Research Article

Mixture Augmented Lagrange Multiplier Method for Tensor Recovery and Its Applications

Huachun Tan,¹ Bin Cheng,¹ Jianshuai Feng,² Li Liu,³ and Wuhong Wang¹

¹ Department of Transportation Engineering, Beijing Institute of Technology, Beijing 100081, China

² Integrated Information System Research Center, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

³ Marvell Semiconductor Inc., 5488 Marvell LN, Santa Clara, CA 95054, USA

Correspondence should be addressed to Huachun Tan; tanhc00@gmail.com

Received 30 November 2013; Accepted 30 January 2014; Published 17 March 2014

Academic Editor: Huimin Niu

Copyright © 2014 Huachun Tan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The problem of data recovery in multiway arrays (i.e., tensors) arises in many fields such as computer vision, image processing, and traffic data analysis. In this paper, we propose a scalable and fast algorithm for recovering a low- n -rank tensor with an unknown fraction of its entries being arbitrarily corrupted. In the new algorithm, the tensor recovery problem is formulated as a mixture convex multilinear Robust Principal Component Analysis (RPCA) optimization problem by minimizing a sum of the nuclear norm and the ℓ_1 -norm. The problem is well structured in both the objective function and constraints. We apply augmented Lagrange multiplier method which can make use of the good structure for efficiently solving this problem. In the experiments, the algorithm is compared with the state-of-art algorithm both on synthetic data and real data including traffic data, image data, and video data.

1. Introduction

A tensor is a multidimensional array. It is the higher-order generalization of vector and matrix, which has many applications in information sciences, computer vision, graph analysis [1], and traffic data analysis [2–4]. In the real world, as the size and the amount of redundancy of the data increase fast and nearly all of the existing high-dimensional real world data either have the natural form of tensor (e.g., multichannel images) or can be grouped into the form of tensor (e.g., tensor face [5], traffic data tensor model [2–4], and videos), challenges come up in many scientific areas when someone confronts with the high-dimensional real world data. Because of some reasons, one wants to capture the underlying low-dimensional structure of the tensor data or seeks to detect the irregular sparse patterns of the tensor data, such as image compression [6], foreground segmentation [7], saliency detection [8], and traffic data completion [2, 3]. As a consequence, it is desirable to develop algorithms that can capture the low-dimensional structure or the irregular sparse patterns in the high-dimensional tensor data.

In the two-dimensional case, that is, the matrix case, the “rank” and “sparsity” are the most useful tools for matrix-valued data analysis. Chandrasekaran et al. [9] proposed the concept of “rank-sparse incoherence” to depict the fundamental identifiability of recovering the low-rank and sparse components. Wright et al. [10] and Candes et al. [11] demonstrated that if the irregular sparse matrix S is sufficiently sparse (relative to the rank of L), one can accomplish the sparse and low-rank recovery by solving the following convex optimization problem:

$$\begin{aligned} \min_{L, S} : \quad & \|L\|_* + \mu \|S\|_1, \\ \text{s.t.} \quad & A = L + S, \end{aligned} \quad (1)$$

where $A \in \mathbb{R}^{m \times n}$ is the given matrix to be recovered; $L \in \mathbb{R}^{m \times n}$ is the low-rank component of A ; $S \in \mathbb{R}^{m \times n}$ is the sparse component of A ; $\|\cdot\|_*$ denotes the nuclear norm defined by the sum of all singular values; $\|\cdot\|_1$ denotes the sum of the absolute values of matrix entries; μ is a positive weighted parameter. This optimization method is called

the Robust Principal Component Analysis [10, 11] (RPCA) or the Principal Component Pursuit (PCP) due to its ability of exactly recovering the underlying low-rank matrix even in the presence of being corrupted by large entries or outliers.

Although the low-rank matrix recovery problem has been well studied, there is not much work on tensors. Li et al. [12] derived a method for the optimal Rank $-(R_1, R_2, R_3)$ tensor decomposition model. Considering a real n -mode tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$, the best Rank $-(R_1, R_2, R_3)$ approximation is to find a tensor $\mathcal{A}^* \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$ with prespecified $\text{rank}_k(A_{(k)}) = R_k$ that minimizes the least-squares cost function:

$$\begin{aligned} \min_{\mathcal{A}^*} \quad & f(\mathcal{A}^*) = \|\mathcal{A} - \mathcal{A}^*\|_F^2, \\ \text{s.t.} \quad & \text{Rank}_k(A_k) = R_k, \quad \forall k. \end{aligned} \quad (2)$$

The n -rank conditions imply that $\mathcal{A}^*$ should have the Tucker decomposition [13] as $\mathcal{A}^* \approx \times_1 U_1 \times_2 U_2 \dots \times_k U_k \dots \times_n U_n$. For the application, they applied the model to the high-dimensional tensor-like visual data by dividing the observed tensor into a low-dimensional structure plus unbounded but sparse irregular patterns: $\mathcal{A} = \mathcal{L} + \mathcal{S}$. By assumption that the n -rank of $\mathcal{L}$ should be small and the corruption $\mathcal{S}$ is bounded, the original function is as follows:

$$\begin{aligned} \min_{\mathcal{L}} \quad & \sum_i \lambda_i \text{rank}_i(\mathcal{L}), \\ \text{s.t.} \quad & \|\mathcal{L} - \mathcal{A}\|_F^2 < \varepsilon^2. \end{aligned} \quad (3)$$

In order to solve the problem, they made some conversions to (3) and extended the matrix robust PCA problem to the tensor case. A relaxation technique was used to separate the dependant relationships and the block coordinate descent (BCD) method was used to solve the low- n -rank tensor recovery problem. Then they proposed the rank sparsity tensor decomposition (RSTD) algorithm. In fact, their algorithm can be seen as a basic version of Lagrange multiplier method. Although being simple and provably correct, the RSTD algorithm requires a very large number of iterations to converge and it is difficult to choose the parameters for speedup. Besides, due to the property of the basic Lagrange multiplier method, the accuracy of results needs to be improved.

