# PREMA: Principled Tensor Data Recovery from Multiple Aggregated Views

Faisal M. Almutairi, Charilaos I. Kanatsoulis, *Student Member, IEEE,*
and Nicholas D. Sidiropoulos, *Fellow, IEEE*

***Abstract*—Multidimensional data have become ubiquitous and are frequently encountered in situations where the information is aggregated over multiple data atoms. The aggregation can be over time or other features, such as geographical location. We often have access to multiple aggregated views of the same data, each aggregated in one or more dimensions, especially when data are collected or measured by different agencies. For instance, item sales can be aggregated temporally, and over groups of stores based on their location or affiliation. However, data mining and machine learning models benefit from detailed data for personalized analysis and prediction. Thus, data disaggregation algorithms are becoming increasingly important in various domains. The goal of this paper is to reconstruct finer-scale data from multiple coarse views, aggregated over different (subsets of) dimensions. The proposed method, called PREMA, leverages low-rank tensor factorization tools to fuse the multiple views and provide recovery guarantees under certain conditions. PREMA can tackle challenging scenarios, such as missing or partially observed data, double aggregation, and even blind disaggregation (without knowledge of the aggregation patterns) using a variant of PREMA called B-PREMA. To showcase the effectiveness of PREMA, the paper includes extensive experiments using real data from different domains: retail sales, crime counts, and weather observations.**

***Index Terms*—Data disaggregation, tensor decomposition, tensor mode product, multidimensional (tensor) data, multiview data**

## I. INTRODUCTION

DATA aggregation is the process of summing (or averaging) multiple data samples from a certain dataset, which results in data resolution reduction and compression. The most common type of aggregation is *temporal aggregation*. For example, the annual income is the aggregate of the monthly salary. Aggregation over other attributes is also common, e.g., data get aggregated geographically (e.g., population of New York) or according to a defined affiliation (e.g., employees of Company X). The latter is known in economics as *contemporaneous aggregation* [1]. The different types of aggregation are often combined, e.g., the number of foreigners who visited different US states in 2019 can be aggregated in time, location (states), and affiliation (nationality).

In some cases, it is the data collection process that limits data resolution in the first place, e.g., Store X records item sales only on a monthly basis. Aggregated data also exist for other reasons, the most important being data summarization. In particular, aggregated data enjoy concise representations, which is critical in the era of data deluge. Aggregation also benefits various other purposes, including scalability [2], communication and storage costs [3], and privacy [4]. Aggregated data are common in a wide range of domains, such as economics, health care [5], education [6], wireless communication, signal and image processing, databases [7], and smart grid systems [8].

Unfortunately, the favorable properties of data aggregation come with major shortcomings. A plethora of data mining and machine learning tasks strive for data in finer granularity (disaggregated), thus data aggregation is undesirable. Along the same lines, algorithms designed for personalized analysis and accurate prediction significantly benefit from enhanced data resolution. Analysis results can differ substantially when using aggregated versus disaggregated data. Particularly, studies in the field of economics show that data aggregation results in information loss and misleading conclusions at the individual level [9], [10]. Furthermore, in supply chain management, researchers have concluded that aggregating sales over time, products, or locations has a negative impact on demand forecasting [11]. On the other hand, disaggregation prior to analysis is very effective in environmental studies [12], and leads to richer findings in learning analytics [13].

The previous discussion reveals a clear *trade-off* between the need for data aggregation and the benefit of disaggregated data. This has motivated numerous works in developing algorithms for data disaggregation. In general, the task of data disaggregation is an inverse ill-posed problem. In order to handle the problem, classic techniques exploit side information or domain knowledge, in their attempt to make the problem overdetermined and consequently enhance the disaggregation accuracy. Some common prior models, imposed on the target higher resolution data, involve smoothness, periodicity [14], and non-negativity plus sparsity over a given dictionary [15]. Such prior constraints are invoked when no other information is available about the data to be disaggregated.

An interesting question arises when a dataset is 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 [16], [17]. In this case temporal *and* contemporaneous aggregated data are available [18]. For instance, given a country consisting of regions, we are interested in estimating the quarterly gross regional product (GRP) values, given the annual GRP per region (temporal aggregate) *and* the

This work was supported in part by the National Science Foundation under Grants IIS-1704074, and ECCS-1608961.

F. M. Almutairi and C. I. Kanatsoulis are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: almut012@umn.edu; kanat003@umn.edu).

N. D. Sidiropoulos is with the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22903, USA (e-mail: nikos@virginia.edu).

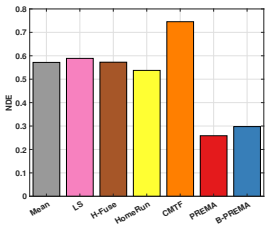



Fig. 1: **PREMA is effective with real data.**

quarterly national series (contemporaneous aggregate) [19]. 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 by groups of hospitals), often to preserve privacy [5].

Algorithms have been developed to integrate the multiple aggregates in the disaggregation task [16], [17], [18], [19], [20]. The general disaggregation problem is ill-posed, which is clearly undesirable, even with multiple aggregates. Therefore, the majority of these works resort to adopting linear regression models with priors and additional information. However, it is unclear whether these formulations can identify the true disaggregated dataset under reasonable conditions. In this context, identifiability has not received the attention it deserves, likely because guaranteed recovery is considered mission impossible under realistic conditions. With multiview data aggregated in different ways, however, the problem can be well-posed, as we will show in this paper.

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. Note that aggregation often happens in more than one dimensions *of the same data* as in the previously explained examples. The informal definition of the problem is given as follows:

*Informal Problem 1 (Multidimensional Disaggregation):*

- **Given:** two (or more) observations of a multidimensional dataset, each representing a different coarse view of the same data aggregated in one dimension (e.g., temporal and contemporaneous aggregates).
- **Recover:** the data in higher resolution (disaggregated) in all the dimensions.

We propose PREMA: a framework for fusing the multiple aggregates of multidimensional data. The proposed approach represents the target high resolution data as a *tensor*, and models that tensor using the *canonical polyadic decomposition* (CPD) to reduce the number of unknowns, while capturing correlations and higher-order statistical dependencies across dimensions. PREMA 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 overdetermined one, by leveraging the uniqueness properties of the CPD. PREMA is flexible in the sense that it can perform the disaggregation task on partially observed data, or data with missing entries. This is practically important as partially observed data appear often in real-world applications.

Our PREMA approach takes into account several well-known challenges that often emerge in real-life databases: the available measurements can have different scales (e.g., mixed monthly and yearly sums), gaps in the timeline (i.e., periods with no value reported), or time overlap (i.e., periods covered by more that one measurement). We also propose a variant of PREMA called B-PREMA that handles the disaggregation task in cases where the aggregation pattern is unknown. The proposed framework not only provides a disaggregation algorithm, but it also gives insights that can be exploited in creating more accurate data summaries for database applications. Interestingly, our work also provides insights on when aggregation *does not* preserve anonymity.

We evaluated PREMA on real data from different domains, i.e., retail sales, crime counts, and weather observations. Experimental results show that the proposed algorithm reduces the disaggregation error of the best baseline by up to 67%. Figure 1 shows the Normalized Disaggregation Error (NDE) of PREMA and the baselines with real data of the weekly sales counts of items in different stores of a retail company (CRA dataset, described in Section IV-A). We are given two observations: 1) monthly sales aggregates per store, and 2) weekly sales aggregated over groups of stores (94 stores are geographically divided into 18 areas). PREMA outperforms all the competitors, even if the aggregation pattern is unknown (B-PREMA)—all the baselines use the aggregation information. The fact that the naive mean (Mean) gives a large error, indicates that the data are not smooth and the task is difficult.

In summary, the contributions of our work are:

- **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:** PREMA recovers data with large improvement over the competing methods on real data.
- **Unknown aggregation:** the proposed model works even when the aggregation mechanism is unknown.
- **Flexibility :** PREMA can disaggregate partially observed data.

**Reproducibility:** The datasets we use are publicly available; our code is also available online[1].

Preliminary results of part of this work were presented in the *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) 2020* [21]. In this journal version, the problem formulation is generalized to handle aggregated data with missing entries. Although accounting for missing entries makes the problem more complicated, our proposed models and careful algorithmic design yield an algorithmic framework that is efficient and comparable to

[1]Code is available at https://github.com/FaisalAlmutairi/Prema

TABLE I: Symbols and Definitions

| Symbol | Definition |
|---|---|
| $\mathbf{x}$, $\mathbf{X}$, $\underline{\mathbf{X}}$ | Vector, matrix, tensor |
| $\mathbf{X}_n$ | Mode-n matricization (unfolding) |
| $\|.\|_F$ | Frobenius norm of a matrix/tensor |
| $\mathbf{X}^T$ | Transpose of matrix $\mathbf{X}$ |
| vec(.) | Vectorization operator of a matrix/tensor |
| $[\![.]\!]$ | Kruskal operator, e.g., $\underline{\mathbf{X}} \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ |
| $\circ$ | Outer product |
| $\otimes$ | Kronecker product |
| $\odot$ | Khatri-Rao product (column-wise Kronecker) |
| $\circledast$ | Hadamard (element-wise) product |

[21] (which does not handle missing entries), both in terms of accuracy and computational complexity. We also provide identifiability proofs, detailed model and complexity analysis, and conduct extensive experiments.

The rest of the paper is structured as follows. We explain the needed background and the related work in Section II, and introduce our proposed method in Section III. Then, we explain our experimental setup in Section IV and show the experimental results in Section V. Finally, we summarize conclusions and take-home points in Section VI.

## II. BACKGROUND & RELATED WORK

In this section, we review some tensor algebraic tools utilized by the proposed framework, define the disaggregation problem, and provide an overview of the related work. Table I summarizes the main symbols and operators used throughout the paper.

### A. Tensor Algebra

Tensors are multidimensional arrays indexed by three or more indices, $(i, j, k, ...)$. A third-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$ consists of three modes: rows $\underline{\mathbf{X}}(:, j, k)$, columns $\underline{\mathbf{X}}(i, :, k)$, and fibers $\underline{\mathbf{X}}(i, j, :)$. Moreover, $\underline{\mathbf{X}}(i, :, :)$, $\underline{\mathbf{X}}(:, j, :)$, and $\underline{\mathbf{X}}(:, :, k)$ denote the $i^{th}$ horizontal, $j^{th}$ lateral, and $k^{th}$ frontal slabs of $\underline{\mathbf{X}}$, respectively.

**Tensor decomposition (CPD/PARAFAC):** The outer product of two vectors $(\mathbf{a} \circ \mathbf{b})$ results in a rank-one matrix. A rank-one third-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$ is an outer product of three vectors: $\underline{\mathbf{X}}(i, j, k) = \mathbf{a}(i)\mathbf{b}(j)\mathbf{c}(k)$, $\forall i \in \{1, ..., I\}$, $j \in \{1, ..., J\}$, and $k \in \{1, ..., K\}$, i.e., $\underline{\mathbf{X}} = (\mathbf{a} \circ \mathbf{b} \circ \mathbf{c})$, 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) (also known as PARAFAC) of a third-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$ decomposes it into a sum of $R$ rank-one tensors [22], i.e.,

$$\underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \tag{1}$$

where $R$ is the *tensor rank* and represents the minimum number of outer products needed, and $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, and $\mathbf{c}_r \in \mathbb{R}^K$. For brevity, we use $\underline{\mathbf{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ to denote the relationship in (1). $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$ are the factor matrices with columns $\mathbf{a}_r$, $\mathbf{b}_r$ and $\mathbf{c}_r$ respectively, i.e., $\mathbf{A} = [\mathbf{a}_1 \;\; \mathbf{a}_2 \dots \mathbf{a}_R]$, and likewise for $\mathbf{B}$ and $\mathbf{C}$.

**CPD uniqueness:** An important property of the CPD is that $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ are essentially unique under mild conditions. CPD identifiability is established by the following theorem:

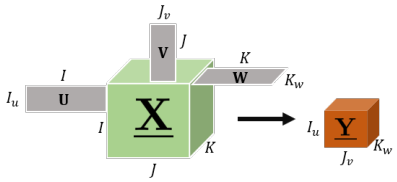



Fig. 2: Illustration of mode product with $(I_u < I)$, $(J_v < J)$, and $(K_w < K)$.

*Theorem 1:* [23] Let $\underline{\mathbf{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ with $\mathbf{A} : I \times R$, $\mathbf{B} : J \times R$, and $\mathbf{C} : K \times R$. Assume $I \geq J \geq K$ without loss of generality. If $R \leq \frac{1}{16}JK$, then the decomposition of $\underline{\mathbf{X}}$ in terms of $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ is essentially unique, almost surely – i.e., for almost every ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$) except for a set of Lebesgue measure zero.

Essential uniqueness means that $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ are unique up to common column permutation and scaling (scaling a column of one matrix that is compensated by counter-scaling the corresponding column of another matrix).

The CPD is also essentially unique, even if the tensor is incomplete (has missing entries). Several results have established CPD identifiability of tensors with missing entries, considering fiber sampled [24], regularly sampled [25] or randomly sampled tensors [26]. The conditions for uniqueness are in general stricter compared to the case where the full tensor is available.

