Outlier-Robust Multi-Aspect Streaming Tensor Completion and Factorization

Mehrnaz Najafi[1*] , **Lifang He**[2] , **Philip S. Yu**[1]
[1]Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA
[2]Department of Biostatistics and Epidemiology, University of Pennsylvania, Philadelphia, PA, USA
{mnajaf2, psyu}@uic.edu, lifanghescut@gmail.com

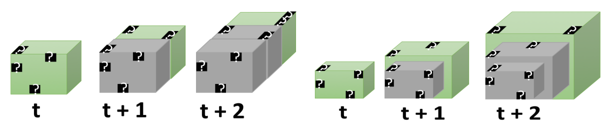
Abstract

With the increasing popularity of streaming tensor data such as videos and audios, tensor factorization and completion have attracted much attention recently in this area. Existing work usually assume that streaming tensors only grow in one mode. However, in many real-world scenarios, tensors may grow in multiple modes (or dimensions), *i.e.*, multi-aspect streaming tensors. Standard streaming methods cannot directly handle this type of data elegantly. Moreover, due to inevitable system errors, data may be contaminated by outliers, which cause significant deviations from real data values and make such research particularly challenging. In this paper, we propose a novel method for Outlier-Robust Multi-Aspect Streaming Tensor Completion and Factorization (OR-MSTC), which is a technique capable of dealing with missing values and outliers in multi-aspect streaming tensor data. The key idea is to decompose the tensor structure into an underlying low-rank clean tensor and a structured-sparse error (outlier) tensor, along with a weighting tensor to mask missing data. We also develop an efficient algorithm to solve the non-convex and non-smooth optimization problem of OR-MSTC. Experimental results on various real-world datasets show the superiority of the proposed method over the baselines and its robustness against outliers.

1 Introduction

Tensors (or multi-way arrays) are higher order generalization of vectors and matrices. They are used to model multi-modal information in many real-world applications including recommendation systems [Karatzoglou *et al.*, 2010], image processing [Lahat *et al.*, 2015], social network analysis [Ermiş *et al.*, 2015] and chemometrics [Mørup and Hansen, 2009]. A fundamental challenge is how to process, analyze and utilize such high-volume tensor data effectively and efficiently. In order to tackle this challenge, many algorithms such as CANDECOMP/PARAFAC (CP) factorization [Kolda and Bader, 2009] and Tucker factorization [Kolda and Bader, 2009] have been developed.

Due to the nature of data or cost of the data collection, the data tensor may be *incomplete* (i.e., tensor with missing values or partially observed tensor). The problem of filling the missing entries of incomplete tensor is called *tensor completion*, which has received much attention from researchers and practitioners in many domains such as recommender systems [Rendle *et al.*, 2009] and image recovery [Liu *et al.*, 2013].



(a) Single-Aspect Streaming Tensor (b) Multi-Aspect Streaming Tensor

Figure 1: Illustration of Single-Aspect Streaming Tensors (traditional settings) and Multi-Aspect Streaming Tensors (t denotes the time-step) with missing data. The grey blocks represent the data at the previous time steps, while green ones represent the data at current time step. Black shapes with question marks are the missing entries.

For many modern applications, data is collected in a streaming fashion carrying time-varying information. For instance, in restaurant recommendation system, data is naturally structured as tensor with three modes (or dimensions), and number of users and restaurants may grow as time goes. In Facebook, users post 684,478 messages per minute. Because of increasingly massive amount of newly created data instances in streaming setting, using static algorithms for tensor completion and factorization is computationally expensive from time and space complexity perspectives. Learning latent factors in a streaming manner has benefit of avoiding irrelevant information and computational overhead of the long-past data. This raises a fundamental challenge for machine learning: how to obtain the complete tensor effectively and efficiently given incomplete streaming tensor.

Existing work on streaming tensor completion and factorization can be roughly classified into two categories: *single-aspect* approach and *multi-aspect* approach. Single-aspect approach [Kasai, 2016; Mardani *et al.*, 2015] is usually based on the assumption that the streaming tensor evolves over one mode, while multi-aspect approach [Song *et al.*, 2017; Nimishakavi *et al.*, 2018] is built upon the fact that the tensor

[*]Contact Author

data may stream across multiple modes. Figure 1 shows an example of single-aspect and multi-aspect streaming tensors with missing values.

Following the first assumption, single-aspect streaming tensor completion and factorization have been extensively studied under the CP factorization. There are many work have been done for this track [Kasai, 2016; Mardani *et al.*, 2015; Zhou *et al.*, 2016]. These approaches iteratively update the latent factors of static modes with partial decomposition of the incremental mode.

Different from single-aspect approach, multi-aspect streaming tensor completion and factorization usually suffer from a high computational and space complexity as multiple modes evolve over time. Also, the incremental part may not be well-structured and therefore cannot be directly factorized. Song et al. [Song *et al.*, 2017] proposed a method, named as MAST, that combines dynamic CP factorization and low-rank tensor completion to fill the missing entries of the incomplete streaming tensor and learn its latent factors without the whole tensor data reconstruction from scratch. MAST uses dynamic tensor factorization to update the model with the incremental part of multi-aspect streaming tensor sequence and track its low-rank subspace via nuclear norm for completion purpose. By leveraging the learned model in the previous time step, MAST accelerates filling the missing entries, while having comparable performance to traditional methods.

A challenging problem may arise when the incomplete tensor is contaminated by *outliers*. Outliers can be defined as significant deviations from real data values. Due to sensor failures, malicious tampering and system errors, real-world data can encounter outliers [Balcan and Zhang, 2016]. With outliers, although errors exist only on several parts of data, they can adversarially affect learning from data. Existing streaming tensor completion and factorization methods concentrate only on clean multi-aspect streaming tensors or outlier-corrupted single-aspect streaming tensors, which leads to the problem of outlier-contaminated streaming tensor completion and factorization. Hawkins et al. [Zhang and Hawkins, 2018] developed a method, named as BRST, for erroneous single-aspect streaming tensor via variational Bayesian inference. BRST models the whole single-aspect erroneous streaming tensor as the sum of a low-rank streaming tensor and a time-varying sparse component. To learn these two parts, bayesian statistical model is applied. This model imposes low-rank and sparsity using the hyperparameters and prior density functions. Then, the posterior density function of the latent factors is obtained by variational Bayesian method.