In this paper, a new algorithm for low- n -tensor recovery, which is termed as Mixture Augmented Lagrange Multiplier Method for Tensor Recovery (MALM-TR), is proposed. In the new algorithm, analogy to the RSTD [12], we convert the tensor recovery problem into a mixture convex optimization problem by adopting the relaxation technique strategy which eliminates the interdependent trace norm and ℓ_1 norm constrain. Actually, the elements involved in problem are all in matrix case. Thus, it can be treated as a multilinear extension of the RPCA problem and subsumes the matrix RPCA problem as a special case. Lin et al. [14] have proved that the matrix RPCA problem can be solved by ALM with achieving higher precision, being less storage/memory demanding, and having a pleasing Q-linear convergence

speed. Inspired by these merits of ALM, we try to extend the augmented Lagrange multiplier method (ALM) to the multilinear RPCA problem and prove that ALM is not only fit to solve the matrix RPCA problem but also suitable to solve the multilinear RPCA problem.

For the usage of this algorithm, it is applied to real world data recovery including traffic data recovery, image restoration, and background modeling.

In traffic data analysis area, due to detector and communication malfunctions, traffic data often confronts with the noising data phenomenon, especially the outlier value noise, which has a great impact on the performance of Intelligent Transportation System (ITS). Therefore, it is essential to solve the issues caused by outlier data in order to fully explore the applicability of the data and realize the ITS applications. In the application part of this paper, we introduce the tensor form to model the traffic data, which can encode the multimode (e.g., week, day, record) correlations of the traffic data simultaneously and preserve the multiway nature of traffic data. For example, it is assumed that a loop detector collects traffic volume data every 15 minutes. Thus, it will have 96 records in a day. If we have 20 weeks traffic volume data, these data can be formed into a tensor of size $20 \times 7 \times 96$. Then, the proposed tensor-based method which can well mine the multimode correlations of traffic data mentioned above is used to remove outlier noise of the traffic data.

It is observed that the multichannel image can be seen as a tensor with multidimensions. For example, RGB image has three channels including Red channel, Green channel, and Black channel. Thus, it can be represented as width $\times$ height $\times$ 3 which is a 3-dimensional tensor. For the application, the proposed method is used to remove the noise of the image. Though the method would not be reasonable for some natural images, it has many applications for visual data such as structured images (e.g., the façade image), CT/MRI data, and multispectral image. Besides images, video data can be grouped into the form of tensor. For example, there is a video which has 300 gray frames and each of which is in size of 200×200 . These video data can form a tensor of size $200 \times 200 \times 300$. For the video application, the proposed method will be used for background modeling.

The rest of the paper is organized as follows. Section 2 presents some notations and states some basic properties of tensors. Section 3 discusses the detailed process of our proposed algorithm. Section 4 tests the algorithm on different settings, varying from simulated data to applications in computer vision, image processing, and traffic recovery. Finally, some concluding remarks are provided in Section 5.

2. Notation and Basics on Tensor Model

In this paper, the nomenclatures and the notations in [1, 12] on tensors are partially adopted. Scalars are denoted by lowercase letters ($a, b, c, \dots$), vectors by bold lowercase letters ($\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots$), and matrices by uppercase letters ($A, B, C, \dots$). Tensors are written as calligraphic letters ($\mathcal{A}, \mathcal{B}, \mathcal{C}, \dots$). N -mode tensors are denoted as $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$. The elements of an N -mode are denoted as $a_{i_1 \dots i_k \dots i_n}$, where $1 \leq i_k \leq I_k$,

$1 \leq K \leq N$. The mode- n unfolding (also called matricization or flattening) of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is defined as $\text{unfold}(\mathcal{A}, n) = A_{(n)}$. The tensor element $(i_1, i_2, \dots, i_N)$ is mapped to the matrix element (i_n, j) , where

$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^N (i_k - 1) J_k, \quad \text{with } J_k = \prod_{\substack{m=1 \\ m \neq k}}^{k-1} I_m. \quad (4)$$

Therefore, $A_{(n)} \in \mathbb{R}^{I_n \times J}$, where $J = \prod_{\substack{k=1 \\ k \neq n}}^N I_k$. Accordingly, its inverse operator fold can be defined as $\text{fold}(A_{(n)}, n) = \mathcal{A}$.

The n -rank of a N -dimensional tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, denoted by r_n , is the rank of the mode- n unfolding matrix $A_{(n)}$:

$$r_n = \text{rank}_n(\mathcal{A}) = \text{rank}(A_{(n)}). \quad (5)$$

If the n -rank is very small related to the size of the tensor, we call it low- n -rank tensor.