**Tensor matricization (unfolding):** There are three different ways to unfold (obtain a matrix view of) a third-order tensor $\underline{\mathbf{X}}$ of size $I \times J \times K$. First, the mode-3 unfolding is obtained by the vectorization and parallel stacking of the frontal slabs $\underline{\mathbf{X}}(:, :, k)$ as follows [27]

$$\mathbf{X}_3 = [\text{vec}(\underline{\mathbf{X}}(:, :, 1)), ..., \text{vec}(\underline{\mathbf{X}}(:, :, K))] \quad \in \mathbb{R}^{IJ \times K}. \tag{2}$$

Equivalently, we can express $\mathbf{X}_3$ using the CPD factor matrices as $\mathbf{X}_3 = (\mathbf{B} \odot \mathbf{A})\mathbf{C}^T$. In the same vein, we may consider horizontal slabs to express the matricization over the first mode

$$\begin{aligned} \mathbf{X}_1 &:= [\text{vec}(\underline{\mathbf{X}}(1, :, :)), ..., \text{vec}(\underline{\mathbf{X}}(I, :, :))] \\ &= (\mathbf{C} \odot \mathbf{B})\mathbf{A}^T \in \mathbb{R}^{JK \times I} \end{aligned} \tag{3}$$

or lateral slabs to obtain mode-2 unfolding

$$\begin{aligned} \mathbf{X}_2 &:= [\text{vec}(\underline{\mathbf{X}}(:, 2, :)), ..., \text{vec}(\underline{\mathbf{X}}(:, J, :))] \\ &= (\mathbf{C} \odot \mathbf{A})\mathbf{B}^T \in \mathbb{R}^{IK \times J}. \end{aligned} \tag{4}$$

**Mode product:** It is the operation of multiplying a tensor by a matrix in one particular mode, e.g., mode-1 product of matrix $\mathbf{U} \in \mathbb{R}^{I_u \times I}$ and tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$ corresponds to multiplying every column $\underline{\mathbf{X}}(i, :, k)$ of the tensor by $\mathbf{U}$ [28]. Similarly, mode-2 (mode-3) product corresponds to multiplying every row (fiber) of $\underline{\mathbf{X}}$ by a matrix. Applying mode-1, mode-2, and mode-3 products to a third-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$ jointly is represented using the following notation:

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} \quad \in \mathbb{R}^{I_u \times J_v \times K_w} \tag{5}$$

where "$\times_n$" denotes the product over the $n^{th}$ mode, $\mathbf{U} \in \mathbb{R}^{I_u \times I}$, $\mathbf{V} \in \mathbb{R}^{J_v \times J}$, and $\mathbf{W} \in \mathbb{R}^{K_w \times K}$. Mode-1 multiplication results in reducing the tensor size in the first dimension *if*

($I_u < I$), similarly with the other modes; see Fig. 2. Moreover, if rows of $\mathbf{U}$ are binary vectors with more than one 1, then each horizontal slab of $\underline{\mathbf{Y}}$ is a sum of horizontal slabs of $\underline{\mathbf{X}}$ that correspond to the 1's in a particular row in $\mathbf{U}$. In the same vein, $\mathbf{V}$ and $\mathbf{W}$ could aggregate the lateral and frontal slabs, respectively. The mode product is also reflected in the CPD of the tensor, i.e., if $\underline{\mathbf{X}}$ in the operation in (5) admits $\underline{\mathbf{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, then $\underline{\mathbf{Y}} = [\![\mathbf{UA}, \mathbf{VB}, \mathbf{WC}]\!]$.

### B. Disaggregation Problem

The goal of the disaggregation task is to estimate a particular dataset in a higher resolution, given observations in lower resolution. In this subsection we present a high level linear algebraic view of disaggregation. This reveals the challenge of the task, which is the relationship between equations versus unknowns; detailed analysis follows in the next section.

In the disaggregation problem, we are given a set of measurements $\mathbf{y} \in \mathbb{R}^{I_u}$ aggregated over the dataset $\mathbf{x} \in \mathbb{R}^I$, and our goal is to find $\mathbf{x}$. 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 relates the measurements to the unknown variables. In this work, we consider the case where the target high-resolution data are multidimensional (tensor). Specifically, let $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$ 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). This 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 $\underline{\mathbf{X}}$ by an aggregation matrix in a particular mode(s). To see this, consider tensor $\underline{\mathbf{X}} \in \mathbb{R}^{4\times 2\times 2}$, a simple example of a set of observations aggregated over the first mode can be expressed as

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}}_{\mathbf{U}\in\mathbb{R}^{2\times 4}} \times \underbrace{\begin{bmatrix} x_{111} & x_{121} & x_{112} & x_{122} \\ x_{211} & x_{221} & x_{212} & x_{222} \\ x_{311} & x_{321} & x_{312} & x_{322} \\ x_{411} & x_{421} & x_{412} & x_{422} \end{bmatrix}}_{\mathbf{X}_1^T \in \mathbb{R}^{4\times(2\times 2)}} = \underbrace{\begin{bmatrix} y_{111} & y_{121} & y_{212} & y_{122} \\ y_{211} & y_{221} & y_{212} & y_{222} \end{bmatrix}}_{\mathbf{Y}_1^T \in \mathbb{R}^{2\times(2\times 2)}} \tag{6}$$

where $\mathbf{X}_1$ and $\mathbf{Y}_1$ are mode-1 unfolding of $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$, respectively. The same idea applies when the aggregation is over the second (third) mode using mode-2 (mode-3) product, respectively. In practical settings, the number of available aggregated measurements is much smaller than the number of variables (i.e., $I_u \ll I$), resulting in an under-determined, ill-posed problem. This is the major challenge of disaggregation, even when more than one set of aggregates are available. An even more challenging case appears when one of the available observation sets is aggregated over more than one mode/dimension simultaneously (e.g., $\underline{\mathbf{Y}} \in \mathbb{R}^{I_u \times J_v \times K}$, where $I_u < I$ and $J_v < J$ ). For instance, sales are reported for categories rather than individual items *and* over groups of stores. This is a double aggregation over stores and items, and the proposed method can work under such a challenging scenario. Moreover, the aggregated observations might be partially observed (i.e., $\mathbf{Y}_1$ in (6) has missing entries). This makes the problem more complicated, however our approach efficiently handles data with missing entries.

### C. Related Work

**Data disaggregation and fusion:** The problem of data integration and fusion [29], [30] from multiple sources has attracted the attention of several communities, due to the increasing access to all kinds of data, especially in database applications. A very challenging task in data integration, is that of recovering a sequence of events (e.g., time series) from multiple aggregated reports [31], [32], [15], [33]. A common approach is to formulate the problem as linear least squares as in (6). In practice, however, the number of available aggregated samples is often significantly smaller than the length of the target series, resulting in an under-determined system of equations. To resolve this, previous algorithms have resorted to Tikhonov-type regularization of the ill-posed problem to impose some domain knowledge constraints, e.g., smoothness and periodicity [14].

Fusing multiple observations aggregated in different dimensions for disaggregation purposes is a well studied task in the field of finance and economics [16], [17], [18], [19], [20]. The considered approaches try to exploit linear relations between the target series in high resolution and the available aggregated measurements. However, this results in an under-determined linear system, even with multiple aggregates. Therefore, the majority of these works assume linear regression models with priors and additional information. Moreover, it is unclear whether the assumed models are identifiable, i.e., the model is not guaranteed to disaggregate the data.

**(Coupled) tensor factorization:** Time series analysis, for various applications, is moving towards modern high-dimensional methods. For example, matrix and tensor factorization have been used in demand forecasting [34], mining and information extraction from complex time-stamped series [35], and prediction of unknown locations in spatio-temporal data [36].

Data share common dimension(s) in a wide spectrum of applications. In such cases, coupled factorization techniques are commonly used to fuse the information for various objectives. For example, coupled factorization is often employed to integrate contextual information into the main data [37]. In recommender systems, for instance, we have a (user $\times$ item $\times$ time) tensor *and* a (user $\times$ features) matrix. In this case, the tensor and the features matrix are coupled in the user mode [38]. Coupled tensor factorization has also been proposed for image processing [39], remote sensing [40], and medical imaging problems [41], [25]. Closest to our work is the approach in [42], 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 and its conference version [21] are the first that propose a tensor factorization approach to tackle data disaggregation applications.

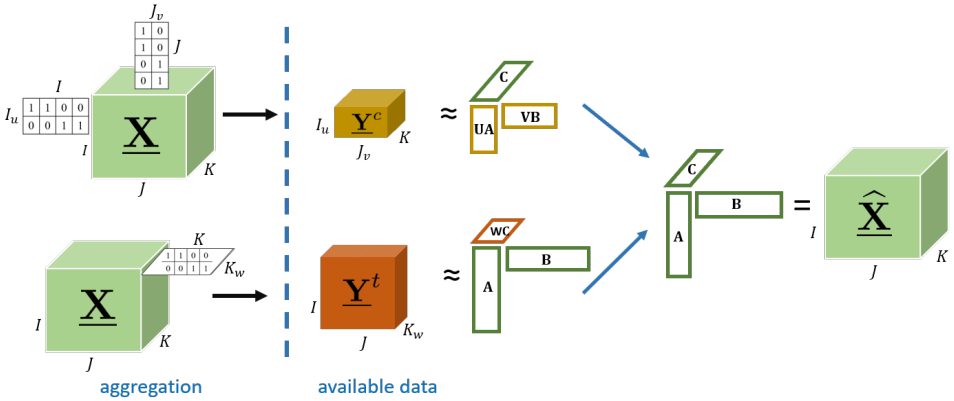



Fig. 3: Overview of PREMA.

## III. PROPOSED FRAMEWORK: PREMA

Multidimensional data are indexed by multiple indices, e.g., $(i, j, k)$. Therefore, they can naturally be represented as a tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$. The different modes represent the physical dimensions of the data (e.g., time stamps, locations, items, users). For the sake of simplicity of exposition, we focus on three-dimensional data in our formulations and algorithms. However, the proposed framework can handle more general cases with data of higher dimensions.

In the remainder of this section, we give a detailed description and analysis of PREMA. Particularly, we state the problem and explain the proposed model in high level in Section III-A, formulate PREMA in Section III-B, and present the main algorithm in Section III-C. We discuss the complexity of PREMA in Section III-D, and identifiability in section III-E. Finally, we introduce B-PREMA in Section III-F, to tackle the disaggregation problem in the case where the aggregation matrices are unknown.

### A. Problem & Model Overview

Multidimensional aggregation is common when data are collected or released by different agencies, resulting in multiple views of the same dataset. We will explain the concept with the example of retail sales, which we use in the experiments. Estimating the retail sales in higher resolution enables accurate forecasting of future demand, and planing of economically efficient commerce. There are two sources of data used for this forecasting task: 1) Point of Sale (POS) data at the store-level, commonly aggregated in time (temporal aggregate $\underline{\mathbf{Y}}^t$); and 2) historical orders made to the suppliers by the retailers' Distribution Centers (DC orders), aggregated over their multiple stores (contemporaneous aggregate $\underline{\mathbf{Y}}^c$). In particular, DC order data are immediately available to the suppliers, whereas the POS data are owned by the retailers. Both DC order and POS data are used to forecast demand, and especially POS data are vital in predicting future orders [43]. For that reason, many retailers share POS with their suppliers to assist in forecasting orders and avoid shortage or excess in inventory [44]. In a more restricted scenario, the second source collects information about each category of items rather than each item individually. Oftentimes, data are partially observed, i.e., $\underline{\mathbf{Y}}^t$ and $\underline{\mathbf{Y}}^c$ have missing entries. In this example, not all items are offered in all stores during all the considered time stamps. 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 the following:

*Problem 1 (***Multidimensional Disaggregation):**

- **Given:** two aggregated views of three-dimensional data $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$: $\underline{\mathbf{Y}}^t \in \mathbb{R}^{I \times J \times K_w}$, and $\underline{\mathbf{Y}}^c \in \mathbb{R}^{I_u \times J \times K}$ (or $\underline{\mathbf{Y}}^c \in \mathbb{R}^{I_u \times J_v \times K}$), with $I_u < I$, $J_v < J$, and $K_w < K$, and possibly missing entries.
- **Recover:** the original disaggregated multidimensional data $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$.

Note that each aggregated view is the result of the mode product of the target data with an aggregation matrix. In particular $\underline{\mathbf{Y}}^t = \underline{\mathbf{X}} \times_3 \mathbf{W}$, where $\mathbf{W} \in \mathbb{R}^{K_w \times K}$ is an aggregation matrix with $K_w < K$, and $\underline{\mathbf{Y}}^c = \underline{\mathbf{X}} \times_1 \mathbf{U}$, where $\mathbf{U} \in \mathbb{R}^{I_u \times I}$ is an aggregation matrix with $I_u < I$. In the case where one view is jointly aggregated in 2 dimensions, e.g., sales are aggregated over groups of stores and groups of items, $\underline{\mathbf{Y}}^c = \underline{\mathbf{X}} \times_1 \mathbf{U} \times_2 \mathbf{V}$, where $\mathbf{V} \in \mathbb{R}^{J_v \times J}$ is an aggregation matrix with $J_v < J$.