Nonetheless, BRST cannot be directly applied to multi-aspect erroneous streaming tensors. To handle outliers in incomplete multi-aspect streaming tensors, we propose a novel Outlier-Robust Multi-Aspect Streaming Tensor Completion and Factorization (OR-MSTC) method based on CP. Different from MAST, OR-MSTC decomposes the data tensor into a low-rank clean tensor and an error tensor, and imposes $\ell_{2,1}$ on the error tensor, aiming to learn better latent factors, while improving the robustness of completion and factorization against outliers. A clean tensor is the low-rank tensor without outlier for an outliers-corrupted tensor.

Symbol	Definition and description
$[1:M]$	set of integers in range of 1 to M inclusively
x	lowercase letter represents a scale
$\mathbf{x}$	boldface lowercase letter represents a vector
$\mathbf{X}$	boldface uppercase letter represents a matrix
$\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$	each calligraphic letter represents a tensor
$\langle \cdot, \cdot \rangle$	inner product
$\circ$	tensor product (outer product)
$*$	Hadamard (element-wise) product
$\odot$	Khatri-Rao product
$\mathbf{X}_{(n)}$	$\mathbf{A}_n(\mathbf{A}_N \odot \ldots \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \ldots \odot \mathbf{A}_1)^\top$
$(\mathbf{A}_k)^{\odot k \neq n}$	$\mathbf{A}_N \odot \ldots \odot \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \odot \ldots \odot \mathbf{A}_1$
$(\mathbf{A}_k)^{* k \neq n}$	$\mathbf{A}_N * \ldots * \mathbf{A}_{n+1} * \mathbf{A}_{n-1} * \ldots * \mathbf{A}_1$
$(\mathbf{a}_n)_{i_n}$	i_n^{th} row of $\mathbf{A}_n$
$\lvert\cdot\rvert$	denotes absolute value
$\lVert\mathcal{X}\rVert_F = \sqrt{\sum_{i,j,k} \lvert x_{i,j,k}\rvert^2}$	(Frobenius) norm of a 3^{rd} order tensor
$\lVert\mathcal{X}\rVert_{2,1} = \sum_j \lVert\mathcal{X}(:j,:)\rVert_F$	$\ell_{2,1}$ norm of a 3^{rd} order tensor
$\lVert\mathbf{X}\rVert_*$	sum of the singular values of $\mathbf{X}$
$\lVert\mathcal{X}\rVert_*$	convex envelop of tensor avg. rank within unit ball

Table 1: List of basic symbols

The error tensor captures the outliers in the tensor data, while the clean low-rank tensor encodes the underlying low-rank structure of the streaming tensor data. Without loss of generality, we assume that outliers are distributed on the 2^{nd} mode of the tensor. We can hence use $\ell_{2,1}$ to characterize this sparsity property. The proposed OR-MSTC method imposes $\ell_{2,1}$ norm on error tensor. The decomposition into two parts and applying $\ell_{2,1}$ norm make the corresponding objective function of the proposed OR-MSTC method challenging to optimize. We present a new efficient optimization algorithm based on Alternating Direction Method of Multipliers (ADMM) [Boyd *et al.*, 2011] to solve it. Our main contributions can be summarized as follows:

- To the best of our knowledge, OR-MSTC is the first work that recovers clean tensor under low-rank subspace, while capturing outliers for outlier-contaminated multi-aspect streaming tensors.
- OR-MSTC is the first method based on CP factorization that isolates the clean low-rank tensor from outliers for an outlier-corrupted multi-aspect streaming tensors.
- We propose a new efficient optimization algorithm based on ADMM to solve the objective function.
- Through extensive experiments on real-world datasets, we show that OR-MSTC is superior to several state-of-the-art methods in tensor completion and factorization and robust against outliers.

2 Outlier-Robust Multi-Aspect Streaming Tensor Completion and Factorization

We begin with some necessary notations and concepts of tensor algebra. Table 1 lists basic symbols that will be used throughout the paper. Specifically, an element of a vector $\mathbf{x}$, a matrix $\mathbf{X}$, or a tensor $\mathcal{X}$ is denoted by x_i, $x_{i,j}$, $x_{i,j,k}$ etc., depending on the number of modes.

Given an N^{th} order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and an integer R (rank), the CP factorization is defined by latent factor matrices $\mathbf{X}_n \in \mathbb{R}^{I_n \times R}$ for $n \in [1 : N]$, respectively, such that $\mathcal{X} = \sum_{r=1}^{R} \mathbf{x}_1^r \circ \mathbf{x}_2^r \circ \cdots \circ \mathbf{x}_N^r = [\![\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_N]\!]$, where $\mathbf{x}_n^r \in \mathbb{R}^{I_n}$ is the r^{th} column of the latent factor matrix $\mathbf{X}_n$, and $[\![\cdot]\!]$ is used for shorthand notation of the sum of rank-one tensors. For convenience, in the following, we write $\mathbf{x}_1 \circ \mathbf{x}_2 \circ \cdots \circ \mathbf{x}_N$ as $\prod_{n=1}^{N} \circ \mathbf{x}_n$.

We will first systematically propose a novel outlier-robust streaming tensor completion and factorization method, followed by a new efficient iterative algorithm to solve the formulated objective function.

Given a multi-aspect streaming N^{th} order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with missing entries and possibly outliers, where $\mathcal{X}^{(t)}$ is the snapshot of the tensor at time-step t and for any $t \in \mathbb{Z}^+$, $\mathcal{X}^{(t-1)}$ is a sub-tensor of $\mathcal{X}^{(t)}$, the goal is to recover the missing and corrupted entries, and learn its latent factors.