The inner product of two same-size tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is defined as the sum of the products of their entries, that is,

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1} \sum_{i_2} \dots \sum_{i_N} a_{i_1 \dots i_k \dots i_n} b_{i_1 \dots i_k \dots i_n}. \quad (6)$$

The corresponding Frobenius norm is $\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. Besides, the ℓ_0 norm of a tensor $\mathcal{A}$, denoted by $\|\mathcal{A}\|_0$, is the number of nonzero elements in $\mathcal{A}$ and the ℓ_1 norm is defined as $\|\mathcal{A}\|_1 := \sum_{i_1 \dots i_k \dots i_n} |a_{i_1 \dots i_k \dots i_n}|$. It is clear that $\|\mathcal{A}\|_F = \|A_{(n)}\|_F$, $\|\mathcal{A}\|_0 = \|A_{(n)}\|_0$ and $\|\mathcal{A}\|_1 = \|A_{(n)}\|_1$ for any $1 \leq n \leq N$.

The n -mode (matrix) product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ with a matrix $M \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{A} \times_n M$ and is of size $I_1 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N$. In terms of flattened matrix, the n -mode product can be expressed as

$$\mathcal{Y} = \mathcal{A} \times_n M \iff Y_{(n)} = M A_{(n)}. \quad (7)$$

3. MALM-TR

This section is separated into 2 parts. In Section 3.1, we convert the low- n -tensor recovery problem into a multilinear RPCA problem. Section 3.2 simply introduces the ALM approach, extends ALM approach to solve the multilinear RPCA problem, and presents the details of the proposed algorithm.

3.1. The Multilinear RPCA Problem. The derivation starts with the general version [10] of matrix recovery problem:

$$\begin{aligned} \min_{L, S} \quad & \text{rank}(L) + \lambda \|S\|_0, \\ \text{s.t.} \quad & L + S = A, \end{aligned} \quad (8)$$

where $A \in \mathbb{R}^{m \times n}$ is the given matrix to be recovered; $L \in \mathbb{R}^{m \times n}$ is the low-rank component of A ; $S \in \mathbb{R}^{m \times n}$ is the sparse component of A ; $\text{rank}(L)$ denotes the rank of L ; $\|\cdot\|_0$ denotes the number of nonzero matrix entries; λ is a positive weighted

parameter. The higher-order tensor recovery problem can be generated from the matrix (i.e., 2nd-order tensor) case by utilizing the form of (8), leading to the formulation of the following:

$$\begin{aligned} \min_{\mathcal{L}, \mathcal{S}} \quad & \text{rank}_{\text{CP}}(\mathcal{L}) + \lambda \|\mathcal{S}\|_0, \\ \text{s.t.} \quad & \mathcal{L} + \mathcal{S} = \mathcal{A}, \end{aligned} \quad (9)$$

where $\mathcal{L}, \mathcal{S}, \mathcal{A}$ are n -mode tensors with identical size in each mode. $\mathcal{A}$ is the observed tensor data. $\mathcal{L}$ and $\mathcal{S}$ represent the correspondent structured part and irregular sparse part, respectively. $\text{rank}_{\text{CP}}(\mathcal{L})$ is the minimum number of rank-1 tensors that generates $\mathcal{L}$ as their sum [15, 16]. However, (9) is unsolvable because there is no straightforward algorithm to determine the CP-rank of a specific given tensor and the ℓ_0 norm is highly nonconvex. But when the given tensor is a low- n -rank tensor we can use the n -rank of unfolding of a tensor $\mathcal{A}$ instead of CP-rank of tensor to capture the global information of the given tensor. Therefore, we can minimize the n -ranks of the given tensor, respectively, instead of minimizing the CP-rank to solve the tensor completion problem. Obviously, $\|\mathcal{S}\|_0$ is equal to $\|S_{(n)}\|_0$. As a result, a function F which minimizes all the n -ranks of the given tensor to replace (9) is obtained as follows:

$$\begin{aligned} \min_{\mathcal{L}, \mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}} \quad & \left\{ F(\text{rank}(L_{(1)}), \text{rank}(L_{(2)}), \right. \\ & \left. \dots, \text{rank}(L_{(N)})) + \frac{1}{n} \sum_{i=1}^n \eta_i \|S_{(i)}\|_0 \right\}, \quad (10) \\ \text{s.t.} \quad & L_{(i)} + S_{(i)} = A_{(i)}, \end{aligned}$$

where $L_{(i)}, S_{(i)}$ are the mode- i unfoldings of $\mathcal{L}$ and $\mathcal{S}$. Equation (10) is a highly nonconvex optimization problem, and no efficient solution is known due to the nonconvexness of the matrix rank and ℓ_0 norm. Fortunately, it is a fact that the nuclear norm and ℓ_1 norm are the tightest convex approximation of rank and ℓ_0 norm [10, 11], respectively. By replacing rank with nuclear norm and replacing ℓ_0 -norm with ℓ_1 -norm, a tractable convex optimization problem can be obtained:

$$\begin{aligned} \min_{\mathcal{L}, \mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}} \quad & \left\{ F(\|L_{(1)}\|_*, \|L_{(2)}\|_*, \dots, \|L_{(n)}\|_*) \right. \\ & \left. + \frac{1}{n} \sum_{i=1}^n \eta_i \|S_{(i)}\|_1 \right\}, \quad (11) \\ \text{s.t.} \quad & L_{(i)} + S_{(i)} = A_{(i)}. \end{aligned}$$