PREMA aims to fuse the different available aggregates in order to estimate the multidimensional data in the desired higher resolution. At a higher level, the main idea behind the proposed method is that the target multidimensional data, $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$, admits a CPD model. Therefore, it can be well approximated using its CPD factors $\mathbf{A}, \mathbf{B}, \mathbf{C}$ (i.e., $\underline{\mathbf{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$). Exploiting the low-rank modeling helps in reducing the number of unknown variables, especially if the data are highly correlated. Then, the CPD factors of the two aggregated observations are

$$\underline{\mathbf{Y}}^t = [\![\mathbf{A}, \mathbf{B}, \mathbf{WC}]\!], \tag{7}$$

$$\underline{\mathbf{Y}}^c = [\![\mathbf{UA}, \mathbf{VB}, \mathbf{C}]\!]. \tag{8}$$

PREMA learns the factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ by applying a coupled CPD model on the available aggregates with respect to the available observations. Note that up to this point, we have not explained how missing entries in $\underline{\mathbf{Y}}^t$ and $\underline{\mathbf{Y}}^c$ are treated, which will be discussed in the next section. Figure 3 illustrates the high level picture of our model.

### B. PREMA: *Formulation*

If we have the original (disaggregated) data in the tensor $\underline{\mathbf{X}}$ with missing entries, a common way to estimate its CPD factors is by adopting a least squares criterion to minimize the difference between the original tensor $\underline{\mathbf{X}}$ and its CPD $[\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ with respect to the available (observed) entries. This can be done by adding a weight tensor that masks the available entries, i.e.,

$$\underset{\mathbf{A},\mathbf{B},\mathbf{C}}{\text{minimize}} \quad \|\underline{\mathbf{\Omega}} \circledast (\underline{\mathbf{X}} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!])\|_F^2 \tag{9}$$

where $\underline{\mathbf{\Omega}}$ is defined as

$$\underline{\mathbf{\Omega}}(i,j,k) = \begin{cases} 1, & \text{if } \underline{\mathbf{X}}(i,j,k) \text{ is available} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Fortunately, many real life data exhibit low-rankness due to the correlation between the elements within each dimension (e.g., stores, items, time stamps), i.e., $R$ in (1) is small relative to the size of the tensor.

In the considered disaggregation task, we only have aggregated views of the multidimensional data (i.e., compressed version of the target tensor $\underline{\mathbf{X}}$). These aggregated views can have missing elements for various application-specific reasons such as privacy, lack of data collection, or absence of events. We use the fact that the aggregated tensors share the same factors (up to aggregation) as shown in equations (7) and (8) to jointly decompose $\underline{\mathbf{Y}}^t$ and $\underline{\mathbf{Y}}^c$ by means of coupled tensor factorization. To this end, we obtain the following formulation:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B},\mathbf{C}} \quad & \mathcal{F}(\mathbf{A},\mathbf{B},\mathbf{C}) := \|\underline{\mathbf{\Omega}}^t \circledast (\underline{\mathbf{Y}}^t - ([\![\mathbf{A}, \mathbf{B}, \mathbf{WC}]\!])\|_F^2 \\ & + \|\underline{\mathbf{\Omega}}^c \circledast (\underline{\mathbf{Y}}^c - ([\![\mathbf{UA}, \mathbf{VB}, \mathbf{C}]\!])\|_F^2 \end{aligned} \tag{11}$$

where $\underline{\mathbf{\Omega}}^t \in \{0,1\}^{I \times J \times K_w}$ and $\underline{\mathbf{\Omega}}^c \in \{0,1\}^{I_u \times J_v \times K}$ are weight tensors with ones at the indices of the available entries in $\underline{\mathbf{Y}}^t$ and $\underline{\mathbf{Y}}^c$, respectively, and zeros elsewhere. As a result, the CPD factors $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are learned with respect to the available data. One could add a regularization parameter $\lambda$ to control the weight between the two terms, however, we observed that it does not significantly affect the disaggregation performance. Enforcing non-negativity constraints on the factors seems natural *if* we are dealing with count data, however, we empirically observed that it does not improve the disaggregation accuracy. Note that if we have additional aggregated observations, we can incorporate them using the same concept. Although (11) assumes that the tensors are three-dimensional, we can handle higher-dimensional data following the same idea of coupling factors and mode product over any aggregated mode by the respective aggregation matrix. For example, assume that the data are four-dimensional and we observe an additional tensor $\underline{\mathbf{Y}}^a = \underline{\mathbf{X}} \times_4 \mathbf{L}$, where $\mathbf{L}$ is an aggregation matrix. Then, we add a fourth factor matrix $\mathbf{D}$ to the factorization terms in (11) (i.e., the first term becomes $\underline{\mathbf{Y}}^t = [\![\mathbf{A}, \mathbf{B}, \mathbf{WC}, \mathbf{D}]\!]$). In this case, we also add a term that minimizes the squared error in $\underline{\mathbf{Y}}^a = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{LD}]\!]$.

### C. PREMA: *Algorithm*

The optimization in (11) is non-convex, and NP-hard in general. To tackle it, we derive a *Block Coordinate Descent* (BCD) algorithm that updates the three variables in an alternating fashion. Starting from initial factors $\mathbf{A}^{(0)}$, $\mathbf{B}^{(0)}$, and $\mathbf{C}^{(0)}$, at every iteration $k \in \mathbb{N}$, we cyclically update each factor while fixing the other two. Each update is a step in the direction of the negative gradient of $\mathcal{F}$ with respect to the corresponding factor. To simplify the expressions, let us define $\widetilde{\mathbf{A}} = \mathbf{UA}$, $\widetilde{\mathbf{B}} = \mathbf{VB}$, and $\widetilde{\mathbf{C}} = \mathbf{WC}$. The partial derivative of the above objective function $\mathcal{F}$ w.r.t. $\mathbf{A}$ is as follows—the derivations are deferred to Appendix A.

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial \mathbf{A}} = \nabla_{\mathbf{A}} \mathcal{F} = 2\big(\underbrace{\mathbf{\Omega}_1^t \circledast ((\widetilde{\mathbf{C}} \odot \mathbf{B})\mathbf{A}^T - \mathbf{Y}_1^t)}_{\mathbf{E}_t}\big)^T (\widetilde{\mathbf{C}} \odot \mathbf{B}) \\ + 2\mathbf{U}^T\big(\underbrace{\mathbf{\Omega}_1^c \circledast ((\mathbf{C} \odot \widetilde{\mathbf{B}})\widetilde{\mathbf{A}}^T - \mathbf{Y}_1^c)}_{\mathbf{E}_c}\big)^T (\mathbf{C} \odot \widetilde{\mathbf{B}}) \end{aligned} \tag{12}$$

where $\mathbf{Y}_1^t$, $\mathbf{Y}_1^c$, $\mathbf{\Omega}_1^t$, and $\mathbf{\Omega}_1^c$ are mode-1 unfolding of the corresponding tensors. Similarly, we derive the derivatives of $\mathcal{F}$ w.r.t. $\mathbf{B}$ and $\mathbf{C}$ using mode-2 and mode-3 unfoldings of the tensors, respectively, and get the following equations:

$$\begin{aligned} \nabla_{\mathbf{B}} \mathcal{F} = 2\big(\mathbf{\Omega}_2^t \circledast ((\widetilde{\mathbf{C}} \odot \mathbf{A})\mathbf{B}^T - \mathbf{Y}_2^t)\big)^T (\widetilde{\mathbf{C}} \odot \mathbf{A}) \\ + 2\mathbf{V}^T\big(\mathbf{\Omega}_2^c \circledast ((\mathbf{C} \odot \widetilde{\mathbf{A}})\widetilde{\mathbf{B}}^T - \mathbf{Y}_2^c)\big)^T (\mathbf{C} \odot \widetilde{\mathbf{A}}), \end{aligned} \tag{13}$$

$$\begin{aligned} \nabla_{\mathbf{C}} \mathcal{F} = 2\mathbf{W}^T\big(\mathbf{\Omega}_3^t \circledast ((\mathbf{B} \odot \mathbf{A})\widetilde{\mathbf{C}}^T - \mathbf{Y}_3^t)\big)(\mathbf{B} \odot \mathbf{A}) \\ + 2\big(\mathbf{\Omega}_3^t \circledast ((\widetilde{\mathbf{B}} \odot \widetilde{\mathbf{A}})\mathbf{C}^T - \mathbf{Y}_3^c)\big)^T (\mathbf{C} \odot \widetilde{\mathbf{A}}). \end{aligned} \tag{14}$$

In the case of higher-dimensional data, mode-4 unfolding is used to derive the gradient w.r.t. the fourth factor, and so on for more dimensions. With the above gradient expressions at hand, we have established the update direction for each block (factor), which is the negative gradient of $\mathcal{F}$ with respect to each factor:

$$\mathbf{A} = \mathbf{A} - \alpha \nabla_{\mathbf{A}} \mathcal{F}, \tag{15}$$

$$\mathbf{B} = \mathbf{B} - \beta \nabla_{\mathbf{B}} \mathcal{F}, \tag{16}$$

$$\mathbf{C} = \mathbf{C} - \gamma \nabla_{\mathbf{C}} \mathcal{F}. \tag{17}$$

We now seek to select the step-size terms $\alpha$, $\beta$, and $\gamma$. We use the *exact line search* approach for this task. At every iteration $k \in \mathbb{N}$, $\alpha$ is chosen to minimize $\mathcal{F}$ along the line $\{\mathbf{A} - \alpha \nabla_{\mathbf{A}} \mathcal{F} | \alpha \geq 0\}$

$$\underset{\alpha \geq 0}{\text{argmin}} \quad \mathcal{F}\big(\mathbf{A} - \alpha \nabla_{\mathbf{A}} \mathcal{F}\big). \tag{18}$$

Luckily, in our case, the above optimization can be solved optimally without extra heavy computations. The optimal solution to (18) is as follows (refer to Appendix B for derivations).

$$\alpha = max\big(0, \frac{\mathbf{e}_t^T \mathbf{g}_t + \mathbf{e}_c^T \mathbf{g}_c}{\mathbf{g}_t^T \mathbf{g}_t + \mathbf{g}_c^T \mathbf{g}_c}\big), \tag{19}$$

where $\mathbf{e}_t = \text{vec}(\mathbf{E}_t)$, $\mathbf{e}_c = \text{vec}(\mathbf{E}_c)$, with $\mathbf{E}_t$ and $\mathbf{E}_c$ are as defined in (12), and

$$\mathbf{g}_t = \text{vec}(\mathbf{\Omega}_1^t \circledast ((\widetilde{\mathbf{C}} \odot \mathbf{B})\nabla_{\mathbf{A}} \mathcal{F}^T)), \tag{20}$$

$$\mathbf{g}_c = \text{vec}(\mathbf{\Omega}_1^c \circledast ((\mathbf{C} \odot \widetilde{\mathbf{B}})(\mathbf{U}\nabla_{\mathbf{A}} \mathcal{F})^T)). \tag{21}$$

Note that $\mathbf{E}_t$ and $\mathbf{E}_c$ are already computed in (12). We have also computed $(\widetilde{\mathbf{C}} \odot \mathbf{B})$ and $(\mathbf{C} \odot \widetilde{\mathbf{B}})$ in (12), which are needed

to obtain $\mathbf{g}_t$ amd $\mathbf{g}_c$, respectively. Thus, the exact line search step only requires:

- Multiplying the transpose of the gradient $\nabla_{\mathbf{A}}\mathcal{F} \in \mathbb{R}^{I\times R}$ by a $K_wJ \times R$ matrix in (20) (and $\mathbf{U}\nabla_{\mathbf{A}}\mathcal{F} \in \mathbb{R}^{I_u\times R}$ by a $KJ_v \times R$ matrix in (21)).
- Computing the inner products in (19).

In a similar fashion, $\beta$ and $\gamma$ are obtained by solving the following optimization functions, respectively:

$$\beta = \underset{\beta\geq 0}{\operatorname{argmin}} \quad \mathcal{F}\big(\mathbf{B} - \beta\nabla_{\mathbf{B}}\mathcal{F}\big), \tag{22}$$

$$\gamma = \underset{\gamma\geq 0}{\operatorname{argmin}} \quad \mathcal{F}\big(\mathbf{C} - \gamma\nabla_{\mathbf{C}}\mathcal{F}\big). \tag{23}$$

The solutions to the above are similar to the case of $\alpha$, but with mode-2 and mode-3 tensor unfoldings. We provide an illustrative example of deriving the solution to (18), (22)-(23) in Appendix B. The overall steps of PREMA are summarized in Algorithm 1.

**Algorithm 1** : PREMA (11)

**input:** $\underline{\mathbf{Y}}^t$, $\underline{\mathbf{Y}}^c$, $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$, $R$
**Initialize:** $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ (refer to Appendix C)
**Repeat**
- Update $\mathbf{A}$ using (15), (12), and (19)
- Update $\mathbf{B}$ using (16), (13), and (22)
- Update $\mathbf{C}$ using (17), (14), and (23)

**Until** criterion is met (max. #iterations)
**output:** $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$

We observed empirically that a careful initialization for the factor matrices in Algorithm 1 results in a better disaggregation accuracy, and substantially reduces the operational time (i.e., reduces the required number of iterations). Thus, we design a careful initialization method based on CPD. First, we set the missing entries to zero, then perform CPD on one tensor to get initial estimates of two factors. Then, we solve a system of linear equations using the other tensor to obtain an initial estimate of the third factor. For instance, from CPD($\underline{\mathbf{Y}}^t$) we get $\mathbf{A}$, $\mathbf{B}$, and $\widetilde{\mathbf{C}}$. Then, we obtain $\mathbf{C}$ by solving the linear system $\mathbf{Y}_3^c = \big((\mathbf{V}\mathbf{B}) \odot (\mathbf{U}\mathbf{A})\big)\mathbf{C}^T$. This way, we establish an initial guess for $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. We provide detailed initialization steps in Appendix C.