2.1 Outlier-Robust Low-Rank Tensor Completion

The low-rank tensor completion aims at learning the low-rank complete tensor given an incomplete tensor, can be formulated as a *rank-minimization* problem as follows:

$$\min_{\mathcal{X}^c} rank(\mathcal{X}^c) \quad \text{s.t.} \quad \Omega * \mathcal{X}^c = \mathcal{X} \tag{1}$$

where $\mathcal{X}^c \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ denotes the complete low-rank tensor, $\mathcal{X}$ indicates the incomplete tensor which is a partial observations of $\mathcal{X}^c$. Ω is a binary tensor with the same size as $\mathcal{X}^c$ whose entries indicate whether each corresponding entry in $\mathcal{X}^c$ is observed or not, where $\omega_{i_1,\cdots,i_N} = 1$ if $x^c_{i_1,\cdots,i_N}$ is observed and $\omega_{i_1,\cdots,i_N} = 0$ otherwise. $*$ denotes the element-wise product. Since $rank(\mathcal{X}^c)$ is non-convex, the objective function in Eq. (1) is an NP-hard problem [Håstad, 1990]. One way is to replace $rank(\mathcal{X}^c)$ with the summation of the nuclear norm of its factorized matrices and factorization error as follows [Liu *et al.*, 2013]:

$$\min_{\mathcal{X}^c, \mathbf{A}_1,\ldots,\mathbf{A}_N} \left\|\mathcal{X}^c - [\![\mathbf{A}_1, ..., \mathbf{A}_N]\!]\right\|_F^2 + \sum_{n=1}^{N} \alpha_n \left\|\mathbf{A}_n\right\|_* \quad \text{s.t.} \quad \Omega * \mathcal{X}^c = \mathcal{X} \tag{2}$$

where α_n are hyperparameters that balance impact of each mode.

$\mathcal{X}^c$ might be also erroneous so that some of its entries are perturbed. As shown in Figure 2, we assume that the complete N^{th} order tensor $\mathcal{X}^c$ is a mixture of clean low-rank tensor $\mathcal{L}$ and error tensor $\mathcal{E}$ that captures sparse outliers in $\mathcal{X}^c$. For example, in network traffic, the clean low-rank tensor is the usual traffic flow and outliers are anomalies [Zhang and Hawkins, 2018]. This decomposition technique is called *low-rank tensor decomposition* and is formulated as $\mathcal{X}^c = \mathcal{L} + \mathcal{E}$.

Without loss of generality, our assumption is that outliers are distributed on 2^{nd} dimension of tensor $\mathcal{X}^c$ (and thus $\mathcal{X}$) and they are very sparse in comparison to the data size. We can thus impose $\ell_{2,1}$ on the error tensor to characterize this sparsity property, which is defined as $\left\|\mathcal{E}\right\|_{2,1}$. Our proposed method can be easily extended to deal with multi-way outliers. Specially, similar to 2^{nd} mode, we can make similar consideration and formulation for other modes. It can be seen that the sparsity-inducing property of $\ell_{2,1}$-norm pushes $\mathcal{E}$ to be sparse in each slice along 2^{nd} mode. More specifically, 2^{nd} mode shrinks to zero if the corresponding entries belongs to outliers. Using this decomposition, tensor completion is a rank minimization problem that can be approximated using tensor nuclear norm. The objective function for low-rank clean tensor decomposition can be stated as follows:

$$\min_{\mathcal{L},\mathcal{E}} \left\|\mathcal{L}\right\|_* + \lambda \left\|\mathcal{E}\right\|_{2,1} \quad \text{s.t.} \quad \mathcal{X}^c = \mathcal{L} + \mathcal{E} \tag{3}$$

where $\left\|\mathcal{L}\right\|_*$ is the convex envelop of the tensor average rank within the unit ball [Lu *et al.*, 2016] and λ is the hyperparameter to balance impact of the error tensor.

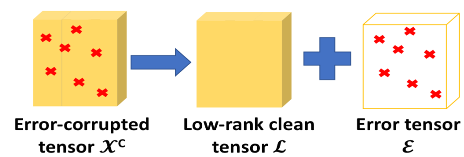


Figure 2: Low-rank decomposition of error-corrupted tensor, where error could exhibit as outliers, into the clean low-rank tensor and error tensor.

By replacing tensor nuclear norm minimization with Eq. (2), Eq. (3) can be written as follows:

$$\min_{\mathcal{X}^c,\mathcal{E},\mathbf{A}_1,\ldots,\mathbf{A}_N} \left\|\mathcal{X}^c - \mathcal{E} - [\![\mathbf{A}_1, ..., \mathbf{A}_N]\!]\right\|_F^2 + \sum_{n=1}^{N} \alpha_n \left\|\mathbf{A}_n\right\|_* + \lambda \left\|\mathcal{E}\right\|_{2,1} \quad \text{s.t.} \quad \Omega * \mathcal{X}^c = \mathcal{X} \tag{4}$$

We refer to the above loss as L_{ORLTC}. Due to outliers and missing entries in $\mathcal{X}$, latent factors $\mathbf{A}_1, \mathbf{A}_2, ...,$ and $\mathbf{A}_N$ may not be exact. Thus, instead of using binary tensor Ω in Eq. (2), we define weighting tensor $\mathcal{W}$ with the same size as $\mathcal{X}^c$ and follow an iterative approach to obtain better latent factors by taking incomplete nature of data into account as follows:

$$w_{i_1,\ldots,i_N} = \begin{cases} 0 & \text{if } \Omega_{i_1,\ldots,i_N} = 0, \\ val & \text{otherwise} \end{cases} \tag{5}$$

The weighting tensor $\mathcal{W}$ represents the reliability of the entries in tensor $\mathcal{X}$. If a tensor entry is not missing, we set initial weight to not be 0. because that entry might be contaminated by outlier, which is set to 0.001 in our experiments. Otherwise, the weight of entry is set to be 0 as missing entry is not reliable. Having latent factors $\mathbf{A}_1, \mathbf{A}_2, ...,$ and $\mathbf{A}_N$, the goal is to keep the most reliable entries in $\mathcal{X}$, while iteratively updating those that have very low reliability. Inspired by [Shao *et al.*, 2015], we use $\mathcal{W}$ to update $\mathcal{X}^c$ as follows:

$$\mathcal{X}^c = \mathcal{W} * \mathcal{X}^c + (\mathbf{1} - \mathcal{W}) * \hat{\mathcal{X}}^c \ \text{s.t.}\ \hat{\mathcal{X}}^c = [\![\mathbf{A}_1, ..., \mathbf{A}_N]\!] \tag{6}$$

where $\mathbf{1}$ is a tensor with all ones. As the iteration goes up, more weight is assigned to less reliable entries. We then revise the weight tensor at the end of each iteration as follows:

$$\mathcal{W} = \mathbf{1} - (\boldsymbol{\beta} \times (\mathbf{1} - \mathcal{W})) \tag{7}$$