In order to utilize the information of each mode as much as possible, the n -rank minimization problems of each mode are combined by weighted parameters to replace the function F

with $(1/n) \sum_i \lambda_i (\|L_{(i)}\|_*)$ which is defined in [17, 18]. Thus, the tensor completion problem becomes

$$\begin{aligned} \min_{\mathcal{L}, \mathcal{S}} \quad & \frac{1}{n} \sum_{i=1}^n \lambda_i \|L_{(i)}\|_* + \frac{1}{n} \sum_{i=1}^n \eta_i \|S_{(i)}\|_1, \\ \text{s.t.} \quad & L_{(i)} + S_{(i)} = A_{(i)}. \end{aligned} \quad (12)$$

Problem (12) is still hard to solve due to the inter-dependent trace norm and ℓ_1 norm constraint. In order to simplify the problem, we introduce additional auxiliary matrix $M_{(i)} = L_{(i)}$, $N_{(i)} = S_{(i)}$. Then, we relax the equality constraints by $\|L_{(i)} - M_{(i)}\|_F^2 \leq \varepsilon_2$ and $\|S_{(i)} - N_{(i)}\|_F^2 \leq \varepsilon_3$. It is easy to check that $\|A_{(i)} - L_{(i)} - S_{(i)}\|_F^2 \leq \varepsilon_1$ corresponds to the stable Principle Component Pursuit (sPCP) in the matrix case [19]. Finally, we get the relaxed form of (12) which can be seen as a multilinear RPCA problem:

$$\begin{aligned} \min_{\mathcal{L}, \mathcal{S}} \quad & \frac{1}{n} \sum_{i=1}^n \lambda_i \|M_{(i)}\|_* + \frac{1}{n} \sum_{i=1}^n \eta_i \|N_{(i)}\|_1, \\ \text{s.t.} \quad & \|A_{(i)} - L_{(i)} - S_{(i)}\|_F^2 \leq \varepsilon_1, \\ & \|L_{(i)} - M_{(i)}\|_F^2 \leq \varepsilon_2, \\ & \|S_{(i)} - N_{(i)}\|_F^2 \leq \varepsilon_3. \end{aligned} \quad (13)$$

3.2. Optimization Process. In [20], the general method of augmented Lagrange multipliers is introduced for solving constrained optimization problems of the kind:

$$\begin{aligned} \min \quad & f(X), \\ \text{s.t.} \quad & h(X) = 0, \end{aligned} \quad (14)$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}$ and $h: \mathbb{R}^n \rightarrow \mathbb{R}^m$, the augmented Lagrange function is defined as

$$(X, Y, \mu) = f(X) + \langle Y, h(X) \rangle + \frac{\omega}{2} \|h(X)\|_F^2, \quad (15)$$

where ω is a positive scalar, and then the optimization problem can be solved via the method of augmented Lagrange multipliers (see [21] for more details).

It is observed that (13) is well structured and the separable structure emerges in both the objective function and constraint conditions. We convert (9) into its augmented

Lagrange form with proper α_i , β_i , and γ_i . The augmented Lagrange of (13) is

$$\begin{aligned} \min_{\mathcal{L}, \mathcal{S}, M_{(i)}, N_{(i)}} \quad & \sum_{i=1}^n \lambda_i \|M_{(i)}\|_* + \sum_{i=1}^n \eta_i \|N_{(i)}\|_1 \\ & + \sum_{i=1}^n \langle Y_i, L_{(i)} - M_{(i)} \rangle + \sum_{i=1}^n \frac{\alpha_i}{2} \|L_{(i)} - M_{(i)}\|_F^2 \\ & + \sum_{i=1}^n \langle Z_i, S_{(i)} - N_{(i)} \rangle + \sum_{i=1}^n \frac{\beta_i}{2} \|S_{(i)} - N_{(i)}\|_F^2 \\ & + \sum_{i=1}^n \langle W_i, A_{(i)} - M_{(i)} - N_{(i)} \rangle \\ & + \sum_{i=1}^n \frac{\gamma_i}{2} \|A_{(i)} - M_{(i)} - N_{(i)}\|_F^2. \end{aligned} \quad (16)$$

Equation (16) can be simplified into its equivalent form:

$$\begin{aligned} \min_{\mathcal{L}, \mathcal{S}, M_{(i)}, N_{(i)}} \quad & \sum_{i=1}^n \lambda_i \|M_{(i)}\|_* + \sum_{i=1}^n \eta_i \|N_{(i)}\|_1 \\ & + \sum_{i=1}^n \frac{\alpha_i}{2} \left\| M_{(i)} - L_{(i)} - \frac{Y_i}{\alpha_i} \right\|_F^2 \\ & + \sum_{i=1}^n \frac{\beta_i}{2} \left\| N_{(i)} - S_{(i)} - \frac{Z_i}{\beta_i} \right\|_F^2 \\ & + \sum_{i=1}^n \frac{\gamma_i}{2} \left\| M_{(i)} + N_{(i)} - A_{(i)} - \frac{W_i}{\gamma_i} \right\|_F^2. \end{aligned} \quad (17)$$

The core idea of solving the optimization problem in (17) is to optimize a group of variables while fixing the other groups. The variables in the optimization are $M_{(1)}, \dots, M_{(n)}, N_{(1)}, \dots, N_{(n)}, S_{(i)}, L_{(i)}$ which can be divided into $2n + 2$ groups. To achieve the optimal solution, the method estimates $M_{(i)}, N_{(i)}, S_{(i)}, L_{(i)}$ sequentially, followed by certain refinement in each iteration.

