Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 20, No. 3, pp. 569–588, 2024

A NEW LOW-RANK TENSOR LINEAR REGRESSION WITH APPLICATION TO DATA ANALYSIS

Chenjian Pan, Hongjin He* and Chen Ling†

Dedicated to Professor Masao Fukushima on the occasion of his 75th birthday

Abstract: Linear regression is one of the most popular tools for data analysis. However, many existing works are limited to vector/matrix based models, which are often less efficiently to deal with high-dimensional data. In this paper, we aim to develop a transformed low-rank tensor regression model and apply it to data analysis. Unlike many tensor regression models, we represent the testing sample as a linear combination of all training samples under the transformed T-product (i.e., tensor-tensor product). Moreover, we accordingly propose an easily implementable ADMM-based algorithm for solving the underlying optimization model. A series of numerical experiments demonstrate that our approach works well on color face classification and traffic flow datasets.

Key words: *Linear regression, tensor, low-rank, T-product, ADMM*

Mathematics Subject Classification: *15A18, 15A69, 90C06, 68U10*

1 Introduction

Linear regression has received much consideration in a variety of fields, such as pattern recognition, economics, reinforcement learning and traffic data prediction, e.g., see [4, 9, 10, 13, 18, 22, 23, 28]. Naseem et al. [17] proposed a linear representation approach for face classification (LRA), which represents a testing sample as a linear combination of the class-specific samples and classifies the testing sample by minimizing the reconstruction error. Following the LRA, there are many variants such as nearest feature line, nearest feature plane, and nearest feature space method, e.g., see [3, 15]. To avoid over-fitting, some regularization terms such as the ℓ_1 and ℓ_2 norms are imposed into linear regression models. For example, Wright et al. [26] introduced a sparse representation-based regression (SRR) method, which is robust when noise is sparse. Zhang et al. [29] studied deeply the rules of SRR and argued that collaborative representation could improve the regression performance than those ℓ_1 norm based models. However, such methods are usually assumed that the errors of all elements are independent Gaussian or Laplacian distribution. Therefore, when

*H.J. He is supported in part by Zhejiang Provincial Natural Science Foundation of China (No. LZ24A010001), Ningbo Natural Science Foundation (Project ID: 2023J014), and National Natural Science Foundation of China (No. 12371303). C. Ling was supported in part by National Natural Science Foundation of China (No. 11971138).

†Corresponding author

DOI: https://doi.org/10.61208/pjo-2023-038

facing more complicated real-world cases, the aforementioned methods do not necessarily work stable. Fortunately, it is documented in [16, 24] that a strongly or approximately low-rank property is hidden in many real-world datasets. Therefore, a natural idea is to pursue a low-rank objective while satisfying some desired constraints. However, due to the well-known NP-hardness of the rank minimization, Fazel [5] accordingly introduced the nuclear norm concept of matrix to approximate the rank function. As an important application of the nuclear norm, Qian et al. [19] introduced a new model equipped with nulcear norm and ℓ_2 norm in objective function. Besides, Yang et al. [27] further studied low-rank based matrix regression (MR), in addition to introducing a quadratically convergent algorithm to solve the underlying MR model.

Compared with the aforementioned regression methods tailored for one- and two-dimensional datasets, developing customized models for high-dimensional datasets is still in its infancy. When dealing with high-dimensional cases, we usually unfold the data as vectors or matrices so that the aforementioned methods could be applied. However, this unfolding way often destroys the intrinsic multi-way structure of the data. Most recently, Gao et al. [7] introduced a tensor linear regression model for three-dimensional datasets based upon the well-known tensor-Singular Value Decomposition (t-SVD for short, see [12]). Given a set of training samples $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_N \in \mathbb{K}^{m\times n\times p}$ and a set of representation coefficients $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$, the tensor linear regression model [7] reads as

$$\begin{aligned} &\min_{\boldsymbol{x},\mathcal{E}} \|\mathcal{E}\|_{\mathrm{TNN}} + \lambda\|\boldsymbol{x}\|_2^2 \\ &s.t.\ \mathcal{X} = \mathcal{T}(\boldsymbol{x}) + \mathcal{E}, \end{aligned} \tag{1.1}$$

where $\mathcal{E}$ is the representation residual, $\|\cdot\|_{\mathrm{TNN}}$ is the so-called tensor nuclear norm (see [30]), $\lambda > 0$ is a tuning parameter, $\mathcal{X}$ is a test sample, and $\mathcal{T}(\boldsymbol{x})$ is defined by

$$\mathcal{T}(\boldsymbol{x}) = x_1\mathcal{A}_1 + \cdots + x_N\mathcal{A}_N.$$

However, model (1.1) represents the test sample as a linear combination of all training samples with a vector $\boldsymbol{x}$, which eventually treats each frontal slice with a same coefficient. In this situation, we could rewrite $\mathcal{T}(\boldsymbol{x})$ as

$$\mathrm{unfold}(\mathcal{T}(\boldsymbol{x})) = x_1\mathrm{unfold}(\mathcal{A}_1) + \cdots + x_N\mathrm{unfold}(\mathcal{A}_N),$$

where 'unfold$(\cdot)$' is a linear operator unfolding a tensor into a matrix (see [16]). Therefore, such an approach can be regarded as a matrix-based method.