### D. PREMA: Complexity Analysis

The complexity of PREMA is determined by the matrix multiplication operations required to obtain the gradients and the step size terms. The products in the gradient expressions have the dominant computational cost. Therefore, we break down the computational complexity below using the gradient w.r.t. $\mathbf{A}$ in (12); the complexity of computing the gradients w.r.t $\mathbf{B}$ and $\mathbf{C}$ are similar. Recall (12):

$$\nabla_{\mathbf{A}}\mathcal{F} = 2\big(\underbrace{\underline{\mathbf{\Omega}}_1^t \circledast ((\widetilde{\mathbf{C}} \odot \mathbf{B})\mathbf{A}^T - \mathbf{Y}_1^t)}_{\mathbf{E}_t \in \mathbb{R}^{JK_w\times I}}\big)^T\big(\widetilde{\mathbf{C}} \odot \mathbf{B}\big)$$
$$+ 2\mathbf{U}^T\big(\underbrace{\underline{\mathbf{\Omega}}_1^c \circledast ((\mathbf{C} \odot \widetilde{\mathbf{B}})\widetilde{\mathbf{A}}^T - \mathbf{Y}_1^c)}_{\mathbf{E}_c \in \mathbb{R}^{J_vK\times I_u}}\big)^T\big(\mathbf{C} \odot \widetilde{\mathbf{B}}\big).$$

1) Computing the two Khatri-Rao products costs $\mathcal{O}(K_wJR + KJ_vR)$, where $R$ is the rank.
2) The cost of multiplying $(\widetilde{\mathbf{C}}\odot\mathbf{B})$ with $\mathbf{A}^T$, and $(\mathbf{C}\odot\widetilde{\mathbf{B}})$ with $\widetilde{\mathbf{A}}^T$ is $\mathcal{O}(IJK_wR + I_uJ_vKR)$.
3) The element-wise products ($\circledast$) cost $\mathcal{O}(nnz(\underline{\mathbf{\Omega}}^t) + nnz(\underline{\mathbf{\Omega}}^c))$.
4) Multiplying $\mathbf{E}_t^T$ and $\mathbf{E}_c^T$ with the Khatri-Rao products costs $\mathcal{O}(R(nnz(\underline{\mathbf{\Omega}}^t) + nnz(\underline{\mathbf{\Omega}}^c)))$.
5) In the worst case where $\mathbf{U}$ and $\mathbf{\Omega}_1^c$ have no zeros, the cost of multiplying $\mathbf{U}^T$ with $\mathbf{E}_c^T(\mathbf{C}\odot\widetilde{\mathbf{B}})$ is $\mathcal{O}(II_uR)$.

The dominant cost terms are in the $2^{nd}$ point above. Thus, the overall complexity is $\mathcal{O}(IJK_wR + I_uJ_vKR)$. Since $R$ is usually very small relative to the size of the tensors in real data, the complexity is linear with the size of $\underline{\mathbf{Y}}^t$ and $\underline{\mathbf{Y}}^c$.

### E. PREMA: Identifiability Analysis

After introducing the model and the algorithm, we establish the identifiability of the PREMA model. As mentioned earlier, the multidimensional disaggregation task is an inverse ill-posed problem. Considering a low rank CPD model on the data, results in a tensor disaggregation problem with a unique solution. In other words, the optimal solution of (11) is guaranteed to be unique, under mild conditions, and identify the original fine-resolution tensor almost surely. For the sake of simplicity we first assume that $\underline{\mathbf{Y}}^t$ does not have any missing values.

*Proposition 1:* Let $\underline{\mathbf{X}} \in \mathbb{R}^{I\times J\times K}$ be the target tensor to disaggregate with CPD $\underline{\mathbf{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ of rank $R$. Also let $\underline{\mathbf{Y}}^t \in \mathbb{R}^{I\times J\times K_w} = \underline{\mathbf{X}} \times_3 \mathbf{W}$ and $\underline{\mathbf{Y}}^c \in \mathbb{R}^{I_u\times J_v\times K} = \underline{\mathbf{\Omega}}^c \circledast (\underline{\mathbf{X}}\times_1\mathbf{U}\times_2\mathbf{V})$ be the two aggregated sets of observations. Assume that $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are drawn from some absolutely continuous joint distribution with respect to the Lebesque measure in $\mathbb{R}^{(I\times J\times K)R}$, and that $(\mathbf{A}^\star, \mathbf{B}^\star, \mathbf{C}^\star)$ is an optimal solution to problem (11). Also assume that the number of observed entries at each frontal slab of $\underline{\mathbf{Y}}^c$ is greater than or equal to $R$. Then, $\widehat{\underline{\mathbf{X}}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ disaggregates $\underline{\mathbf{Y}}^t$, $\underline{\mathbf{Y}}^c$ to $\underline{\mathbf{X}}$ almost surely if $R \leq \frac{1}{16}\min\{IJ, IK_w, JK_w, 16I_uJ_v\}$.

The proof is intuitive and parallels recent results obtained in the hyperspectral imaging literature [42]. **Proof sketch:** We use Theorem 1 to claim identifiability of $\underline{\mathbf{Y}}^t$. Then factors $\mathbf{A}$, $\mathbf{B}$ can be identified up to common permutation and scaling. The solution for $\mathbf{C}$ is obtained via solving an overdetermined linear system of equations using $\underline{\mathbf{Y}}^c$. This way permutation and scaling is preserved and the target tensor is recovered as $\underline{\mathbf{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$. In the case where $\underline{\mathbf{Y}}^t$ has missing entries, identifiability depends on the pattern of missings. Specifically, the results in [24], [25], [26] can be employed, when the available measurements are fiber, regularly or randomly sampled respectively. The conditions are more restrictive compared to the case of fully observed tensor, but guarantee identifiability of $\mathbf{A}$, $\mathbf{B}$ up to common permutation and scaling. The solution for $\mathbf{C}$ is the same as in the previous case. The detailed proof is presented in Appendix D.

### F. B-PREMA: PREMA with Unknown Aggregation

In most practical applications, the aggregation details are known. However, there exist cases with limited knowledge on

how the data are aggregated, i.e., we do not know (or have partial knowledge of) $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$. We consider the case where each available view is aggregated in one dimension, and propose the following formulation to get the factors of the disaggregated tensor ($\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$):

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C},\widetilde{\mathbf{A}},\widetilde{\mathbf{C}}} \mathcal{L}(\mathbf{A},\mathbf{B},\mathbf{C},\widetilde{\mathbf{A}},\widetilde{\mathbf{C}}) := \|\underline{\mathbf{\Omega}}^t \circledast (\underline{\mathbf{Y}}^t - [\![\mathbf{A},\mathbf{B},\widetilde{\mathbf{C}}]\!])\|_F^2 + \|\underline{\mathbf{\Omega}}^c \circledast (\underline{\mathbf{Y}}^c - [\![\widetilde{\mathbf{A}},\mathbf{B},\mathbf{C}]\!])\|_F^2 + \mu\mathcal{R}(\mathbf{C},\widetilde{\mathbf{C}}) \quad (24)$$

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 $\mathbf{W}$, and $\mathcal{R}$ is a regularization function. This problem is more challenging than (11) as the number of variables has been 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 regularization term in (24). Scaling and counter-scaling the factors of the tensor $\underline{\mathbf{Y}}^t$ (or $\underline{\mathbf{Y}}^c$) does not change its estimated value, or the value of the cost function in (24). For example, scaling $\mathbf{A}$ by a $\lambda$, and $\widetilde{\mathbf{C}}$ by $1/\lambda$ does not change the value of $\widehat{\mathbf{Y}}_1^t = (\widetilde{\mathbf{C}} \odot \mathbf{B})\mathbf{A}^T$, and as a result, it gives the same cost value. However, this scaling changes the estimated value of the disaggregated tensor $\widehat{\mathbf{X}}_1 = (\mathbf{C} \odot \mathbf{B})\mathbf{A}^T$. This is because tensor $\underline{\mathbf{X}}$ shares factors with both $\underline{\mathbf{Y}}^t$ and $\underline{\mathbf{Y}}^c$. To overcome this, we observe that the temporal aggregation $\mathbf{W}$ in most aggregated data is non-overlapping and includes all the time ticks[2]. This means that the respective column sums of $\mathbf{C}$ and $\widetilde{\mathbf{C}}$ should be equal. We exploit this observation by choosing the following regularization term for (24)

$$\mathcal{R}(\mathbf{C},\widetilde{\mathbf{C}}) = \|\mathbf{1}^T\mathbf{C} - \mathbf{1}^T\widetilde{\mathbf{C}}\|_2^2,$$

which reconciles for the scaling ambiguity.

In order to tackle the problem above, we derive a BCD algorithm, in the same fashion as Algorithm 1. The steps are summarized in Algorithm 2. We alternate between updating the five variables. In each update, we take a step in the direction of the negative gradient w.r.t. the corresponding variable. The derivations of the gradients are shown in Appendix A. The step size parameters $\alpha, \rho, \beta, \gamma$, and $\sigma$ are chosen using the exact line search explained in Sec. III-C above, and Appendix B.

To initialize the factors in Algorithm 2, we set the missing entries to zero, then we use `Tensorlab` and compute (`CPD`($\underline{\mathbf{Y}}^c$)) to get $\widetilde{\mathbf{A}}$, $\mathbf{B}$, and $\mathbf{C}$. 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. Therefore, we sum every consecutive $w = \frac{K}{K_W}$ rows in $\mathbf{C}$. This way we approximate the temporal aggregation process in a very intuitive way, the true aggregation matrix being unknown[3].

**Algorithm 2** : B-PREMA

**input:** $\underline{\mathbf{Y}}^t$, $\underline{\mathbf{Y}}^c$, $R$, $\mu$
**Initialize:** $\widetilde{\mathbf{A}}$, $\mathbf{B}$, $\mathbf{C}$, $\leftarrow$ `CPD`($\underline{\mathbf{Y}}^c$)
$\widetilde{\mathbf{C}}(k_w,:) \leftarrow \sum_{k=w(k_w-1)+1}^{w\times k_w} \mathbf{C}(k,:)$
$\mathbf{A} \leftarrow$ solve $\mathbf{Y}_3^t = \mathbf{A}(\widetilde{\mathbf{C}} \odot \mathbf{B})^T$
**Repeat**
- $\alpha \leftarrow \text{argmin}_{\alpha\geq 0}\ \mathcal{L}(\mathbf{A} - \alpha\nabla_{\mathbf{A}}\mathcal{L})$; $\mathbf{A} = \mathbf{A} - \alpha\nabla_{\mathbf{A}}\mathcal{L}$
- $\rho \leftarrow \text{argmin}_{\rho\geq 0}\ \mathcal{L}(\widetilde{\mathbf{A}} - \rho\nabla_{\widetilde{\mathbf{A}}}\mathcal{L})$; $\widetilde{\mathbf{A}} = \widetilde{\mathbf{A}} - \rho\nabla_{\widetilde{\mathbf{A}}}\mathcal{L}$
- $\beta \leftarrow \text{argmin}_{\beta\geq 0}\ \mathcal{L}(\mathbf{B} - \beta\nabla_{\mathbf{B}}\mathcal{L})$; $\mathbf{B} = \mathbf{B} - \beta\nabla_{\mathbf{B}}\mathcal{L}$
- $\gamma \leftarrow \text{argmin}_{\gamma\geq 0}\ \mathcal{L}(\mathbf{C} - \gamma\nabla_{\mathbf{C}}\mathcal{L})$; $\mathbf{C} = \mathbf{C} - \gamma\nabla_{\mathbf{C}}\mathcal{L}$
- $\sigma \leftarrow \text{argmin}_{\sigma\geq 0}\ \mathcal{L}(\widetilde{\mathbf{C}} - \sigma\nabla_{\widetilde{\mathbf{C}}}\mathcal{L})$; $\widetilde{\mathbf{C}} = \widetilde{\mathbf{C}} - \sigma\nabla_{\widetilde{\mathbf{C}}}\mathcal{L}$

**Until** termination criterion is met (max. #iterations)
**output:** $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$

## IV. EXPERIMENTAL DESIGN

In this section, we provide a detailed description of the setup we use in our experiments. First, we describe the data used in the experiments. Then, we explain the aggregation applied on these data to generate aggregated views. Last, we present the evaluation metrics and baselines used for comparison.

### A. Datasets

We evaluate PREMA using the following public datasets, which are readily available online:

**DFF**[4]**:** Retail sales data, called Dominick's Finer Foods (DFF), collected by the James M. Kilts Center, University of Chicago Booth School of Business. DFF used to be a grocery store chain based in the Chicago area until all of its stores were closed. Sales, in this dataset, are divided into category-specific files. In particular, each file contains the weekly sales (i.e., number of sold units) of items belonging to a specific category (e.g., cheese, cookies, soft drinks, etc) in about 100 stores. DFF data contain the geographical locations of the different stores, which we use to aggregate stores into groups. We create ground truth three-dimensional tensors, using 10 different category-specific datasets. This way, a (stores $\times$ items $\times$ weeks) tensor is formed *for each category*. These 10 department-specific datasets are listed as the first group in Table II—we use the three bold letters acronym for these categories in the results. We pick the 50 most popular items from each category. Note that this results in an 'incomplete' tensor, owing to the fact that not all items were offered in all stores, or they were offered only for part of the time in some stores. These tensors have varying statistics (see Table II), which allows thorough testing and analysis. We also form an additional (stores $\times$ items $\times$ weeks) tensor that contains items from all the 10 different categories combined, 50 items from each (namely **Mixed DFF** in Table II).