Computing $M_{(i)}$. The optimal $M_{(i)}$ with all other variables fixed is the solution to the following subproblem:

$$\begin{aligned} \min_{M_{(i)}} \quad & \lambda_i \|M_{(i)}\|_* + \frac{\alpha_i}{2} \left\| M_{(i)} - L_{(i)} - \frac{Y_i}{\alpha_i} \right\|_F^2 \\ & + \frac{\gamma_i}{2} \left\| M_{(i)} + N_{(i)} - A_{(i)} - \frac{W_i}{\gamma_i} \right\|_F^2. \end{aligned} \quad (18)$$

As shown in [22], the optimal solution of (18) is given by

$$M_{(i)} = U_i D_{\lambda_i / \alpha_i + \gamma_i}(\Lambda) V_i^T, \quad (19)$$

where $U_i \Lambda V_i^T$ is the singular value decomposition given by

$$U_i \Lambda V_i^T = \frac{(\alpha_i (L_{(i)} + Y_i / \alpha_i) + \gamma_i (A_{(i)} - N_{(i)} + W_i / \gamma_i))}{(\alpha_i + \gamma_i)}, \quad (20)$$

Input: n -mode tensor $\mathcal{A}$.

Parameters: $\alpha, \beta, \gamma, \lambda, \eta, \rho$.

(1) Initialization: $M_{(i)} = L_{(i)}, N_{(i)} = 0, k = 1, \rho > 0$.

(2) Repeat until convergence

(3) for $i = 1$ to n

(4) $M_{(i)}^{k+1} = U_i D_{\lambda_i/\alpha_i + \gamma_i}(\Lambda) V_i^T$,

Where $U_i \Lambda V_i^T = \left(\alpha_i (L_{(i)}^k + Y_i^k / \alpha_i) + \gamma_i (A_{(i)}^k - N_{(i)}^k + W_i^k / \gamma_i) \right) / (\alpha_i + \gamma_i)$.

(5) $N_{(i)}^{k+1} = \mathcal{D}_{\eta_i/\beta_i + \gamma_i} \left(\left(\beta_i (S_{(i)}^k + Z_i^k / \beta_i) + \gamma_i (A_{(i)}^k - M_{(i)}^k + W_i^k / \gamma_i) \right) / (\beta_i + \gamma_i) \right)$.

(6) $Y_i^{k+1} = Y_i^k + \alpha_i (L_{(i)}^{k+1} - M_{(i)}^{k+1})$,

(7) $Z_i^{k+1} = Z_i^k + \beta_i (S_{(i)}^{k+1} - N_{(i)}^{k+1})$,

(8) $W_i^{k+1} = W_i^k + \gamma_i (A_{(i)} - M_{(i)}^{k+1} - N_{(i)}^{k+1})$.

(9) end for

(10) $\mathcal{L}^{k+1} = \sum_{i=1}^n \alpha_i \text{fold} (M_{(i)}^{k+1} - Y_i^{k+1} / \alpha_i, i) / \sum_{i=1}^n \alpha_i$,

(11) $\mathcal{S}^{k+1} = \sum_{i=1}^n \beta_i \text{fold} (N_{(i)}^{k+1} - Z_i^{k+1} / \beta_i, i) / \sum_{i=1}^n \beta_i$.

(12) $\alpha = \rho \alpha, \beta = \rho \beta, \gamma = \rho \gamma$.

(13) $k = k + 1$.

(14) End

Output: n -mode tensor $\mathcal{L}, \mathcal{S}$.

ALGORITHM 1: MALM-TR: MALM for tensor recovery.

and $\mathcal{D}_\tau$ is the “shrinkage” operation. The “shrinkage” operator $\mathcal{D}_\tau(x)$ with $\tau > 0$ is defined as

$$\mathcal{D}_\tau(x) = \begin{cases} x - \tau, & \text{if } x > \tau \\ x + \tau, & \text{if } x < -\tau \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

The operator can be extended to the matrix or tensor case by performing the shrinkage operator towards each element.

Computing $N_{(i)}$. The optimal $N_{(i)}$ with all other variables fixed is the solution to the following subproblem:

$$\begin{aligned} \min_{N_{(i)}} : & \eta_i \|N_{(i)}\|_1 + \frac{\beta_i}{2} \left\| N_{(i)} - S_{(i)} - \frac{Z_i}{\beta_i} \right\|_F^2 \\ & + \frac{\gamma_i}{2} \left\| M_{(i)} + N_{(i)} - A_{(i)} - \frac{W_i}{\gamma_i} \right\|_F^2. \end{aligned} \quad (22)$$

By the well-known ℓ_1 norm minimization [23], the optimal solution of (22) is

$$\begin{aligned} N_{(i)} = & \mathcal{D}_{\eta_i/\beta_i + \gamma_i} \\ & \times \left(\frac{\beta_i (S_{(i)} + Z_i/\beta_i) + \gamma_i (A_{(i)} - M_{(i)} + W_i/\gamma_i)}{(\beta_i + \gamma_i)} \right). \end{aligned} \quad (23)$$

Computing $S_{(i)}$. The optimal $\mathcal{S}$ with all other variables fixed is the solution to the following subproblem:

$$\min_{\mathcal{S}} : \sum_{i=1}^n \frac{\beta_i}{2} \left\| N_{(i)} - S_{(i)} - \frac{Z_i}{\beta_i} \right\|_F^2. \quad (24)$$

It is easy to show that the solution to (24) is given by

$$\mathcal{S} = \sum_{i=1}^n \frac{\beta_i \text{fold} (N_{(i)} - Z_i/\beta_i, i)}{\sum_{i=1}^n \beta_i}. \quad (25)$$

Computing $L_{(i)}$. The optimal $\mathcal{L}$ with all other variables fixed is the solution to the following subproblem:

$$\min_{\mathcal{L}} : \sum_{i=1}^n \frac{\alpha_i}{2} \left\| M_{(i)} - L_{(i)} - \frac{Y_i}{\alpha_i} \right\|_F^2. \quad (26)$$

It is easy to show that the solution to (26) is given by

$$\mathcal{L} = \sum_{i=1}^n \frac{\alpha_i \text{fold} (M_{(i)} - Y_i/\alpha_i, i)}{\sum_{i=1}^n \alpha_i}. \quad (27)$$

The pseudo-code of the proposed MALM-TR algorithm is summarized in Algorithm 1.