Algorithm 1 Outlier-Robust Streaming Tensor Completion and Factorization (OR-MSTC)

Input: $\mathcal{X}, \Omega$

Parameter: $\{\tilde{\mathbf{A}_n}\}_{n=1}^N, R, \{\alpha_n\}_{n=1}^N, \beta, \mu, \lambda, \eta, \eta_{max}, \rho, tol$

Output: $\mathcal{X}^c, \{\mathbf{A}_n\}_{n=1}^N, \mathcal{E}$

$\mathbf{A}_n^{(0)} = \tilde{\mathbf{A}_n}, \mathbf{A}_n^{(1)} = rand(d_n, R), \mathbf{Z}_n = \mathbf{Y}_n = 0$ where d_n denotes the size of incremental part in mode n

repeat

  $\eta = \min(\eta * \rho, \eta_{max})$

  for $n = 1$ **to** N **do**

    Update $\mathbf{A}_n^{(0)}$ and $\mathbf{A}_n^{(1)}$ using Eq. (13)

  end for

  Update $\mathcal{X}^c$ using Eq. (6)

  Update $\mathcal{W}$ using Eq. (7)

  Update $\mathcal{E} = \text{prox\_l21}_\lambda([\![\mathbf{A}_1, ..., \mathbf{A}_N]\!] - \mathcal{X}^c)$

  for $n = 1$ **to** N **do**

    $\mathbf{Y}_n = \mathbf{Y}_n + \eta(\mathbf{Z}_n - \mathbf{A}_n)$

    $\mathbf{Z}_n = SVT_{\frac{\alpha_n}{\eta}}(\mathbf{A}_n - \frac{\mathbf{Y}_n}{\eta})$

  end for

until $\frac{\|\mathcal{X}^c_{pre} - \mathcal{X}^c\|_F}{\|\mathcal{X}^c_{pre}\|_F} < tol$

where β is a decay parameter, which we fix to 0.97 in our experiments. As the number of iterations increases, $\mathcal{W}$ converges to 1. The resulted objective function can be formulated as $\min_{\mathcal{X}^c, \mathcal{E}, \mathbf{A}_1, ..., \mathbf{A}_N} L_{ORLTC}$, where $\mathcal{X}^c$ is updated using Eqs. (6) and (7).

2.2 Dynamic CP Factorization

In streaming settings, a dynamic CP method obtains latent factors in an efficient and effective fashion. We use $\tilde{\mathcal{X}}$ and $\mathcal{X}$ to represent two consecutive snapshots $\mathcal{X}^{(t-1)}$ (tensor at time step $t-1$) and $\mathcal{X}^{(t)}$ (tensor at time step t) such that $\tilde{\mathcal{X}} \subsetneq \mathcal{X}$. The general CP factorization objective function for tensor $\mathcal{X}$ is defined as $\min_{\mathbf{A}_1, ..., \mathbf{A}_N} \|\mathcal{X} - [\![\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_N]\!]\|_F^2$.

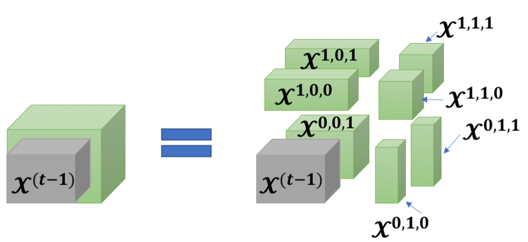


Figure 3: Subtensors created by partitioning of incremental part (superscript tuples with the same size as N are used to refer to the subtensors of N^{th} order $\mathcal{X}$).

In CP factorization for $\tilde{\mathcal{X}}$, we have $\tilde{\mathcal{X}} \approx [\![\tilde{\mathbf{A}_1}, \tilde{\mathbf{A}_2}, ..., \tilde{\mathbf{A}_N}]\!]$ ($\tilde{\mathbf{A}_1}, \tilde{\mathbf{A}_2}, ...,$ and $\tilde{\mathbf{A}_N}$ denote latent factors for the former snapshot). We take advantage of former snapshot latent factors to learn latent factors for $\mathcal{X}$ by incorporating $\tilde{\mathcal{X}}$ into CP factorization of $\mathcal{X}$. This in fact accelerates factorization of current snapshot. Dynamic CP factorization for $\mathcal{X}$ can be hence rewritten as summation of two terms: factorization related to former snapshot and factorization for the incremental part *i.e.*, part in snapshot at time step t that didn't exist in snapshot at time step $t-1$. The resulted objective function for $\mathcal{X}$ can be stated as follows (θ indicates entries of the incremental part):

$$\min_{\mathbf{A}_1, ..., \mathbf{A}_N} \left\|\tilde{\mathcal{X}} - [\![\mathbf{A}_1^{(0)}, ..., \mathbf{A}_N^{(0)}]\!]\right\|_F^2 + \sum_{i_1, i_2, ..., i_N \notin \theta} (\|\mathbf{X}_{i_1, ..., i_N} - [\![(\mathbf{a}_1)_{i_1}, ..., (\mathbf{a}_N)_{i_N}]\!]\|_F^2) \tag{8}$$

where $\mathbf{A}_n^\top = [\mathbf{A}_n^{(0)\top}, \mathbf{A}_n^{(1)\top}]$. Let d_n denote increment along n^{th} mode at time step t in comparison to time step $t-1$. $\mathbf{A}_n^{(0)} \in \mathbb{R}^{(I_n - d_n) \times R}$ refers to partition of the latent factor w.r.t. tensor snapshot at time step $t-1$ in the tensor snapshot at time step t, while $\mathbf{A}_n^{(1)} \in \mathbb{R}^{d_n \times R}$ is for incremental part. Using CP factorization for $\tilde{\mathcal{X}}$ in Eq. (8) results in the following objective function:

$$\min_{\mathbf{A}_1, ..., \mathbf{A}_N} \mu \left\|[\![\tilde{\mathbf{A}_1}, \tilde{\mathbf{A}_2}, ..., \tilde{\mathbf{A}_N}]\!] - [\![\mathbf{A}_1^{(0)}, ..., \mathbf{A}_N^{(0)}]\!]\right\|_F^2 + \sum_{i_1, i_2, ..., i_N \notin \theta} (\|\mathbf{X}_{i_1, ..., i_N} - [\![(\mathbf{a}_1)_{i_1}, ..., (\mathbf{a}_N)_{i_N}]\!]\|_F^2) \tag{9}$$

where $\mu \in [0, 1]$ is a forgetting factor to control trade-off between previous and current factorizations for time step $t-1$. As shown in Figure 3, second term can be defined as a summation of CP factorization for a set of sub-tensors [Song *et al.*, 2017].