**Walmart**[5]**:** Historical weekly sales data for 99 different departments in 45 Walmart stores located in different regions. A (stores $\times$ departments $\times$ weeks) tensor is created from these data. The resulting tensor is complete and has no missing entries. The size of each store (in square feet) is included in the data (we use this information to form groups of stores).

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

[3]In the experiments, we make sure that the true temporal aggregation and the estimated one do not align.

[4]https://www.chicagobooth.edu/research/kilts/datasets/dominicks

[5]https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data

TABLE II: Summary of datasets and their statistics.

| Dataset ($\underline{\mathbf{X}}$) | Size | Max | Avg | SD | % (missing entries) | % (zero entries) |
|---|---|---|---|---|---|---|
| **BATh Soap** | $93 \times 50 \times 266$ | 52 | 0.79 | 1.34 | 44.73 | 33.37 |
| **B**ottled **J**ui**C**es | $93 \times 50 \times 393$ | 12288 | 13.76 | 50.08 | 8.79 | 9.19 |
| **CHE**eses | $93 \times 50 \times 393$ | 18176 | 26.65 | 88.29 | 8.59 | 5.51 |
| **COO**kies | $94 \times 50 \times 390$ | 14080 | 16.00 | 56.86 | 9.81 | 7.57 |
| **CRA**ckers | $94 \times 50 \times 382$ | 14080 | 8.21 | 29.61 | 14.21 | 7.57 |
| **C**anned **SO**up | $93 \times 50 \times 379$ | 34494 | 40.46 | 133.42 | 8.64 | 4.54 |
| **F**abric **S**o**F**teners | $93 \times 50 \times 397$ | 7168 | 5.68 | 18.84 | 18.64 | 27.48 |
| **GRO**oming | $93 \times 50 \times 272$ | 232 | 1.94 | 2.94 | 7.66 | 32.66 |
| **P**aper **T**o**W**els | $93 \times 50 \times 389$ | 19712 | 45.36 | 117.82 | 36.72 | 23.49 |
| **S**oft **DR**inks | $93 \times 50 \times 391$ | 18944 | 48.81 | 155.09 | 8.58 | 11.18 |
| **Mixed DFF** | $93 \times 500 \times 230$ | 17610 | 19.01 | 71.30 | 15.30 | 17.83 |
| **Walmart** | $45 \times 81 \times 143$ | 6.93e+05 | 1.29e+04 | 2.14e+04 | 0 | 19.38 |
| **Crime** | $304 \times 388 \times 221$ | 325 | 0.26 | 1.47 | 0 | 91.56 |
| **Weather** | $49 \times 17 \times 365$ | 1038 | 10.23 | 95.65 | 0 | 93.30 |

**Crime**[6]**:** Reported incidents of crimes that occurred in the city of Chicago from 2001 to present. Each incident is marked with its beat (police geographical area), and a code indicating the crime type. There are 304 geographical areas and 388 crime types in total. Using this dataset, we form a (locations (by beat) $\times$ crime types $\times$ months) tensor.

**Weather**[7]**:** Daily weather observations from 49 stations in Australia. These observations contain 17 different variables, e.g., min temperature, max temperature, cloud, humidity, wind, etc. We form a (station (location) $\times$ variables $\times$ days) tensor using one year of daily observations.

Table II summarizes the different datasets described above, with their size, maximum and average values, Standard Deviation (SD), and percentage of missing entries and zeros. These datasets are the ground truth in our experiments, and represented by $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$.

### B. Aggregation Configuration

The aggregated observations (compressed tensors), that are used as inputs to the disaggregation methods, are generated from $\underline{\mathbf{X}}$ following two practical scenarios described below:

**Scenario A**: The multidimensional data, we aim to disaggregate, are represented by $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$. Instead of the full tensor $\underline{\mathbf{X}}$, we are given two aggregated views: 1) temporally aggregated tensor $\underline{\mathbf{Y}}^t = \underline{\mathbf{X}} \times_3 \mathbf{W}$, i.e., aggregated in the third dimension; and 2) contemporaneously aggregated tensor $\underline{\mathbf{Y}}^c = \underline{\mathbf{X}} \times_1 \mathbf{U}$, aggregated in the first mode (e.g., stores/locations dimension). We use the 10 category-specific datasets from DFF and Walmart data to test this scenario. The stores are aggregated according to their geographical locations in the DFF datasets, and based on their sizes in Walmart data. We also test this scenario on Weather data, where the temporal aggregate represents the weather observations averaged over a course of time, and the contemporaneous aggregate is the average of the observations over a geographical region.

**Scenario B**: In this scenario, two aggregated views of $\underline{\mathbf{X}}$ are given: 1) similar to the previous scenario, temporally aggregated tensor $\underline{\mathbf{Y}}^t = \underline{\mathbf{X}} \times_3 \mathbf{W}$; and 2) contemporaneously aggregated tensor $\underline{\mathbf{Y}}^c = \underline{\mathbf{X}} \times_1 \mathbf{U} \times_2 \mathbf{V}$, aggregated in the first *and* second dimensions (e.g., sales counts that are *jointly* aggregated over groups of stores *and* groups of items). We use Mixed DFF and Crime data to test this scenario. The stores are aggregated into groups according to their locations in Mixed DFF data, whereas items are aggregated according to their categories. In Crime data, locations and types are grouped based on the closeness in geographical location and similarity in crime type, respectively. Note that when $\mathbf{V} = \mathbf{I}$, this yields to Scenario A. Evidently, this scenario is more challenging since the second observation is aggregated in two modes, i.e., double aggregation, resulting in fewer measurements.

The difficulty of the problem also depends on the *aggregation level*, i.e., the number of data points (e.g., weeks, items, or stores) in one sum. Fewer aggregated measurements result in more challenging problems from an "equations versus unknowns" standpoint. We test the disaggregation performance using different aggregation levels for each dimension.

### C. Evaluation Baselines & Metrics

We evaluate the disaggregation performance of the proposed method using the Normalized Disaggregation Error (NDE = $\|\underline{\mathbf{X}} - \widehat{\underline{\mathbf{X}}}\|_F^2 / \|\underline{\mathbf{X}}\|_F^2$), where $\widehat{\underline{\mathbf{X}}}$ is the estimated data. The baseline methods are described next. Note that we compare to state-of-art approaches in the time series disaggregation literature as well as methods developed to fuse multiple views of multidimensional data, but for different tasks. To the best of our knowledge our work is the first to perform disaggregation on multidimensional data from multiple views.

**Mean:** This baseline assumes that the constituent data atoms (entries in $\underline{\mathbf{X}}$) have equal contribution in their aggregated samples. The final estimate of Mean is the average of the estimation from the temporal and the contemporaneous aggregates. For example, the contemporaneous aggregate reports 100 units sold in 10 stores in the first week of January, and the temporal one tells us that 80 units were sold in January (4 weeks) in Store 1. Then, Mean estimation of week 1 and store 1 is $(100/10 + 80/4)/2 = 15$

**LS:** This baseline is inspired by [19], [20], where a least squares criterion is adopted on the linear relationship between the target time series in high resolution and the available aggregates. The resulting linear system is underdetermined, thus, these works assume a linear regression model between the target series and some set of indicators. In their context, indicators are time series available in high resolution that are expected to display similar fluctuations to the target series. For example, the stock price of an oil company is a linear combination of the stock prices of other relevant companies. This assumption requires additional data that are not available

[6]https://www.kaggle.com/chicago/chicago-crime/activity
[7]http://www.bom.gov.au/climate/data/

in our datasets. Therefore, we resort to the minimum-norm solution

$$\min_{\underline{\mathbf{X}}} \quad \|\text{vec}(\mathbf{\Omega}_3^{t\,T}) \circledast \big(\text{vec}(\mathbf{Y}_3^{t\,T}) - \widetilde{\mathbf{W}}\text{vec}(\mathbf{X}_3{}^T)\big)\|_2^2 + \|\text{vec}(\mathbf{\Omega}_3^c) \circledast \big(\text{vec}(\mathbf{Y}_3^c) - \widetilde{\mathbf{U}}\text{vec}(\mathbf{X}_3)\big)\|_2^2 \tag{25}$$

where $\widetilde{\mathbf{W}} = \mathbf{I} \otimes \mathbf{W}$ and $\widetilde{\mathbf{U}} = \mathbf{I} \otimes \mathbf{V} \otimes \mathbf{U}$.

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

**HomeRun:**[15] To circumvent the indeterminacy of the linear system in the time series disaggregation problem, this baseline solves for the disaggregated series in the frequency domain. More specifically, HomeRun searches for the coefficients of the Discrete Cosine Transform (DCT) that represent the target high-resolution series. The key point is that the number of non-negligible DCT coefficients of the time series is much smaller than its length. In other words, the DCT is used as a sparsifying dictionary to reduce the number of variables. HomeRun also imposes smoothness and non-negativity constraints.

**CMTF:** Couple Matricized Tensor Factorization has been widely used, to fuse multiple views of multidimensional data, in the hyperspectral imaging application [45], [46]—the work in [45] adds non-negativity constraints. These images are three-dimensional tensors, and the motivation behind these works is to exploit the low-rankness of the matricized image. We compare to this model because real world multidimensional data are often well-approximated using low-rank, as we will show empirically. Using our notation, CMTF solves

$$\min_{\mathbf{A},\mathbf{B}} \quad \|\mathbf{\Omega}_3^t \circledast (\mathbf{Y}_3^t - \mathbf{A}(\mathbf{WB})^T)\|_F^2 + \|\mathbf{\Omega}_3^c \circledast (\mathbf{Y}_3^c - (\mathbf{V} \otimes \mathbf{U})\mathbf{A}\mathbf{B}^T)\|_F^2. \tag{26}$$

We solve (26) using a BCD algorithm with exact line search. Similar to PREMA, a good initialization for the low-rank factors improves the performance of CMTF. To ensure fair comparison, we initialize using SVD with missing entries set to be zeros.

Note that all the baselines described above use the aggregation information; B-PREMA is the only method that disaggregates without using the aggregation matrices. In addition to the above baseline methods, we also test the estimation of the target disaggregated data with the following *oracle* baseline.

**CPD**: We fit a CPD model directly to the ground truth tensor $\underline{\mathbf{X}}$ *with respect to the observed entries*. We use the Matlab-based package `Tensorlab` to compute the CPD. Then, we reconstruct $\widehat{\underline{\mathbf{X}}}$ from the learned factors $(\mathbf{A}, \mathbf{B}, \mathbf{C})$. This baseline can also serve as a lower bound for the error produced by the proposed method PREMA.

## V. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of PREMA and B-PREMA in terms of disaggregation accuracy using real data. The two aforementioned aggregation scenarios (refer to Section IV-B) are considered with different aggregation levels. In the experiments, we choose the rank $R$ for PREMA (and the CPD baseline) based on Proposition 1, unless stated otherwise. On the other hand, for CMTF, we perform a grid search and show the results with the best $R$. We run 10 iterations of the CPD step in the initialization of PREMA in Algorithm 1 (or B-PREMA in Algorithm 2) using `Tensorlab`, then run 10 iterations of the iterative procedure in the algorithms. We set $\mu = 100$ for B-PREMA in Algorithm 2. All experiments were performed using Matlab on a Linux server with an Intel Core i7–4790 CPU 3.60 GHz processor and 32 GB memory.

### A. Results on Scenario A

Two aggregated views $\underline{\mathbf{Y}}^t$, $\underline{\mathbf{Y}}^c$ are observed. Table III shows the disaggregation error in terms of NDE, achieved by the proposed method and the baselines on the 10 category-specific datasets from DFF. The proposed methods, PREMA and B-PREMA, along with the CPD oracle are shown under 3 different ranks ($R = 10,\ R = 25,\ R = 40$). In $\underline{\mathbf{Y}}^t$, the weekly sales counts are observed on a monthly basis, while in $\underline{\mathbf{Y}}^c$, the 93 (or 94 for some categories) stores are clustered geographically into 18 areas. This means that the measurements in the temporal aggregate $\underline{\mathbf{Y}}^t$ are about 25% of the original size, and the number of the contemporaneously aggregated measurements in $\underline{\mathbf{Y}}^c$ is only 19.35% of the disaggregated data size.

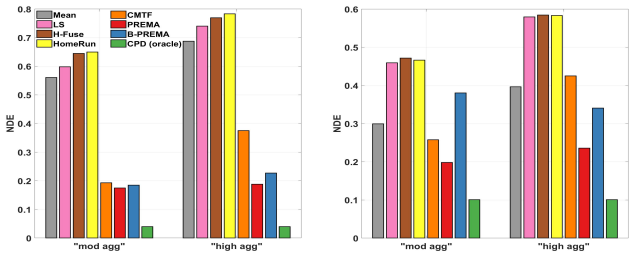
For all datasets in Table III, except BAT, PREMA markedly outperforms the baselines—to highlight the improvement, we make the smallest error in bold and underline the second smallest. The naive mean (Mean) is good enough with BAT dataset because it is smooth (SD = 1.34) and has the largest percentage of missing entries, compared to the other datasets. The time series methods, H-Fuse and HomeRun, do not perform well with these datasets because they are designed for smooth and quasi-periodic data, respectively. To provide an example, we noticed that HomeRun improves the error of LS and H-Fuse baselines with CRA data, and found that CRA exhibit more periodicity compared to the rest of the categories. Comparing PREMA with CPD, we see that PREMA achieves error very close to CPD of the ground truth data with the same rank, e.g., with GRO, PTW, and SDR datasets. By looking at the performance of B-PREMA in the table, we can see that the proposed algorithm works remarkably well when the aggregation matrices are unknown. For example, with GRO data and $R = 40$, the NDE of B-PREMA is 0.2472, while NDE = 0.2284 with CPD. B-PREMA disaggregates with smaller, or very similar, error compared to the baselines that uses the aggregation pattern information—see results with CRA, FSF, GRO, and SDR datasets. With all datasets, there is always a wide range of $R$ under which the proposed algorithm works similarly well.