Under some rather general conditions, when $\{\alpha, \beta, \gamma\}$ is an increasing sequence and both the objective function and the constraints are continuously differentiable functions, it has been proven in [20] that the Lagrange multipliers $\{Y_i, Z_i, W_i\}$ produced by Algorithm 1 converge Q-linearly to the optimal solution when $\{\alpha, \beta, \gamma\}$ is bounded and super-Q-linearly when $\{\alpha, \beta, \gamma\}$ is unbounded. Another merit of MALM-TR is that the optimal step size to update $\{Y_i, Z_i, W_i\}$ is proven to be the chosen penalty parameters $\{\alpha, \beta, \gamma\}$, making the parameter tuning much easier. A third merit of MALM-TR is that the algorithm converges to the exact optimal solution, even without requiring $\{\alpha, \beta, \gamma\}$ to approach infinity [20].

4. Experiments

In this section, using both the numerical simulations and the real world data, we evaluate the performance of our proposed

TABLE 1: $\mathcal{A}_0 \in \mathbb{R}^{40 \times 40 \times 40}$, n -rank = [5, 5, 5], spr = 5%, 15%, 25%, 35%.

spr	Algorithm: MALM-TR				Algorithm: RSTD			
	RSE- $\mathcal{L}_0$ ($e - 5$)	RSE- $\mathcal{S}_0$ ($e - 5$)	# iter	Time (s)	RSE- $\mathcal{L}_0$ ($e - 5$)	RSE- $\mathcal{S}_0$ ($e - 5$)	# iter	Time (s)
0.05	0.01	0.006	136	17.7	450	430	226	19.5
0.15	0.06	0.02	167	19.5	910	440	330	27.4
0.25	1.1	0.3	281	29.1	1510	490	714	44.4
0.35	2010	390	450	41.9	5620	1140	608	44.1

TABLE 2: $\mathcal{A}_0 \in \mathbb{R}^{40 \times 40 \times 40}$, n -rank = [10, 10, 10], spr = 5%, 10%, 15%, 20%.

spr	Algorithm: MALM-TR				Algorithm: RSTD			
	RSE- $\mathcal{L}_0$ ($e - 5$)	RSE- $\mathcal{S}_0$ ($e - 5$)	# iter	Time (s)	RSE- $\mathcal{L}_0$ ($e - 5$)	RSE- $\mathcal{S}_0$ ($e - 5$)	# iter	Time (s)
0.05	0.2	0.1	243	29.4	400	230	411	33.8
0.10	10.1	4.2	323	43.5	520	270	568	51.3
0.15	120	42	596	62.5	650	260	1103	68.5
0.20	1250	370	972	88.3	2650	780	1235	88.1

algorithm and then compare the results with RSTD on the low- n -rank tensor recovery problem.

In all the experiments, the Lanczos bidiagonalization algorithm with partial reorthogonalization [24] is adopted to obtain a few singular values and vectors in all iterations. A major challenge of our algorithm is the selection of parameters. We simply set the parameters $\alpha = \beta = \gamma = [I_1/I_{\max}, I_2/I_{\max}, \dots, I_n/I_{\max}]^T$ for all experiments, where $I_{\max} = \max\{I_i\}$. Similarly, we choose $\eta = 1/\sqrt{I_{\max}}$ as suggested in [11] and tune λ with the change of η . For comparing with RSTD [12], we also use the difference of $\mathcal{L}$ and $\mathcal{S}$ in successive iterations against a certain tolerance as the stopping criterion. All the experiments are conducted and timed on the same desktop with a Pentium (R) Dual-Core 2.50 GHz CPU that has 4 GB memory, running on Windows 7 and Matlab.

4.1. Numerical Simulations. A low- n -rank tensor $\mathcal{L}_0$ is generated as follows. The N -way Tensor Toolbox [25] is used to generate a third-order tensor with the size of $I_1 \times I_2 \times I_3$ and the relative small n -rank $[r_1 \ r_2 \ r_3]$. The generated tensor is in Tucker model [13] described as $\mathcal{L}_0 = \mathcal{C}_{\times 1} X_{\times 2} Y_{\times 3} Z$. To impose these rank conditions, $\mathcal{C}$ is $\mathbb{R}^{r_1 \times r_2 \times r_3}$ core tensor with each entry being sampled independently from a standard Gaussian distribution $\mathcal{N}(0, 1)$, X, Y and Z are $I_1 \times r_1, I_2 \times r_2, I_3 \times r_3$ factor matrices generated by randomly choosing each entry from $\mathcal{N}(0, 1)$. Here without loss of generality we make the factor matrices orthogonal. But one major difference is that the n -ranks are always different along each mode while the column rank and row rank of a matrix are equal to each other. For simplicity, in this paper we set the mode- n ranks with the same value.