2.3 Problem Formulation

We combine outlier-robust low-rank tensor completion and dynamic CP factorization as follows:

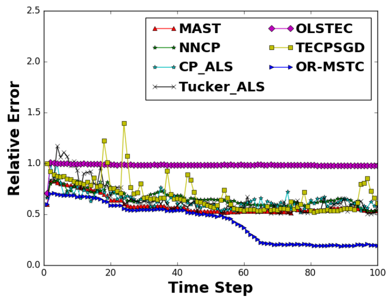
$$\min_{\mathcal{X}^c, \mathcal{E}, \mathbf{A}_1, ..., \mathbf{A}_N} \mu \left\|[\![\tilde{\mathbf{A}_1}, ..., \tilde{\mathbf{A}_N}]\!] - [\![\mathbf{A}_1^{(0)}, ..., \mathbf{A}_N^{(0)}]\!]\right\|_F^2 + \sum_{i_1, i_2, ..., i_N \notin \theta} (\|(\mathbf{X}^c - \mathbf{E})_{i_1, ..., i_N} - [\![(\mathbf{a}_1)_{i_1}, ..., (\mathbf{a}_N)_{i_N}]\!]\|_F^2 + \sum_{n=1}^N \alpha_n \|\mathbf{A}_n\|_* + \lambda \|\mathcal{E}\|_{2,1} = \min_{\mathcal{X}^c, \mathcal{E}, \mathbf{A}_1, ..., \mathbf{A}_N} L_{ORDLTC} \tag{10}$$

where $\mathcal{X}^c$ is updated using Eqs. (6) and (7). In the second term, we use $\mathcal{X}^c - \mathcal{E}$ as substitution of $\mathcal{X}$.

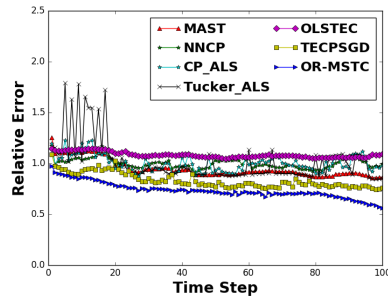
2.4 Optimization Procedure

Using ADMM, by inducing $\mathbf{Z}_n^\top = [\mathbf{Z}_n^{(0)\top}, \mathbf{Z}_n^{(1)\top}] \in \mathbb{R}^{I_n \times R}$ and Lagrange multipliers $\mathbf{Y}_n^\top = [\mathbf{Y}_n^{(0)\top}, \mathbf{Y}_n^{(1)\top}] \in \mathbb{R}^{I_n \times R}$, the augmented Lagrangian function for Eq. (10) is as follows:

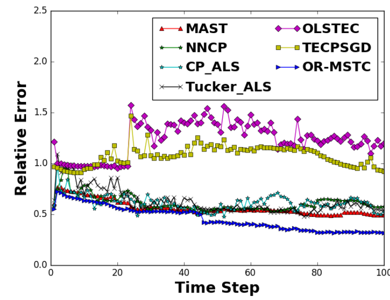
$$\min_{\mathcal{X}^c, \mathcal{E}, \{\mathbf{A}_n\}, \{\mathbf{Z}_n\}, \{\mathbf{Y}_n\}} L_{ORDLTC} + \sum_{n=1}^N (\alpha_n \|\mathbf{Z}_n\|_* + \langle \mathbf{Y}_n, \mathbf{Z}_n - \mathbf{A}_n \rangle + \frac{\eta}{2} \|\mathbf{Z}_n - \mathbf{A}_n\|_F^2) \quad \text{s.t. } \mathbf{Z}_n = \mathbf{A}_n \tag{11}$$



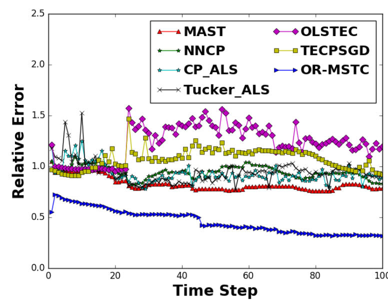
(a) NT-M10-E10



(b) NT-M10-E20



(c) NT-M20-E10



(d) NT-M20-E20

Figure 4: Performance of the methods, where MX denotes missing percentage = X% and EX indicates outlier percentage =X%. For example, NT-M10-E20 denotes network traffic dataset with missing percentage 10% and outlier percentage 20%.

Dataset		CMRI				Yelp				Network Traffic			
	Missing	10%		20%		10%		20%		10%		20%	
	Outlier	10%	20%	10%	20%	10%	20%	10%	20%	10%	20%	10%	20%
Static	CP-ALS	6.74	11.29	6.79	12.80	661.41	567.05	604.09	790.00	1.26	1.37	1.25	1.23
	Tucker-ALS	0.21	0.21	0.23	0.27	107.61	150.78	274.76	332.87	2.99	2.46	5.03	3.90
	NNCP	2.12	4.23	2.50	4.54	273.70	309.40	327.18	457.00	4.81	2.62	3.65	1.96
Dynamic	OLSTEC	1.20	1.06	1.10	1.08	0.11	0.38	0.13	0.65	0.18	0.20	0.19	0.29
	TeCPSGD	0.72	0.92	0.95	0.87	0.06	0.22	0.09	0.43	0.24	0.22	0.23	0.38
	MAST	0.46	0.32	0.43	0.51	17.21	20.00	16.95	23.00	0.60	0.60	1.26	0.72
	BRST	3.04	2.47	3.04	2.62	NA	NA	NA	NA	10.76	11.12	9.03	6.69
	OR-MSTC	0.20	0.20	0.19	0.23	21.01	20.00	25.01	22.00	0.40	0.34	0.82	0.33