Next, we examine the performance when we change the level of aggregation from moderate ("mod agg") to very high ("high agg"). The disaggregation error is shown with two datasets from DFF data, FSF and PTW, in Figure 4, and with Walmart and Weather datasets in Figure 5.

The aggregation levels in Figure 4 are: 1) monthly basis measurements (every 4 weeks) in $\underline{\mathbf{Y}}^t$, and the 93 stores are divided geographically into 18 areas ("mod agg"); and 2) quarterly samples (every 12 weeks) in $\underline{\mathbf{Y}}^t$, and the stores are divided into only 9 areas ("high agg"). The rank $R$ for

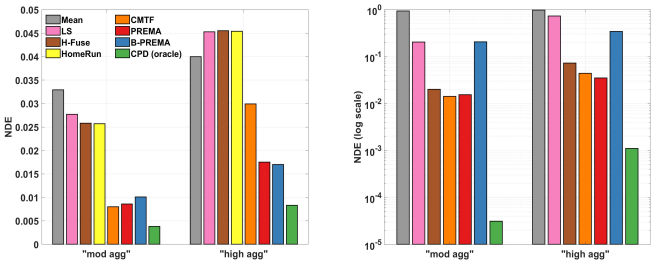
TABLE III: NDE of the proposed methods and the baselines using the 10 category-specific datasets.

| Dataset | BAT | BJC | CHE | COO | CRA | CSO | FSF | GRO | PTW | SDR |
|---|---|---|---|---|---|---|---|---|---|---|
| % (missings) | 44.73% | 8.79% | 8.59% | 9.81% | 14.21% | 8.64% | 18.64% | 7.66% | 36.72% | 8.58% |
| SD | 1.34 | 50.08 | 88.29 | 56.86 | 29.61 | 133.42 | 18.84 | 2.94 | 117.82 | 155.09 |
| **Mean** | **0.3284** | 0.4441 | 0.3118 | 0.3596 | 0.5217 | 0.3309 | 0.5609 | 0.2464 | 0.2994 | 0.2860 |
| **LS** | 0.3328 | 0.6077 | 0.4650 | 0.6224 | 0.5889 | 0.4664 | 0.5982 | 0.2831 | 0.4593 | 0.5420 |
| **H-FUSE** | 0.3411 | 0.6437 | 0.4870 | 0.6414 | 0.5726 | 0.4885 | 0.6451 | 0.2863 | 0.4719 | 0.5644 |
| **HomeRun** | 0.3461 | 0.6453 | 0.4818 | 0.6284 | 0.5376 | 0.4856 | 0.6496 | 0.2877 | 0.4662 | 0.5594 |
| **CMTF** | 0.4254 | 0.1818 | 0.1954 | 0.1783 | 0.7455 | 0.1564 | 0.1930 | 0.2908 | 0.2577 | 0.1633 |
| **PREMA**, R=10 | 0.5203 | 0.1978 | 0.1756 | 0.1757 | **0.2587** | 0.2057 | 0.2019 | 0.3198 | 0.2844 | 0.2039 |
| **PREMA**, R=25 | 0.5079 | 0.1684 | 0.1516 | 0.1371 | 0.2624 | 0.1373 | 0.1790 | 0.2581 | 0.2132 | 0.1438 |
| **PREMA**, R=40 | 0.4972 | **0.1572** | **0.1491** | **0.1318** | 0.2589 | **0.1332** | **0.1747** | **0.2458** | **0.1969** | **0.1348** |
| **CPD** (oracle), R=10 | 0.4782 | 0.0937 | 0.0723 | 0.1205 | 0.0776 | 0.0776 | 0.0810 | 0.2919 | 0.2356 | 0.1329 |
| **CPD** (oracle), R=25 | 0.4345 | 0.0586 | 0.0419 | 0.0676 | 0.0518 | 0.0476 | 0.0494 | 0.2448 | 0.1358 | 0.0822 |
| **CPD** (oracle), R=40 | 0.4109 | 0.0443 | 0.0321 | 0.0532 | 0.0438 | 0.0345 | 0.0399 | 0.2284 | 0.1007 | 0.0605 |
| **B-PREMA**, R=10 | 0.5242 | 0.3012 | 0.3525 | 0.2207 | 0.3080 | 0.1752 | 0.2090 | 0.3156 | 0.3594 | 0.2008 |
| **B-PREMA**, R=25 | 0.5002 | 0.3583 | 0.3553 | 0.2496 | 0.2976 | 0.1756 | 0.1892 | 0.2557 | 0.3758 | 0.1539 |
| **B-PREMA**, R=40 | 0.4914 | 0.3909 | 0.3823 | 0.2942 | 0.3042 | 0.1825 | 0.1846 | 0.2472 | 0.3963 | 0.1620 |




(a) FSF dataset (b) PTW dataset

Fig. 4: **PREMA works well with extreme aggregation.**




(a) Walmart dataset (b) Weather dataset[8]

Fig. 5: **PREMA works well with different data.**

PREMA, B-PREMA, and CPD is set to 40 in this figure. By comparing the moderate and high aggregation levels in Figure 4, we conclude that PREMA is more robust with aggressive aggregation where only few samples are available. With "high agg", the number of aggregation samples is only 8.56% of the original size in the temporal aggregate, and 9.68% in the contemporaneous aggregate. In this case, the NDE of the best baseline is 3.04 (1.68)x the error of PREMA with FSF (PTW) dataset, respectively. PTW dataset is more challenging as it has relatively high percentage of missing entries (36.72%). Moreover, with no knowledge of the aggregation pattern, B-PREMA outperforms all baselines that have access to the aggregation information with FSF data. Although, B-PREMA has NDE larger than Mean and CMTF with "mod agg" on PTW data, it becomes superior to all baselines when the aggregation level is high.

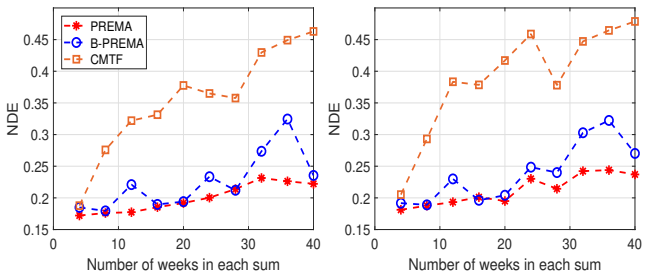
With Walmart data in Figure 5 (a), "mod agg" means that weeks are aggregated into months in $\underline{\mathbf{Y}}^t$, and the 45 stores are divided into 15 groups, whereas time is aggregated quarterly (12 weeks) and stores are clustered into 9 groups in "high agg". CMTF works slightly better when the aggregation is moderate, owing to the fact that the second mode in Walmart data is departments as apposed to items in DFF data. Departments are less correlated than items from the same category. As a result, the advantage of tensor models over the matricized tensor in capturing the higher-order dependencies becomes less clear. However, PREMA is more immune to aggressive aggregation. In the "high level" case, The NDE of CMTF is 1.71 times the error of PREMA. Even without access to the aggregation information, B-PREMA significantly reduces the error of the baselines.

In Figure 5 (b), "mod agg" corresponds to the daily weather measurements averaged into weekly samples, and the 49 stations are averaged over 13 stations. On the other hand, the daily measurements are averaged over monthly samples, and the 49 stations are clustered into 7 stations in the "high agg" case. PREMA, CMTF, and H-Fuse perform similarly with Weather data[8] (it has 93.30% zeros) with moderate aggregation. The size of the second dimension of Weather data is small ($J = 17$), thus, the advantage of a tensor model over a matricized tensor model is less clear. H-Fuse works well with this data as it penalizes the large jumps between the adjacent time ticks (i.e., days), and weather data are well suited for such constraint. Nevertheless, PREMA improves the error of CMTF and H-Fuse when the aggregation level is high. Although B-PREMA 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.

Next, we show the disaggregation performance on a wider range of aggregation levels using FSF dataset. The results are shown in Figure 6. The number of areas in $\underline{\mathbf{Y}}^c$ is fixed to 18 in Figure 6 (a) and 9 in Figure 6 (b), whereas the number of weeks in each sum in $\underline{\mathbf{Y}}^t$ ranges from 4 to 40 ($x$-axis). The total number of weeks in the dataset is 397; thus, we only have 10 temporally aggregated samples if we have 40 weeks in each sum. In this set of results, we focus on comparing the proposed models with CMTF since it is the best performing among the baselines. The rank is set to $R = 40$ for PREMA and B-PREMA, while for CMTF we use a grid search to

[8]HomeRun is excluded from the results with Weather data as it has non-negativity constraints.

select the best rank. One can see that the proposed models are less affected as the aggregation level increases, even when the aggregation matrices are unknown with B-PREMA.




(a) 18 areas in $\underline{\mathbf{Y}}^c$ (b) 9 areas in $\underline{\mathbf{Y}}^c$

Fig. 6: **PREMA is more immune to aggressive aggregation.**

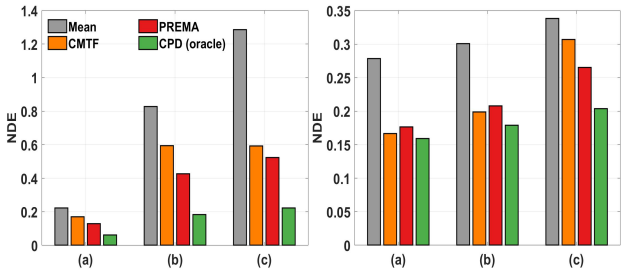
### B. Results on Scenario B

The contemporaneous aggregate $\underline{\mathbf{Y}}^c$ in this scenario is aggregated in two dimensions: stores and items with Mixed DFF data, or crime locations and types with Crime data[9]. We test this with three different aggregation levels with each data. Difficulty (i.e., level of aggregation), increases as we move from case (a) to (c)—Figure 7 shows the performance for these three cases. B-PREMA is not included in this set of experiments as it does not perform well. The reason is because double aggregation significantly reduces the number of equations, and the number of unknown parameters in B-PREMA is almost doubled since $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{C}}$ are treated as separate variables from $\mathbf{A}$ and $\mathbf{C}$. Combining double aggregation and blind disaggregation makes it hard for the identifiability conditions to be satisfied.

With Mixed DFF data, these levels are: a) $\underline{\mathbf{Y}}^t$ aggregates weeks into monthly samples, while $\underline{\mathbf{Y}}^c$ groups the 93 stores into 18 areas with no aggregation over the items, b) samples in $\underline{\mathbf{Y}}^t$ have monthly resolution, and $\underline{\mathbf{Y}}^c$ groups the stores into 18 areas *and* items into groups of 10, and c) $\underline{\mathbf{Y}}^t$ contains temporal aggregates for each quarter of the year, and $\underline{\mathbf{Y}}^c$ groups stores into 18 areas *and* items into groups of 25. One can see that the naive mean totally fails and its error exceeds 1 in case (c) with Mixed DFF data in Figure 7 (a). Notwithstanding, PREMA works well with double aggregation *and* few available samples.

With Crime data, the aggregation levels are: a) $\underline{\mathbf{Y}}^t$ aggregates the months into quarterly resolution, while $\underline{\mathbf{Y}}^c$ clusters both the crime locations and types into groups of 5, b) $\underline{\mathbf{Y}}^t$ has a quarterly time resolution, and $\underline{\mathbf{Y}}^c$ aggregates both the locations and types into groups of 10, and c) $\underline{\mathbf{Y}}^t$ aggregates the months into bi-yearly resolution, and $\underline{\mathbf{Y}}^c$ groups the crime locations and types into groups of 20. Figure 7 (b) shows the performance with these levels using Crime data. These data are challenging as they have 91.56% zero values and small SD. PREMA reduces the error of Mean significantly. Although CMTF performs slightly better with the first two levels, PREMA becomes superior with extreme aggregation.

[9]LS, H-Fuse, and HomeRun are excluded from this comparison as they run out of memory.




(a) Mixed DFF dataset (b) Crime dataset

Fig. 7: **PREMA works well with double aggregation (Scenario B).**

### C. Run time Comparison

In Table IV, we compare the run time of all the different methods for disaggregating the FSF dataset with the same setup as in Table III and $R = 40$. We can see that PREMA and B-PREMA are very scalable and faster than all the baselines (except for Mean, which only requires simple averaging). Our methods handle the missing entries very efficiently compared to the plain vanilla CPD using `TensorLab`.