The entries of sparse tensor $\mathcal{S}_0$ are independently distributed, each taking on value 0 with probability $1 - \text{spr}$, and each taking on impulsive value with probability spr . The recovered tensor $\mathcal{A}_0$ is generated as $\mathcal{A}_0 = \mathcal{L}_0 + \mathcal{S}_0$.

The simulated tensor used in the experiments is of size $40 \times 40 \times 40$, varying the n -rank r and the sparse ratio spr . The parameters are adjusted according to the different r and spr . The quality of recovery is measured by the relative square error (RSE) to $\mathcal{L}_0$ and $\mathcal{S}_0$, which is defined as

$$\text{RSE-}\mathcal{L}_0 = \frac{\|\widehat{\mathcal{L}} - \mathcal{L}_0\|_F}{\|\mathcal{L}_0\|_F}, \quad \text{RSE-}\mathcal{S}_0 = \frac{\|\widehat{\mathcal{S}} - \mathcal{S}_0\|_F}{\|\mathcal{S}_0\|_F}. \quad (28)$$

Tables 1 and 2 present the average results (across 10 instances) for different sparse ratio. The results demonstrate that our proposed algorithm MALM-TR outperforms RSTD on either efficiency or accuracy.

4.2. Image Restoration. One straightforward application of our algorithm is the image restoration. Same as [12] pointed, our algorithm also assumes the image to be well structured. Though the assumption would not be reasonable for some natural images, it has many applications for visual data such as structured images (e.g., the façade image), CT/MRI data, and multispectral image. In experiments, we apply the algorithm on image restoration of the façade image, which is also used in [12, 17]. We add different percent of random impulsive noise to the image and compare MALM-TR with RSTD. The results produced by both algorithms are shown in Figure 1.

4.3. Background Modeling. Another application of our algorithm is to estimate a good model for the background variations in a scene (i.e., background modeling). In this situation, it is natural to model the background variation as approximately low rank. Foreground objects generally occupy only a fraction of the image pixels and hence can be treated as sparse part.

We test our algorithm using an example from [26] and compare with RSTD [12]. The visual comparisons of the background modeling are shown in Figure 2. It is observed that

FIGURE 1: Comparisons in terms of visual effects. The rows (1), (2), and (3) correspond to the images before recovery, the obtained results by MALM-TR and RSTD [12], respectively. The columns (a), (b), and (c) correspond to the images corrupted by 15%, 25%, and 35% sparse impulsive noise, respectively. (d) is the original image.

FIGURE 2: Background modeling. Top: original video sequence of a scene. Middle: foreground object recovered by MALM-TR. Bottom: foreground object recovered by RSTD [12]. The results are highlighted for both RSTD and MALM-TR.

our algorithm is effective in separating the background which is even a dynamic scene. The results are also comparable to RSTD.

4.4. Traffic Data Recovery. In our previous work [3, 4], we have proposed two tensor-based methods on traffic data application. In [3], a tensor imputation method based on Tucker decomposition is developed to estimate the missing

value. As a fact that the exact coordinate and the number of the missing data in the tensor form can be observed and obtained because if an element in the tensor form is missing, it doesn't have value so we can recognize it easily. While this paper recovers a low- n -rank tensor that is arbitrarily corrupted by a fraction of noise based on the trace norm and ℓ_1 -norm optimization. The number and the coordinate of the corrupted data are unknown or not easy to obtain. That

FIGURE 3: Tensor model of size $8 \times 7 \times 288$.

TABLE 3: Comparison of RSE on traffic data.

SPR	MALM-TR(RSE)	RSTD(RSE)	Unrecovered(RSE)
0.05	0.0406	0.0525	0.2614
0.10	0.0432	0.0543	0.4117
0.15	0.0593	0.1370	0.5427
0.20	0.0784	0.2198	0.6859
0.25	0.1478	0.3959	0.8218

means that it is hard to recognize the corrupted data, because the corrupted data have values and hardly be separated from the correct data. The problems solved by the two papers are two different problems. Paper [4] is written from the point of traffic data recovery application which is the same problem that will be solved in this section. The main difference of the two proposed methods is how to use the constraint condition $\mathcal{A} = \mathcal{L} + \mathcal{S}$. Reference [4] puts the constraint condition to the minimized function with only one parameter γ , which leads the objective function to contain not only tensor but also matrix. However, as the size and structure of each mode of the given tensor data are not always the same, the contribution of each mode of the tensor to the final result may be different. In order to utilize the information of the constraint condition as much as possible, this paper unfolds constraint condition along each mode and use weighted parameters γ_i to obtain the new constraint condition in matrix versions $\sum_{i=1}^n (\gamma_i/2) \|A_{(i)} - M_{(i)} - N_{(i)}\|_F^2$ which is put into the minimized function using the augmented Lagrange multiplier strategy. With different objective functions, the optimized process is different too. More details can be found in [4].

In the fourth part of the experiment section, we will apply the proposed algorithm to traffic data recovery. The data used in the experiment are collected by a fixed loop in Sacramento County and downloaded from <http://pems.dot.ca.gov/>. The period of the data lasts for 77 days from March 14 to May 29, 2011. The traffic volume data are recorded every 5 minutes. Therefore, a daily traffic volume series for a loop detector contains 288 records. To finish traffic data recovery by the proposed algorithm, the first step is to convert the mass traffic data into a tensor form. In this part, we choose 8-week

FIGURE 4: Comparison of RSE curves on traffic data.

complete traffic volume data from the 77 days. Then, the 8-week data are formed into a tensor model of size $8 \times 7 \times 288$ as Figure 3 shows. In this model, “8” stands for 8 weeks, “7” stands for seven days in one week, and “288” stands for 288 records in one day.