Table 2: Average running time of different methods on the datasets

where η is penalty parameter and $\mathcal{X}^c$ is updated by Eqs. (6) and (7). Let $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ denote the following ($\mathcal{L} = \mathcal{X}^c - \mathcal{E}$):

$$\mathbf{P}_1 = \tilde{\mathbf{A}_n}[(\tilde{\mathbf{A}_k}^\top \mathbf{A}_k^{(0)})^{*k \neq n}], \mathbf{P}_3 = \sum_{i \in S_n^0} (\mathbf{L})^i_{(n)} ((\mathbf{a}_k)_{i_k})^{\odot k \neq n}$$

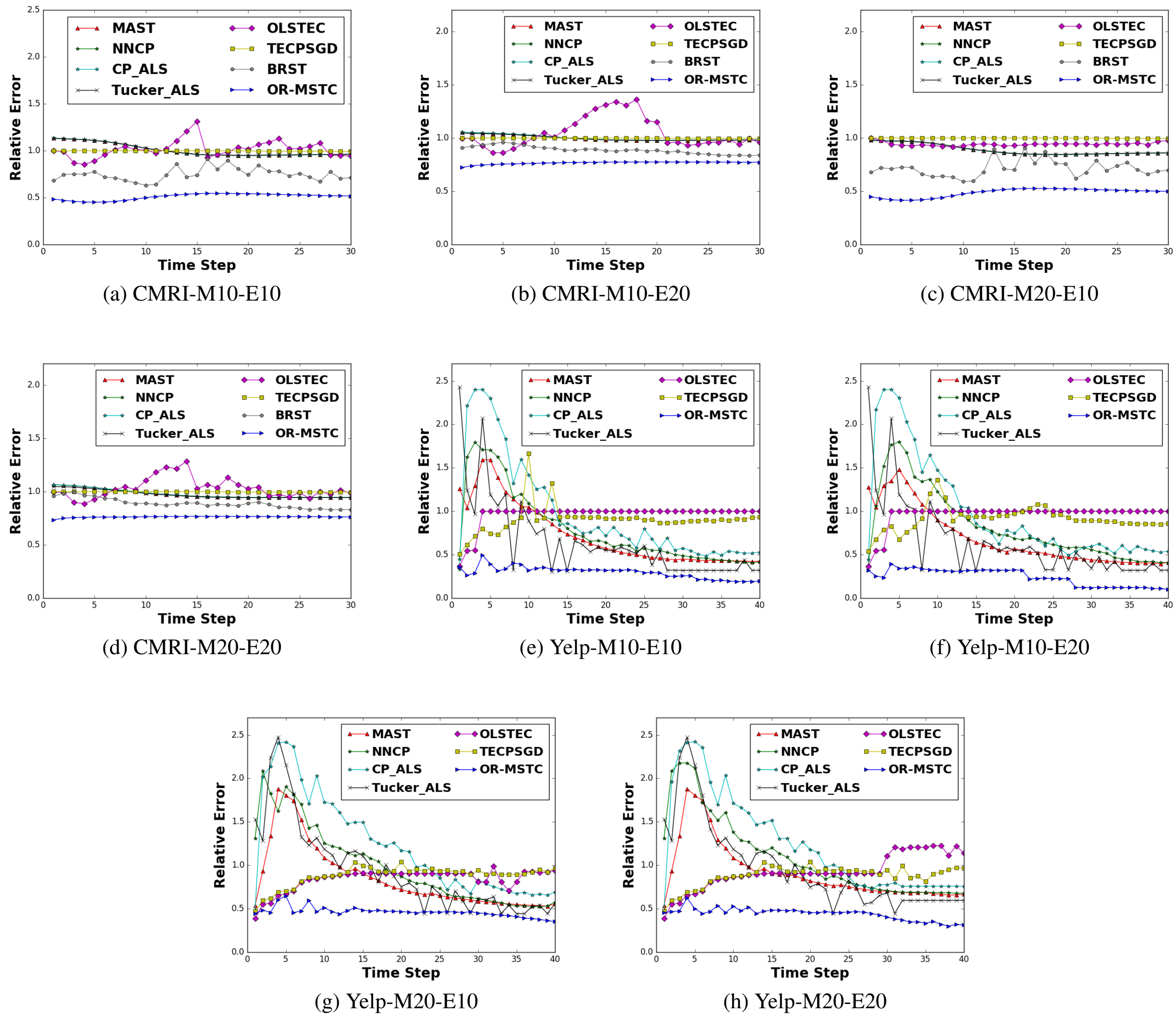
$$\mathbf{P}_2 = (\mathbf{A}_k^{(0)\top} \mathbf{A}_k^{(0)} + \mathbf{A}_k^{(1)\top} \mathbf{A}_k^{(1)})^{*k \neq n} \quad (12)$$

The update rules for $\mathbf{A}_n^{(0)}$ and $\mathbf{A}_n^{(1)}$ are as follows:

$$\mathbf{A}_n^{(0)} = \frac{\mu \mathbf{P}_1 + \eta \mathbf{Z}_n^{(0)} + \mathbf{Y}_n^{(0)} + \mathbf{P}_3}{\mathbf{P}_2 - (1-\mu)(\mathbf{A}_k^{(0)\top} \mathbf{A}_k^{(0)})^{*k \neq n} + \eta \mathbf{I}_R} \quad (13)$$

$$\mathbf{A}_n^{(1)} = \frac{\sum_{i \in S_n^1} (\mathbf{L})^i_{(n)} ((\mathbf{a}_k)_{i_k})^{\odot k \neq n} + \eta \mathbf{Z}_n^{(1)} + \mathbf{Y}_n^{(1)}}{\mathbf{P}_2 + \eta \mathbf{I}_R}$$