TABLE IV: Run time comparisons.

| Method | Run time (seconds) |
|---|---|
| Mean | 0.10 |
| LS | 222.53 |
| H-Fuse | 6116.34 |
| HomeRun | 117.10 |
| CMTF | 1.26 |
| CPD | 13.85 |
| PREMA | 0.90 |
| B-PREMA | 0.89 |

## VI. CONCLUSIONS

In this work, we proposed a novel framework, called PREMA, 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. PREMA works with partially observed data, and can disaggregate effectively, even without any knowledge of the aggregation mechanism (B-PREMA). Experimental results on real data show that the proposed algorithm is very effective, even in challenging scenarios, such as data with double aggregation and high level of aggregation. The contributions of our work are summarized as follows:

- **Formulation**: we formally defined the problem of multidimensional data disaggregation from views aggregated in different dimensions.
- **Identifiability:** The considered tensor model provably converts a highly ill-posed problem to an identifiable one.
- **Effectiveness:** PREMA reduced the disaggregation error of the competing alternatives by up to 67%.
- **Unknown aggregation:** B-PREMA works even when the aggregation mechanism is unknown.
- **Flexibility :** PREMA can perform disaggregation on partially observed data.

## APPENDIX A
## DERIVATION OF GRADIENT EXPRESSIONS

The terms in (11) and (24) can be divided into two types: 1) CPD of a tensor, with some aggregation matrices multiplied with the factors; and 2) the regularization term $\mathcal{R}$ in (24). Because the gradient of a sum is the sum of the gradients, it is enough to show the derivation of the gradients using the function below. This function consists of two terms, each represents one of the terms types listed above. Consider:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \underbrace{\|\underline{\mathbf{\Omega}} \circledast (\underline{\mathbf{X}} - (\llbracket \mathbf{UA}, \mathbf{VB}, \mathbf{WC} \rrbracket)\|_F^2}_{\mathcal{T}} + \underbrace{\|\mathbf{1}^T\mathbf{C} - \mathbf{1}^T\widetilde{\mathbf{C}}\|_2^2}_{\mathcal{R}} \tag{27}$$

where $\underline{\mathbf{\Omega}}$ is as defined in (10), and $\underline{\mathbf{X}} \in \mathbb{R}^{I\times J\times K}$ is our data tensor. Note that all the CPD terms in (11) and (24) are similar to the term $\mathcal{T}$, with one (or more) of the aggregation matrices $\{\mathbf{U}, \mathbf{V}, \mathbf{W}\}$ is equal to $\mathbf{I}$. Thus, the term $\mathcal{T}$ generalizes all the CPD terms in our models. Using mode-3 unfolding, $\mathcal{T}$ is equivalent to

$$\mathcal{T} = \|\mathbf{\Omega}_3 \circledast \big(\mathbf{X}_3 - ((\mathbf{VB}) \odot (\mathbf{UA}))(\mathbf{WC})^T\big)\|_F^2. \tag{28}$$

Vectorizing the above, we get

$$\mathcal{T} = \|\mathbf{Sx} - \mathbf{S}((\mathbf{VB}) \odot (\mathbf{UA}) \odot (\mathbf{WC}))\mathbf{1}\|_F^2 \tag{29}$$

where $\mathbf{x} = \text{vec}(\mathbf{X}_3)$, and $\mathbf{S} \in \{0, 1\}^{N\times IJK}$ is a fat matrix with one 1 in each row to select the available entries in $\mathbf{x}$, where $N = nnz(\underline{\mathbf{\Omega}})$. Thus, the role of $\mathbf{S}$ with $\mathbf{x}$, is similar to the role of $\underline{\mathbf{\Omega}}$ with the tensor form $\underline{\mathbf{X}}$. Equation (29) is also equivalent to

$$\mathcal{T} = \|\mathbf{Sx} - \mathbf{S}\big(\mathbf{I} \otimes ((\mathbf{VB}) \odot (\mathbf{UA}))\big)(\mathbf{W} \otimes \mathbf{I})\mathbf{c}\|_F^2 \tag{30}$$

where $\mathbf{c} = \text{vec}(\mathbf{C}^T)$. We show the derivative of $\mathcal{T}$ and $\mathcal{R}$ w.r.t. $\mathbf{C}$ (derivatives w.r.t. $\mathbf{A}$ and $\mathbf{B}$ are derived similarly by using mode-1 and mode-2 unfolding and rotating the factors accordingly). First, we derive the gradient of $\mathcal{T}$ w.r.t. $\mathbf{C}$:

$$\nabla_{\mathbf{C}}\mathcal{T} = 2(\mathbf{W}^T \otimes \mathbf{I})(\mathbf{I} \otimes (\mathbf{VB} \odot \mathbf{UA})^T)\mathbf{S}^T\mathbf{S}\ (\mathbf{I} \otimes (\mathbf{VB} \odot \cdot \mathbf{UA}))(\mathbf{W} \otimes \mathbf{I})\mathbf{c} - 2(\mathbf{W}^T \otimes \mathbf{I})(\mathbf{I} \otimes (\mathbf{VB} \odot \mathbf{UA})^T)\mathbf{S}^T\mathbf{Sx} \tag{31}$$

$$= 2(\mathbf{I} \otimes (\mathbf{VB} \odot \mathbf{UA})^T)(\mathbf{W}^T \otimes \mathbf{I})\mathbf{S}^T\mathbf{S}(\mathbf{I} \otimes (\mathbf{VB} \odot \cdot \mathbf{UA}))(\mathbf{W} \otimes \mathbf{I})\mathbf{c} - 2(\mathbf{I} \otimes (\mathbf{VB} \odot \mathbf{UA})^T)(\mathbf{W}^T \otimes \mathbf{I})\mathbf{S}^T\mathbf{Sx} \tag{32}$$

$$= 2(\mathbf{I} \otimes (\mathbf{VB} \odot \mathbf{UA})^T)(\mathbf{W}^T \otimes \mathbf{I})\mathbf{S}^T\mathbf{S}\big((\mathbf{I} \otimes (\mathbf{VB} \odot \cdot \mathbf{UA}))(\mathbf{W} \otimes \mathbf{I})\mathbf{c} - \mathbf{x}\big). \tag{33}$$

We can use mode-3 unfolding to rewrite (33) above as

$$\nabla_{\mathbf{A}}\mathcal{T} = 2\mathbf{W}^T\big(\mathbf{\Omega}_3 \circledast (\widehat{\mathbf{X}}_3 - \mathbf{X}_3)\big)^T\big((\mathbf{VB}) \odot (\mathbf{UA})\big) \tag{34}$$

where $\widehat{\mathbf{X}}_3 = \big((\mathbf{VB}) \odot (\mathbf{UA})\big)(\mathbf{WC})^T$. The gradient above can be computed efficiently by the following steps:

1) Compute $\mathbf{L} = \mathbf{\Omega}_3 \circledast (\widehat{\mathbf{X}}_3 - \mathbf{X}_3)$.
2) Compute $\mathbf{M} = \mathbf{L}^T\big((\mathbf{VB}) \odot (\mathbf{UA})\big)$.
3) Compute $2\mathbf{W}^T\mathbf{M}$

Next, the derivative of $\mathcal{R}$ w.r.t. $\mathbf{C}$ is

$$\nabla_{\mathbf{C}}\mathcal{R} = 2(\mathbf{11}^T\mathbf{C} - \mathbf{11}^T\widetilde{\mathbf{C}}). \tag{35}$$

## APPENDIX B
## DERIVATION OF STEP SIZE EXPRESSIONS

The step size terms in both Algorithm 1 and 2 are chosen using the exact line search optimization method. Recall (27)

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \underbrace{\|\underline{\mathbf{\Omega}} \circledast (\underline{\mathbf{X}} - (\llbracket \mathbf{UA}, \mathbf{VB}, \mathbf{WC} \rrbracket)\|_F^2}_{\mathcal{T}} + \underbrace{\|\mathbf{1}^T\mathbf{C} - \mathbf{1}^T\widetilde{\mathbf{C}}\|_2^2}_{\mathcal{R}}.$$

As mentioned earlier in Appendix A, the function above generalizes all the terms in PREMA and B-PREMA models. Thus, we use (27) to show how to find the step size $\gamma$ associated with updating $\mathbf{C}$ as an illustrative example. In this case, the exact line search chooses $\gamma$ to be the minimizer of

$$\underset{\gamma\geq 0}{\text{argmin}} \quad \mathcal{F}\big(\mathbf{C} - \gamma\nabla_{\mathbf{C}}\mathcal{F}\big) \tag{36}$$

where $\mathcal{F} = \mathcal{L} + \mathcal{R}$, which are as defined in (27). Plugging the variable $\mathbf{C} - \gamma\nabla_{\mathbf{C}}\mathcal{F}$ into (27) and rearranging the terms, we get

$$\begin{aligned}\underset{\gamma\geq 0}{\text{argmin}} \quad & \|\underbrace{\mathbf{\Omega}_3 \circledast \big(\mathbf{Y}_3 - \big((\mathbf{VB}) \odot (\mathbf{UA})\big)\mathbf{W}^T\mathbf{C}^T\big)}_{\mathbf{E}} \\ & + \gamma\underbrace{\mathbf{\Omega}_3 \circledast \big((\mathbf{VB} \odot \mathbf{UA})\mathbf{W}^T\nabla_{\mathbf{C}}\mathcal{F}^T\big)}_{\mathbf{D}}\|_F^2 \\ & + \|\underbrace{\mathbf{1}^T\mathbf{C} - \mathbf{1}^T\widetilde{\mathbf{C}}}_{\mathbf{e}^T} - \gamma\underbrace{\mathbf{1}^T\nabla_{\mathbf{C}}\mathcal{F}}_{\mathbf{d}^T}\|_2^2.\end{aligned} \tag{37}$$

One can see that at the optimal solution to (37), we have:

$$-\text{vec}(\mathbf{E})^T = \gamma\text{vec}(\mathbf{D})^T \tag{38}$$

$$\mathbf{e}^T = \gamma\mathbf{d}^T \tag{39}$$

Multiplying (38) by $\text{vec}(\mathbf{D})$ and (39) by $\mathbf{d}$, and summing up the resulting two equations, we get

$$-\text{vec}(\mathbf{E})^T\text{vec}(\mathbf{D}) + \mathbf{e}^T\mathbf{d} = \gamma(\text{vec}(\mathbf{D})^T\text{vec}(\mathbf{D}) + \mathbf{d}^T\mathbf{d}) \tag{40}$$

Respecting the non-negativity constraint, we can see that the optimal solution is

$$\gamma = max\Big(0, \frac{-\text{vec}(\mathbf{E})^T\text{vec}(\mathbf{D}) + \mathbf{e}^T\mathbf{d}}{\text{vec}(\mathbf{D})^T\text{vec}(\mathbf{D}) + \mathbf{d}^T\mathbf{d}}\Big) \tag{41}$$

## APPENDIX C
## INITIALIZATION ALGORITHM

The initialization steps of Algorithm 1 are as follows

Set missing entries in $\underline{\mathbf{Y}}^t$, and $\underline{\mathbf{Y}}^c$ to zeros.
**if** $\mathbf{V} = \mathbf{I}$ and $K > I$ **then**
  $\widetilde{\mathbf{A}}, \mathbf{B}, \mathbf{C} \leftarrow$ `CPD`$(\underline{\mathbf{Y}}^c)$;
  $\mathbf{A} \leftarrow$ solve $\mathbf{Y}_1^t = ((\mathbf{WC}) \odot \mathbf{B})\mathbf{A}^T$
**else**
  $\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{C}} \leftarrow$ `CPD`$(\underline{\mathbf{Y}}^t)$;
  $\mathbf{C} \leftarrow$ solve $\mathbf{Y}_3^c = ((\mathbf{VB}) \odot (\mathbf{UA}))\mathbf{C}^T$
**end if**

Note that the missing entries are set to 0 only in the initialization steps. We use the Matlab-based package `Tensorlab` to compute the CPD in the initialization (e.g., `CPD`$(\underline{\mathbf{Y}}^c)$).

## APPENDIX D
## PROOF OF PROPOSITION 1

Let $\underline{\mathbf{X}} \in \mathbb{R}^{I\times J\times K}$ be the target tensor to disaggregate with CPD $\underline{\mathbf{X}} = [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]$ of rank $R$ and $\underline{\mathbf{Y}}^t \in \mathbb{R}^{I\times J\times K_w} = \underline{\mathbf{X}}\times_3 \mathbf{W}$. Then, under the conditions of Theorem 1, $\underline{\mathbf{Y}}^t$ admits a unique CPD $\underline{\mathbf{Y}}^t = [\![\mathbf{A}_t,\mathbf{B}_t,\mathbf{C}_t]\!]$. Since it is unique, it holds that:

$$\mathbf{A}_t = \mathbf{A}\mathbf{\Pi}\mathbf{\Lambda}_1, \mathbf{B}_t = \mathbf{B}\mathbf{\Pi}\mathbf{\Lambda}_2, \mathbf{C}_t = \mathbf{W}\mathbf{C}\mathbf{\Pi}\mathbf{\Lambda}_3, \tag{42}$$

where $\mathbf{\Pi}$ is a permutation matrix and $\mathbf{\Lambda}_1,\ \mathbf{\Lambda}_2,\ ,\mathbf{\Lambda}_3$ are diagonal matrices such that $\mathbf{\Lambda}_1\mathbf{\Lambda}_2\mathbf{\Lambda}_3 = \mathbf{I}$. In the case where $\underline{\mathbf{Y}}^t$ has missing entries the conditions under which $[\![\mathbf{A}_t,\mathbf{B}_t,\mathbf{C}_t]\!]$ are identifiable are stricter and depend on the pattern of misses. We can use the conditions in [24], [25], [26] for fiber, regular and random sampling respectively. So far, factors $\mathbf{A},\ \mathbf{B}$ have been identified up to column permutation and scaling. What remains to be proven is that:

$$\underline{\mathbf{\Omega}}^c \circledast \underline{\mathbf{Y}}^c = \underline{\mathbf{\Omega}}^c \circledast (\underline{\mathbf{X}} \times_1 \mathbf{U} \times_2 \mathbf{V}) = \underline{\mathbf{\Omega}}^c \circledast ([\![\mathbf{UA},\mathbf{VB},\mathbf{C}]\!]) \tag{43}$$

yields a solution for $\mathbf{C}_c$ such that $\mathbf{C}_c = \mathbf{C}\mathbf{\Pi}\mathbf{\Lambda}_3$. Equation (43) can be equivalently written as:

$$\mathbf{S}_c\mathbf{y}_c = \mathbf{S}_c(\mathbf{C}\odot\mathbf{VB}\odot\mathbf{UA})\mathbf{1} = \mathbf{S}_c(\mathbf{I}\otimes(\mathbf{VB}\odot\mathbf{UA}))\mathbf{c}, \tag{44}$$

where $\mathbf{y}_c, \mathbf{c}$ are vectorized versions of $\underline{\mathbf{Y}}^c,\ \mathbf{C}^T$ , and $\mathbf{S}_c \in \{0,1\}^{N_c\times I_uJ_vK}$ is a fat selection matrix that selects the available entries in $\mathbf{y}_c$, where $N_c = nnz(\underline{\mathbf{\Omega}}^c)$.