In our previous work [3], the similarity coefficient [27] had been used to analyze the high multimode correlation (“link” mode, “week” mode, “day mode,” and “hour” mode) of traffic data from the point of view of statistic characteristic. For the high multicorrelations of traffic data, the tensor form of size $8 \times 7 \times 288$ can be approximated by a low- n -rank tensor.

According to the above description, the traffic data are reasonably converted into a tensor form which can be approximated by a low- n -rank tensor. In the traffic data recovery experiment, it is assumed that a subset of entries of the traffic data tensor form is corrupted by impulsive noise at random. The ratios of noisy are set from 5% to 25% with the tolerance 5%. Then we compare the proposed method with RSTD algorithm using RSE as the criterion. The criterion is defined as the following function shows:

$$\text{RSE} = \frac{\|\text{recovered data} - \text{original data}\|_F}{\|\text{original data}\|_F}. \quad (29)$$

Table 3 tabulates the RSEs by sparse impulsive noise with different ratio on traffic data. Especially, the unrecovered column presents the RSE between corrupted data and the original data. From data in the table, it is observed that the RSEs obtained by MALM-TR and RSTD are much smaller than the unrecovered data, which means that both algorithm can improve the quality of the corrupted data. Moreover, the RSEs of MALM-TR are smaller than RSTD. From the curves of Figure 4, it is vividly shown that our method performs better than RSTD.

5. Conclusion

In this paper, we extend the matrix recovery problem to low- n -rank tensor recovery and propose an efficient algorithm

based on mixture augmented Lagrange multiplier method. The proposed algorithm can automatically separate the low- n -rank tensor data and sparse part. Experiments show that the proposed algorithm is more stable and accurate in most cases and has excellent convergence rate. Different application examples show the broad applicability of our proposed algorithm in computer vision, image processing, and traffic data recovery.

In the future, we would like to investigate how to automatically choose the parameters in our algorithm and develop more efficient method for tensor recovery problem. Also we will explore more applications of our method.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The research was supported by NSFC (Grant nos. 61271376, 51308115, and 91120010), the National Basic Research Program of China (973 Program: no. 2012CB725405), and Beijing Natural Science Foundation (4122067). The authors would like to thank Professor Bin Ran from the University of Wisconsin-Madison and Yong Li from the University of Notre Dame for the suggestive discussions.

References

- [1] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [2] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011.
- [3] H. Tan, G. Feng, J. Feng, W. Wang, Y. J. Zhang, and F. Li, "A tensor-based method for missing traffic data completion," *Transportation Research Part C*, vol. 28, pp. 15–27, 2013.
- [4] H. Tan, G. Feng, J. Feng, W. Wang, and Y. J. Zhang, "Traffic volume data outlier recovery via tensor model," *Mathematical Problems in Engineering*, vol. 2013, Article ID 164810, 8 pages, 2013.
- [5] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *Proceedings of the European Conference on Computer Vision (ECCV '02)*, pp. 447–460, 2002.
- [6] A. S. Lewis and G. Knowles, "Image compression using the 2-D wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 244–250, 1992.
- [7] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1778–1792, 2005.
- [8] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [9] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," <http://arxiv.org/abs/0906.2220>.
- [10] J. Wright, Y. Peng, Y. Ma, A. Ganesh, and S. Rao, "Robust principal component analysis: exact recovery of corrupted low-rank matrices by convex optimization," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*, pp. 2080–2088, December 2009.
- [11] E. J. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" 2009, arXiv:0912.3599v1.
- [12] Y. Li, J. Yan, Y. Zhou, and J. Yang, "Optimum subspace learning and error correction for tensors," in *Proceedings of the 11th European Conference on Computer Vision (ECCV '10)*, Crete, Greece, 2010.
- [13] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
- [14] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of a corrupted low-rank matrices," *Mathematical Programming*, 2009.
- [15] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n -way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.
- [16] R. A. Harshman, "Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.
- [17] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," in *Proceedings of the International Conference on Computer Vision (ICCV '09)*, 2009.
- [18] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013.
- [19] Z. Zhou, X. Li, J. Wright, E. Candès, and Y. Ma, "Stable principal component pursuit," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '10)*, pp. 1518–1522, June 2010.
- [20] D. P. Bertsekas and A. E. Ozdaglar, "Pseudonormality and a lagrange multiplier theory for constrained optimization," *Journal of Optimization Theory and Applications*, vol. 114, no. 2, pp. 287–343, 2002.
- [21] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1999.
- [22] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.
- [23] E. T. Hale, W. Yin, and Y. Zhang, "Fixed-point continuation for ℓ_1 -minimization: methodology and convergence," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1107–1130, 2008.
- [24] H. D. Simon, "The lanczos algorithm with partial reorthogonalization," *Mathematics of Computation*, vol. 42, pp. 115–142, 1984.
- [25] C. A. Andersson and R. Bro, "The N-way toolbox for MATLAB, chemometrics and intelligent laboratory systems," *Chemometrics & Intelligent Laboratory Systems*, vol. 52, no. 1, pp. 1–25.4, 2000.
- [26] J. Zhong and S. Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust Kalman filter," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 44–50, October 2003.
- [27] Y. Zhang and Y. Liu, "Missing traffic flow data prediction using least squares support vector machines in urban arterial streets," in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009*, pp. 76–83, April 2009.