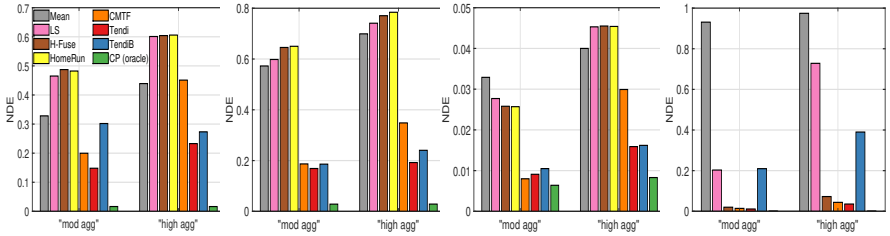
CMTF:[18] is coupled low-rank matrix factorization of the matricized tensors.

CP: fits a CPD model to the *ground-truth* tensor $\boldsymbol{\mathcal{X}}$ using `Tensorlab`. Then, $\hat{\boldsymbol{\mathcal{X}}}$ is reconstructed from the learned factors (lower bound on the NDE we can achieve).

4.3 Effectiveness

In the experiments, we set $\mu = 100$, and choose R for TENDI (and CP baseline) based on Proposition 1. For CMTF we perform a grid search and show results with the best R. We run 10 iterations of the CPD in the initialization step of Algorithm 1 and 2 using `Tensorlab`, then 10 iterations of TENDI (or TENDIB).

Results on Scenario A: We test this scenario with two aggregation levels on four datasets (CHE, FSF, Walmart, and Weather) as shown in Fig. 3. The aggregation levels with CHE and FSF data are: 1) weeks are aggregated into months in $\boldsymbol{\mathcal{Y}}_t$, whereas 93 stores are divided into 16 areas in $\boldsymbol{\mathcal{Y}}_c$ with the moderate aggregation ("mod agg") level; and 2) quarterly samples (every 12 weeks) in $\boldsymbol{\mathcal{Y}}_t$, and stores are divided into only 9 areas with the high aggregation ("high agg") level. We conclude from the results of CHE and FSF that TENDI is more robust compared to all baselines when aggregation is aggressive (only few samples are available). For instance, with "high agg", the number of available samples in $\boldsymbol{\mathcal{Y}}_t$



(a) CHE data (b) FSF data (c) Walmart data (d) Weather data

Fig. 3: **Tendi works well with extreme aggregation on different data**.

and $\boldsymbol{\mathcal{Y}}_c$ is only 8.56% and 9.68% of the original size, respectively. In this case, the NDE of the second best baseline is 1.89x (1.81x) the error of TENDI with CHE (FSF) data. The best baseline is CP, which is a lower bound of the NDE we can achieve. Moreover, TENDIB, which does not have access to the aggregation matrices, works remarkably well. It reduces the NDE of the second best baseline, that uses the aggregation information, by 37.77% (30.98%) with "high agg" level on CHE (FSF) data.

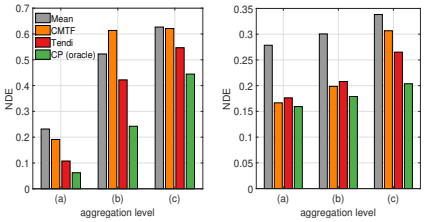
With Walmart data in Fig. 3 (c), "mod agg" means: weeks are aggregated into months in $\boldsymbol{\mathcal{Y}}_t$, and 45 stores are clustered into 15 groups in $\boldsymbol{\mathcal{Y}}_c$. Whereas, "high agg" is: weeks $\rightarrow$ quarterly samples in $\boldsymbol{\mathcal{Y}}_t$, and 45 stores $\rightarrow$ 9 groups in $\boldsymbol{\mathcal{Y}}_c$. CMTF works slightly better when the aggregation is moderate, which can be explained from the fact that departments (second mode in Walmart data) do not exhibit high correlation levels and thus the advantage of a tensor model over a matricized tensor one is not obvious. However, TENDI works markedly better with aggressive aggregation, even without using the aggregation information (TENDIB).