Now let $\widetilde{\mathbf{A}} = \mathbf{UA}$ and $\widetilde{\mathbf{B}} = \mathbf{VB}$. Following [42, Lemma 1] $\widetilde{\mathbf{A}},\ \widetilde{\mathbf{B}}$ are drawn from absolutely continuous non-singular distributions. Also let $\mathbf{P} = \widetilde{\mathbf{B}}\odot\widetilde{\mathbf{A}}$. Since $I_uJ_v \geq R$ the determinant of any $R\times R$ submatrix of $\mathbf{P}$ is a non-trivial analytic function of $\widetilde{\mathbf{A}},\ \widetilde{\mathbf{B}}$. Therefore any $R\times R$ minor of $\mathbf{P}$ is non-zero almost surely [47, Lemma 3] and any $R$ rows of $\mathbf{P}$ are independent.

Taking a closer look at matrix $\mathbf{G} = \mathbf{I}\otimes(\mathbf{VB}\odot\mathbf{UA}) = \mathbf{I}\otimes(\widetilde{\mathbf{B}}\odot\widetilde{\mathbf{A}})$ we observe that it is an $I_uJ_vK\times KR$ block diagonal matrix of the form:

$$\mathbf{G} = \begin{bmatrix} \mathbf{P} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{P} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \\ \vdots \\ \mathbf{G}_K \end{bmatrix} \tag{45}$$

Each block $\mathbf{G}_k$ corresponds to the $k-$th frontal slab of $\underline{\mathbf{Y}}^c$ and the rows between different $\mathbf{G}_k$'s are independent by construction. Since we have assumed that the minimum number of observed entries for each frontal slab is greater than or equal to $R$, then $\mathbf{S}_c\mathbf{G}$ has full column rank equal to $KR$ and the solution for $\mathbf{c}$ in (46) is unique with probability 1. Plugging $\mathbf{A}_t,\ \mathbf{B}_t$ in equation (46) we get:

$$\begin{aligned}\mathbf{S}_c\mathbf{y}_c &= \mathbf{S}_c(\mathbf{C}\odot\mathbf{VB}_t\odot\mathbf{UA}_t)\mathbf{1} \\ &= \mathbf{S}_c(\mathbf{C}\odot\mathbf{VB}\mathbf{\Pi}\mathbf{\Lambda}_2\odot\mathbf{UA}\mathbf{\Pi}\mathbf{\Lambda}_1)\mathbf{1}\end{aligned} \tag{46}$$

Then the unique solution for $\mathbf{C}$ satisfies $\mathbf{C}_c = \mathbf{C}\mathbf{\Pi}\mathbf{\Lambda}_3$ and $\widehat{\underline{\mathbf{X}}} = [\![\mathbf{A}_t,\mathbf{B}_t,\mathbf{C}_c]\!]$ disaggregates $\underline{\mathbf{Y}}^t,\ \underline{\mathbf{Y}}^c$ to $\underline{\mathbf{X}}$ almost surely.

## REFERENCES

[1] A. Silvestrini and D. Veredas, "Temporal aggregation of univariate and multivariate time series models: a survey," *J. of Econ. Surveys*, vol. 22, no. 3, pp. 458–497, 2008.
[2] S. Uludag, K.-S. Lui, K. Nahrstedt, and G. Brewster, "Analysis of topology aggregation techniques for qos routing," *ACM Computing Surveys (CSUR)*, vol. 39, no. 3, p. 7, 2007.
[3] P. D. Patel, P. B. Lapsiwala, and R. V. Kshirsagar, "Data aggregation in wireless sensor network," *International Journal of Managment, IT and Engineering*, vol. 2, no. 7, pp. 457–472, 2012.
[4] E. Shi, H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Proc. of the Network and Distributed System Security Symposium (NDSS 2011)*, San Diego, California, USA, Feb. 2011.
[5] Y. Park and J. Ghosh, "Ludia: An aggregate-constrained low-rank reconstruction algorithm to leverage publicly released health data," in *Proc. of the 20th ACM SIGKDD*, New York, NY, USA, 2014.
[6] R. H. Ellaway, M. V. Pusic, R. M. Galbraith, and T. Cameron, "Developing the role of big data and analytics in health professional education," *Medical Teacher*, vol. 36, no. 3, pp. 216–222, 2014.
[7] I. Motakis and C. Zaniolo, "Temporal aggregation in active database rules," in *Proc. of the 1997 ACM SIGMOD Intl. Conf. on Mgmt. of Data*, New York, NY, USA, 1997.
[8] Z. Erkin, J. R. Troncoso-Pastoriza, R. L. Lagendijk, and F. Pérez-González, "Privacy-preserving data aggregation in smart metering systems: An overview," *IEEE Signal Process. Mag.*, vol. 30, no. 2, pp. 75–86, 2013.
[9] W. A. Clark and K. L. Avery, "The effects of data aggregation in statistical analysis," *Geographical Analysis*, vol. 8, no. 4, pp. 428–438, 1976.
[10] T. A. Garrett, "Aggregated versus disaggregated data in regression analysis: implications for inference," *Economics Letters*, vol. 81, no. 1, pp. 61–65, 2003.
[11] Y. Jin, B. D. Williams, M. A. Waller, and A. R. Hofer, "Masking the bullwhip effect in retail: the influence of data aggregation," *Intl. J. of Physical Distrib. & Logistics Mgmt.*, vol. 45, no. 8, pp. 814–830, 2015.
[12] M. Lenzen, "Aggregation versus disaggregation in input–output analysis of the environment," *Econ. Sys. Research*, vol. 23, no. 1, pp. 73–89, 2011.
[13] D. Cole, "The effects of student-faculty interactions on minority students' college grades: Differences between aggregated and disaggregated data." *J. of the Professoriate*, vol. 3, no. 2, pp. 137–160, 2010.
[14] Z. Liu, H. A. Song, V. Zadorozhny, C. Faloutsos, and N. Sidiropoulos, "H-fuse: Efficient fusion of aggregated historical data," in *Proc. of the SIAM Int. Conf. on Data Mining (SDM 2017)*, Houston, Texas, USA, April 2017.
[15] F. M. Almutairi, F. Yang, H. A. Song, C. Faloutsos, N. Sidiropoulos, and V. Zadorozhny, "Homerun: scalable sparse-spectrum reconstruction of aggregated historical data," *Proc. of the VLDB Endowment*, vol. 11, no. 11, pp. 1496–1508, 2018.
[16] N. Rossi *et al.*, "A note on the estimation of disaggregate time series when the aggregate is known," *The Review of Econ. and Stats.*, vol. 64, no. 4, pp. 695–696, 1982.
[17] G. C. Chow and A.-l. Lin, "Best linear unbiased interpolation, distribution, and extrapolation of time series by related series," *The review of Econ. and Stats.*, vol. 53, no. 4, pp. 372–375, 1971.
[18] J. M. Pavía-Miralles, "A survey of methods to interpolate, distribute and extra-polate time series," *J. of Service Science and Mgmt. anagement*, vol. 3, no. 04, p. 449, 2010.
[19] J. M. Pavía-Miralles and B. Cabrer-Borrás, "On estimating contemporaneous quarterly regional gdp," *J. of Forecasting*, vol. 26, no. 3, pp. 155–170, 2007.
[20] T. Di Fonzo, "The estimation of m disaggregate time series when contemporaneous and temporal aggregates are known," *The Rev. of Econ. and Stats.*, vol. 72, no. 1, pp. 178–182, 1990.
[21] F. M. Almutairi, C. I. Kanatsoulis, and N. D. Sidiropoulos, "Tendi: Tensor disaggregation from multiple coarse views," in *Proc. of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2020)*, Singapore, May 2020.
[22] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Journal of Mathematics and Physics*, vol. 6, no. 1-4, pp. 164–189, 1927.
[23] L. Chiantini and G. Ottaviani, "On generic identifiability of 3-tensors of small rank," *SIAM J. on Matrix Analys. and App.*, vol. 33, no. 3, pp. 1018–1037, 2012.

[24] M. Sørensen and L. De Lathauwer, "Fiber sampling approach to canonical polyadic decomposition and application to tensor completion," *SIAM Journal on Matrix Analysis and Applications*, vol. 40, no. 3, pp. 888–917, 2019.
[25] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and M. Akçakaya, "Tensor completion from regular sub-nyquist samples," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1–16, 2019.
[26] M. Ashraphijuo and X. Wang, "Fundamental conditions for low-cp-rank tensor completion," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 2116–2145, 2017.
[27] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.
[28] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
[29] M. Lenzerini, "Data integration: A theoretical perspective," in *Proc. of the 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database sys.*, Madison, Wisconsin, USA, June 2002.
[30] X. L. Dong and F. Naumann, "Data fusion: resolving data conflicts for integration," *Proc. of the VLDB Endowment*, vol. 2, no. 2, pp. 1654–1655, 2009.
[31] C. Faloutsos, H. V. Jagadish, and N. Sidiropoulos, "Recovering information from summary data," *Proc. of the VLDB*, vol. 1, no. 1, pp. 36–45, 1997.
[32] C. Sax and P. Steiner, "Temporal disaggregation of time series," *The R Journal*, vol. 5, no. 2, pp. 80–87, 2013.
[33] F. Yang, F. M. Almutairi, H. A. Song, C. Faloutsos, N. D. Sidiropoulos, and V. Zadorozhny, "Turbolift: fast accuracy lifting for historical data recovery," *The VLDB Journal*, pp. 1–20, 2020.
[34] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *Proc. of the 30th Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain, 2016.
[35] Y. Matsubara, Y. Sakurai, C. Faloutsos, T. Iwata, and M. Yoshikawa, "Fast mining and forecasting of complex time-stamped events," in *Proc. of the 18th ACM SIGKDD*, Beijing, China, 2012.
[36] K. Takeuchi, H. Kashima, and N. Ueda, "Autoregressive tensor factorization for spatio-temporal predictions," in *Proc. of the IEEE Intl. Conf. on Data Mining (ICDM 2017)*, New Orleans, LA, USA, 2017.
[37] F. M. Almutairi, N. D. Sidiropoulos, and G. Karypis, "Context-aware recommendation-based learning analytics using tensor and coupled matrix factorization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 5, pp. 729–741, 2017.
[38] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 2, pp. 16:1–16:44, 2016.
[39] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and W.-K. Ma, "Hyperspectral super-resolution: Combining low rank tensor and matrix structure," in *Proc. of the 25th IEEE International Conference on Image Processing (ICIP 2018)*, Athens, Greece, Oct. 2018.
[40] ——, "Hyperspectral super-resolution via coupled tensor factorization: Identifiability and algorithms," in *Proc. of the IEEE ICASSP*, Calgary, Alberta, Canada, April 2018.
[41] C. I. Kanatsoulis, N. D. Sidiropoulos, M. Akçakaya, and X. Fu, "Regular sampling of tensor signals: Theory and application to fmri," in *Proc. of the IEEE ICASSP*, Brighton, United Kingdom, May 2019.
[42] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and W. Ma, "Hyperspectral super-resolution: A coupled tensor factorization approach," *IEEE T. on Signal Process.*, vol. 66, no. 24, pp. 6503–6517, 2018.
[43] B. D. Williams and M. A. Waller, "Creating order forecasts: point-of-sale or order history?" *J. of Business Logistics*, vol. 31, no. 2, pp. 231–251, 2010.
[44] Y. Jin, B. D. Williams, T. Tokar, and M. A. Waller, "Forecasting with temporally aggregated demand signals in a retail supply chain," *Journal of Business Logistics*, vol. 36, no. 2, pp. 199–211, 2015.
[45] N. Yokoya, T. Yairi, and A. Iwasaki, "Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 2, pp. 528–537, 2011.
[46] M. Simões, J. Bioucas-Dias, L. B. Almeida, and J. Chanussot, "A convex formulation for hyperspectral image superresolution via subspace-based regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3373–3388, 2014.
[47] R. C. Gunning and H. Rossi, *Analytic functions of several complex variables*. American Mathematical Society, 2009, vol. 368.