where $S_n^0 = \{(s_1, ..., s_N) | \sum_{k=1}^N s_k \neq 0, s_k \in \{0,1\}, s_n = 0\}$ and $S_n^1 = \{(s_1, ..., s_N) | \forall k \in [1, ..., N], s_k \in \{0,1\}, s_n = 1\}$ are sets of tuples to refer to subtensors. $\mathbf{Z}_n$ can be updated as $\mathbf{Z}_n = SVT_{\frac{\alpha_n}{\eta}}(\mathbf{A}_n - \frac{\mathbf{Y}_n}{\eta}), n = 1, 2, ..., N$, where $SVT_{\frac{\alpha_n}{\eta}}$ is the singular value thresholding operator defined as $SVT_\omega(\mathbf{A}) = \mathbf{U}(diag\{\sigma_i - \omega\})_+ \mathbf{V}^\top$, where $\mathbf{U}(diag\{\sigma_i\})_{1<=i<=r} \mathbf{V}^\top$ is the singular value decomposition of $\mathbf{A}$. $\mathbf{X}_+ = max\{\mathbf{X}, 0\}$, where $max\{.,.\}$ is an element-wise operator. The update rule for $\mathcal{E}$ can be stated as $\mathcal{E} = \text{prox\_l21}_\lambda([\![\mathbf{A}_1, ..., \mathbf{A}_N]\!] - \mathcal{X}^c)$ for incremental part. For 3^{rd} order tensor $\mathcal{R}$, $\text{prox\_l21}_\lambda(\mathcal{R})$ approximates $\mathcal{R}^i = \frac{||\mathcal{R}^i||_F - \lambda}{||\mathcal{R}^i||_F} \mathcal{R}^i$, if $||\mathcal{R}^i||_F > \lambda$; 0, otherwise ($\mathcal{R}^i = \mathcal{R}(:, i, :)$). The framework of OR-MSTC is summarized in Algorithm 1. The convergence condition is that the relative changing of tensor $\mathcal{X}^c$ in two consecutive iterations is smaller than the tolerance.



(a) CMRI-M10-E10 (b) CMRI-M10-E20 (c) CMRI-M20-E10 (d) CMRI-M20-E20 (e) Yelp-M10-E10 (f) Yelp-M10-E20 (g) Yelp-M20-E10 (h) Yelp-M20-E20

Figure 5: Performance of the methods, where MX denotes missing percentage = X% and EX indicates outlier percentage =X%.

3 Experimental Evaluation

To evaluate the performance of the proposed OR-MSTC method, we conduct experiments on real-world datasets and compare with the following seven baselines: (1) **Static NNCP** is based on trace norm constrains and optimized by ADMM [Liu *et al.*, 2015]. (2) **Static CP-ALS** employs CP factorization and optimized by Alternating Least Square (ALS) [Bro, 1998]. (3) **Static Tucker-ALS** is based on tucker factorization and optimized by ALS [Comon *et al.*, 2009]. For all static baselines, the former-step factors is given as the initialization for the current step to improve learning. (4) **OLSTEC** is a single-aspect streaming tensor completion method optimized by ALS [Kasai, 2016]. (5) **TeCPSGD** that is pioneering CP factorization work for single-aspect streaming tensor completion and optimized by stochastic gradient descent [Mardani *et al.*, 2015]. (6) **MAST** is a multi-aspect streaming tensor completion approach that assumes no error in the data [Song *et al.*, 2017]. (7) **BRST** is a error-robust single-aspect streaming tensor completion approach based on bayesian inference [Zhang and Hawkins, 2018].

Similar to [Song *et al.*, 2017], in order to apply single-aspect streaming tensor methods on multi-aspect streaming tensor, we divide the multi-aspect incremental parts into several single-aspect streaming ones and update them in random order in each iteration until convergence.

We apply grid search to identify optimal values for each hyperparameter from $\{10^{-9}, 10^{-8}, ..., 10^{8}, 10^{9}\}$. The tolerance rate is set to 10^{-4}, the maximum number of iterations to 500 for all the methods. The rank is tuned using 10 ranks varying from 5 to 40 based on relative error which we define later. In OR-MSTC and MAST, $\alpha_n = \frac{1}{10N}$, $n = 1, ..., N$, $\eta = 10^{-4}$, $\rho = 1.05$ and $\eta_{max} = 10^6$. We tuned forgetting

	CMRI	Yelp	Network Traffic
Modes	pixel, pixel, time	user, business, time	node, node, time
Stream Type	Single-Aspect	Multi-Aspect	Single-Aspect
Tensor Size	$128 \times 128 \times 35$	$820 \times 820 \times 82$	$11 \times 11 \times 110$
Starting Size	$128 \times 128 \times 5$	$20 \times 20 \times 2$	$11 \times 11 \times 10$
Inc. Step	0, 0, 1	20, 20, 2	0, 0, 1

Table 3: Summary of datasets used in the paper

factor μ in our method and MAST based on the missing percentage. Following [Song *et al.*, 2017], the initial completion and warm start matrices are calculated using NNCP method. We use **Relative Error** (RE) as performance evaluation metric, which is defined as follows:

$$RE = \frac{\|reconstructed\_tensor - real\_tensor\|_2}{\|real\_tensor\|_2} \quad (14)$$

where $reconstructed\_tensor$ denotes the output complete tensor produced by the methods and $real\_tensor$ indicates the true data tensor (i.e., ground truth). RE measures the deviation from the ground truth. The efficiency is measured by **Average Running Time** (ART) and defined as $\frac{1}{T}\sum_{t=1}^{T} RT_t$, where RT_t is the running time at time step t. Each experiment is repeated for 5 times, and mean of the metric in each dataset is reported. The datasets used in the paper are Cardiac MRI (CMRI) [Sharif and Bresler, 2007], downsampled Yelp [Jeon *et al.*, 2016] and Network Traffic (NT) [Lakhina *et al.*, 2004], and summarized in Table 3. Following [Song *et al.*, 2017], we randomly cover a percentage of data (**missing percentage**) ($\{10\%, 20\%\}$) and consider the remaining entries as observed information. We generate outliers on 2^{nd} mode with mean of 1^{st} slice of tensor data, magnitude of 5, variance of 0 and **outlier percentage** of $\{10\%, 20\%\}$.