With Weather data (it has 93.30% zeros) in Fig. 3 (d), "mod agg" corresponds to the daily weather observations averaged into weekly samples in $\boldsymbol{\mathcal{Y}}_t$, and the 49 stations are clustered into 13 stations resolution in $\boldsymbol{\mathcal{Y}}_c$. On the other hand, days $\rightarrow$ months in $\boldsymbol{\mathcal{Y}}_t$, and 49 stations $\rightarrow$ 7 groups in $\boldsymbol{\mathcal{Y}}_c$ in the "high agg" level. Although CMTF and H-Fuse work better with this datasets compared to the other data, TENDI improves their error, especially with "high agg". HomeRun is excluded in Fig. 3 (d) as it imposes non-negativity. CMTF works better with this dataset owing to the fact that the second mode is small ($J = 17$), thus the advantage of a tensor over a matricized tensor model is less clear. H-Fuse works well as it imposes smoothness, and weather data are suitable for such constraint. Although TENDIB does not work as well as with other data, it still has smaller error than the simple baselines (Mean and LS), especially with aggressive aggregation. The CP error is invisible in Fig. 3 (d) as it is close to zero.

Results on Scenario B: In this scenario, $\boldsymbol{\mathcal{Y}}_c$ is doubly aggregated in two dimensions: stores *and* items, or crime locations *and* types. We test the performance on DFF and Crime data and compare with Mean, CMTF, and CP baselines. We omit the other baselines as they run out of memory. Difficulty (i.e., level of aggregation), increases as we move from case (a) to (c) in Fig. 4. With DFF data, these levels are: a) weeks $\rightarrow$ months in $\boldsymbol{\mathcal{Y}}_t$, and 93 stores $\rightarrow$ 16 areas in $\boldsymbol{\mathcal{Y}}_c$ with no aggregation over the items; b) weeks $\rightarrow$ months in $\boldsymbol{\mathcal{Y}}_t$, and 93 stores $\rightarrow$ 16

areas *and* 500 items → 50 categories in $\boldsymbol{\mathcal{Y}}_c$; and c) weeks → quarters (12 weeks), and 93 stores → 16 areas *and* 500 items → 20 categories in $\boldsymbol{\mathcal{Y}}_c$. One can see that TENDI significantly improves the disaggregation accuracy of the baselines with DFF data in Fig. 4 (a), with double aggregation *and* few available samples.

With Crime data in Fig. 4 (b), the aggregation levels are: a) months → quarters in $\boldsymbol{\mathcal{Y}}_t$, and 304 locations → 61 areas *and* 388 types → 78 categories in $\boldsymbol{\mathcal{Y}}_c$; b) months → quarters in $\boldsymbol{\mathcal{Y}}_t$, and 304 locations → 31 areas *and* 388 types → 39 categories in $\boldsymbol{\mathcal{Y}}_c$; and c) months → bi-yearly samples in $\boldsymbol{\mathcal{Y}}_t$, and 304 locations → 16 areas *and* 388 types → 20 categories in $\boldsymbol{\mathcal{Y}}_c$. Crime dataset is challenging as it has 91.56% zero values and small SD. The naive mean (Mean) has a relatively large NDE with moderate aggregation in case (a), which indicates that the task is difficult. Although CMTF performs slightly better with the the first two levels, TENDI becomes superior with extreme aggregation.



(a) DFF data (b) Crime data

Fig. 4: **Tendi works with double aggregation**.

5 Conclusions

In this work, we proposed a novel framework for fusing multiple aggregated views of multidimensional data. The proposed method leverages the properties of tensors in estimating the low-rank factors of the target data in higher resolution. The assumed model is provably transforming a highly ill-posed problem to an identifiable one. Experimental results show that the proposed algorithm is very effective, even with aggressive aggregation. The contributions of our work are summarized as follows: **1) Formulation**: we formally defined the problem of multidimensional data disaggregation from views aggregated in different dimensions; **2) Identifiability:** The considered tensor model provably converts a highly ill-posed problem to an identifiable one; **3) Effectiveness:** TENDI reduces the disaggregation error of the competing alternatives by up to 48% on real data; and **4) Unknown aggregation:** TENDIB works even when the aggregation mechanism is unknown.

Acknowledgements. The work of C. I. Kanatsoulis and N. D. Sidiropoulos was supported in part by the National Science Foundation under Grants NSF IIS-1704074, and NSF ECCS-1608961.