Figure 4 and 5 show performance of the methods with various missing and outlier percentages. We don't report results for BRST on Yelp and NT as its RE is very large in comparison to the proposed OR-MSTC method and other baselines. On all datasets, OR-MSTC outperforms the dynamic baselines, especially BRST, error-robust streaming scheme, with varying percentages of missing entries and outliers. On CMRI and Yelp, OR-MSTC is significantly better than dynamic baselines (in average at least 0.4 less). In Yelp, number of businesses consistently increases as time evolves. This pattern reduces the quantity of substitution on corrupted and missing entries and results in noticeable improvement in performance of the proposed method. On NT, OR-MSTC always obtains nearly perfect result. The proposed OR-MSTC method shows strong stability in performance on all datasets as time goes. On Yelp, MAST, TeCPSGD and static methods suffer from high instability in performance. This is because incremental part at first 15 time steps contains high number of outlier and missing entries, and they account for most of relative errors in baselines. For various ratios of missing entries and outliers, OR-MSTC outperforms static baselines. In Table 2, OR-MSTC's ART is far better than previous error robust streaming scheme (BRST), but not necessarily better than the other streaming baselines.

On CMRI, performance of all methods is unaffected by varying ratio of missing data with the same outliers, while this observation is not true on varying ratios of outliers with the same missing data. Nevertheless, the proposed OR-MSTC method is more effective and robust with different ratios of missing data and outliers. Yelp is sparse in comparison to CMRI, and adding outliers and missing data does not therefore affect the dataset a lot. As ratio of missing data increases, performance of all methods decreases, but this drop is not high. Compared to baselines, the proposed OR-MSTC method shows stable performance. On NT, TeCPSGD and OLSTEC have large variation and drop in performance with increase in ratio of outliers and missing data compared to our method. They thus don't identify anomalies in network well.

4 Conclusion

We developed a method named OR-MSTC for multi-aspect streaming tensor completion and factorization. OR-MSTC has several advantages over existing streaming tensor completion and factorization methods. First, it effectively captures low-rank subspace of multi-aspect streaming tensor for completion and factorization tasks. Second, it handles outliers well. Third, iterative scalable optimization framework is proposed for OR-MSTC. Compared to existing streaming tensor completion and factorization methods, OR-MSTC showed better performance on three datasets.

Acknowledgements

This work is supported in part by NSF through grants IIS-1526499, IIS-1763325, CNS-1626432, Guangdong Province through grant 2017A030313339 and NSFC 61672313.

References

[Balcan and Zhang, 2016] Maria-Florina Balcan and Hongyang Zhang. Noise-tolerant life-long matrix completion via adaptive sampling. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2963–2971. Curran Associates Inc., 2016.

[Boyd *et al.*, 2011] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[Bro, 1998] Rasmus Bro. *Multi-way analysis in the food industry: models, algorithms, and applications*. PhD thesis, 1998.

[Comon *et al.*, 2009] Pierre Comon, Xavier Luciani, and André de Almeida. Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics*, 23(7):393–405, 2009.

[Ermiş *et al.*, 2015] Beyza Ermiş, Evrim Acar, and A. Taylan Cemgil. Link prediction in heterogeneous data via generalized coupled tensor factorization. *Data Mining and Knowledge Discovery*, 29(1):203–236, January 2015.

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

[Jeon *et al.*, 2016] ByungSoo Jeon, Inah Jeon, Lee Sael, and U Kang. Scout: Scalable coupled matrix-tensor factorization - algorithm and discoveries. In *IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 811–822, May 2016.

[Karatzoglou *et al.*, 2010] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86. ACM, 2010.

[Kasai, 2016] Hiroyuki Kasai. Online low-rank tensor subspace tracking from incomplete data by cp decomposition using recursive least squares. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2519–2523, 2016.

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

[Lahat *et al.*, 2015] Dana Lahat, Tülay Adali, and Christian Jutten. Multimodal data fusion: An overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103:1449–1477, 2015.

[Lakhina *et al.*, 2004] Anukool Lakhina, Konstantina Papagiannaki, Mark Crovella, Christophe Diot, Eric D. Kolaczyk, and Nina Taft. Structural analysis of network traffic flows. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, pages 61–72. ACM, 2004.

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

[Liu *et al.*, 2015] Yuanyuan Liu, Fanhua Shang, Licheng Jiao, James Cheng, and Hong Cheng. Trace norm regularized CANDECOMP/PARAFAC decomposition with missing data. *IEEE Transactions on Cybernetics*, 45(11):2437–2448, 2015.

[Lu *et al.*, 2016] Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan. Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5249–5257, 2016.

[Mardani *et al.*, 2015] Morteza Mardani, Gonzalo Mateos, and Georgios B. Giannakis. Subspace learning and imputation for streaming big data matrices and tensors. *IEEE Transactions on Signal Processing*, 63(10):2663–2677, 2015.

[Mørup and Hansen, 2009] Morten Mørup and Lars Kai Hansen. Automatic relevance determination for multiway models. *Journal of Chemometrics, Special Issue: In Honor of Professor Richard A. Harshman*, 23(7-8):352 – 363, 2009.

[Nimishakavi *et al.*, 2018] Madhav Nimishakavi, Bamdev Mishra, Manish Gupta, and Partha Talukdar. Inductive framework for multi-aspect streaming tensor completion with side information. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 307–316. ACM, 2018.

[Rendle *et al.*, 2009] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 727–736, 2009.

[Shao *et al.*, 2015] Weixiang Shao, Lifang He, and Philip S. Yu. Clustering on multi-source incomplete data via tensor modeling and factorization. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 485–497, 2015.

[Sharif and Bresler, 2007] Behzad Sharif and Yoram Bresler. Physiologically improved ncat phanotm enables in-silico study the effects of beat-to-beat variability on cardiac mri. In *Proceedings of International Society for Magnetic Resonance in Medicine (ISMRM)*, 2007.

[Song *et al.*, 2017] Qingquan Song, Xiao Huang, Hancheng Ge, James Caverlee, and Xia Hu. Multi-aspect streaming tensor completion. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 435–443. ACM, 2017.

[Zhang and Hawkins, 2018] Zheng Zhang and Cole Hawkins. Variational bayesian inference for robust streaming tensor factorization and completion. In *IEEE International Conference on Data mining (ICDM)*, pages 1446–1451, 2018.

[Zhou *et al.*, 2016] Shuo Zhou, Nguyen Xuan Vinh, James Bailey, Yunzhe Jia, and Ian Davidson. Accelerating online cp decompositions for higher order tensors. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1375–1384. ACM, 2016.