References

1. Almutairi, F.M., Sidiropoulos, N.D., Karypis, G.: Context-aware recommendation-based learning analytics using tensor and coupled matrix factorization. IEEE Journal of Selected Topics in Signal Processing **11**(5), 729–741 (2017)
2. Almutairi, F.M., Yang, F., Song, H.A., Faloutsos, C., Sidiropoulos, N., Zadorozhny, V.: Homerun: scalable sparse-spectrum reconstruction of aggregated historical data. Proc. of the VLDB Endowment **11**(11), 1496–1508 (2018)

3. Chiantini, L., Ottaviani, G.: On generic identifiability of 3-tensors of small rank. SIAM J. on Matrix Analys. and App. **33**(3), 1018–1037 (2012)
4. Chow, G.C., Lin, A.l.: Best linear unbiased interpolation, distribution, and extrapolation of time series by related series. The Review of Econ. and Stats. pp. 372–375 (1971)
5. Cole, D.: The effects of student-faculty interactions on minority students' college grades: Differences between aggregated and disaggregated data. J. of the Professoriate **3**(2) (2010)
6. Erkin, Z., Troncoso-Pastoriza, J.R., Lagendijk, R.L., Pérez-González, F.: Privacy-preserving data aggregation in smart metering systems: An overview. IEEE Signal Process. Mag. **30**(2), 75–86 (2013)
7. Ermiş, B., Acar, E., Cemgil, A.T.: Link prediction in heterogeneous data via generalized coupled tensor factorization. Data Mining and Knowledge Discovery **29**(1), 203–236 (2015)
8. Garrett, T.A.: Aggregated versus disaggregated data in regression analysis: implications for inference. Economics Letters **81**(1), 61–65 (2003)
9. Jin, Y., Williams, B.D., Tokar, T., Waller, M.A.: Forecasting with temporally aggregated demand signals in a retail supply chain. Journal of Business Logistics **36**(2), 199–211 (2015)
10. Kanatsoulis, C.I., Fu, X., Sidiropoulos, N.D., Akcakaya, M.: Tensor completion from regular sub-nyquist samples. IEEE T. on Sig. Proc. pp. 1–1 (2019)
11. Kanatsoulis, C.I., Fu, X., Sidiropoulos, N.D., Ma, W.: Hyperspectral superresolution: A coupled tensor factorization approach. IEEE T. on Signal Process. **66**(24), 6503–6517 (2018)
12. Kirrinnis, P.: Fast algorithms for the sylvester equation $ax - xb^t = c$. Theoretical Computer Science **259**(1), 623 – 638 (2001)
13. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM review **51**(3), 455–500 (2009)
14. Liu, Z., Song, H.A., Zadorozhny, V., Faloutsos, C., Sidiropoulos, N.: H-fuse: Efficient fusion of aggregated historical data. In: Proc. of the 2017 SIAM Int. Conf. on Data Mining. pp. 786–794. Houston, Texas, USA (April 2017)
15. Park, Y., Ghosh, J.: Ludia: An aggregate-constrained low-rank reconstruction algorithm to leverage publicly released health data. In: Proc. of the 20th ACM SIGKDD. pp. 55–64. ACM (2014)
16. Pavía-Miralles, J.M., Cabrer-Borrás, B.: On estimating contemporaneous quarterly regional gdp. J. of Forecasting **26**(3), 155–170 (2007)
17. Sidiropoulos, N.D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E.E., Faloutsos, C.: Tensor decomposition for signal processing and machine learning. IEEE Transactions on Signal Processing **65**(13), 3551–3582
18. Simões, M., Bioucas-Dias, J., Almeida, L.B., Chanussot, J.: A convex formulation for hyperspectral image superresolution via subspace-based regularization. IEEE Transactions on Geoscience and Remote Sensing **53**(6), 3373–3388 (2014)
19. Song, Q., Huang, X., Ge, H., Caverlee, J., Hu, X.: Multi-aspect streaming tensor completion. In: Proc. of the 23rd ACM SIGKDD Intl. Conf. on Knowl. Discovery and Data Mining. pp. 435–443 (2017)
20. Williams, B.D., Waller, M.A.: Creating order forecasts: point-of-sale or order history? J. of Business Logistics **31**(2), 231–251 (2010)
21. Yu, H.F., Rao, N., Dhillon, I.S.: Temporal regularized matrix factorization for high-dimensional time series prediction. In: Advances in neural information processing systems. pp. 847–855 (2016)