In this paper, we make a further study on tensor linear regression for high-dimensional data analysis. Specifically, we introduce a low-rank tensor linear regression model, which is based on the widely used transformed T-product (i.e., tensor-tensor product, see [12]). As shown in [8, 20], the transformed T-product possibly yields better numerical performance than the pure T-product versions when an appropriate transformation is chosen. Moreover, transformed T-product not only inherits many promising properties of matrices, but also can better exploit the multi-way structure of tensors than those matrix-based approaches. Comparing with the methods directly unfolding tensors as matrices, our proposed method is able to efficiently consider the information of the third direction of tensors. Moreover, to avoid over-fitting, we impose a Tikhonov regularization term and a nuclear norm term on the coefficient matrix. To tackle the nonsmoothness of the objective function, we propose an implementable algorithm based on the classical ADMM (i.e., alternating direction method of multipliers) framework. It is noteworthy that our algorithm enjoys easy subproblems with closed-form solutions. To highlight the reliability of our approach, we conduct a series

of numerical experiments on color face classification and traffic datasets. Computational results support the idea of this paper.

This paper is divided into five parts. In Section 2, we summarize some notations and recall some basic definitions that will be used in this paper. In Section 3, we introduce our new tensor linear regression model. By introducing an auxiliary variable to separate the nonsmooth and smooth terms, we propose an implementable algorithm to solve the underlying model. Moreover, we establish the convergence result of the proposed algorithm. In Section 4, we conduct the numerical performance of our method in synthetic and real-world datasets. Finally, some concluding remarks are drawn in Section 5.

2 Preliminaries

Throughout this paper, the field of real numbers is denoted as $\mathbb{K}$. Without special instructions, we denote scalars and vectors by lowercase letters (e.g., $x, y, \ldots$) and boldfaced lowercase letters (e.g., $\boldsymbol{x}, \boldsymbol{y}, \ldots$), respectively. Matrices and tensors are denoted by capital letters (e.g., $X, Y, \ldots$) and calligraphic letters (e.g., $\mathcal{X}, \mathcal{Y}, \ldots$), respectively. For given integer n, we denote $[n] = \{1, 2, \ldots, n\}$.

For a third-order tensor $\mathcal{A} \in \mathbb{K}^{m\times n\times p}$, we denote its (i,j,k)th entry as $\mathcal{A}_{ijk}$, use the MATLAB notation $\mathcal{A}(:,:,k)$ to denote the kth frontal slice, use the notation $\mathbf{a}_{ij} \in \mathbb{K}^{1\times 1\times p}$ to denote the (i,j)th **tube fiber** of $\mathcal{A}$, that is, $\boldsymbol{a}_{ij} = \mathcal{A}(i,j,:)$ for $i \in [m]$ and $j \in [n]$, and denote the kth entry in the tube $\boldsymbol{a}_{ij}$ by $\boldsymbol{a}_{ij}(k)$ for $k \in [p]$. Accordingly, by using the tube notation and introducing the notation $\mathbb{K}_p^{m\times n}$ in palace of $\mathbb{K}^{m\times n\times p}$, the third-order tensor $\mathcal{A} \in \mathbb{K}^{m\times n\times p}$ can be viewed as an $m \times n$ matrix consisting of tubal scalars, which is denoted by $\mathcal{A} = (\boldsymbol{a}_{ij}) \in \mathbb{K}_p^{m\times n}$. Under this setting, we denote the kth frontal slice of $\mathcal{A} \in \mathbb{K}_p^{m\times n}$ by $A_{(k)} = (\boldsymbol{a}_{ij}(k))$ being exactly an $m \times n$ matrix. Moreover, for given $\mathcal{A} = (\boldsymbol{a}_{ij}), \mathcal{B} = (\boldsymbol{b}_{ij}) \in \mathbb{K}_p^{m\times n}$, their inner product is defined as $\langle \mathcal{A}, \mathcal{B}\rangle = \sum_{i=1}^m \sum_{j=1}^n \langle \boldsymbol{a}_{ij}, \boldsymbol{b}_{ij}\rangle$, and the associated Frobenius norm of $\mathcal{A}$ is defined as $\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A}\rangle}$. In particular, when A, B are matrices, their the inner product is given by $\langle A, B\rangle = \mathrm{trace}(A^\top B)$ with $A^\top$ denoting the transpose of A and $\mathrm{trace}(\cdot)$ representing the trace of a matrix. Throughout, $\|A\|_* = \sum_{i=1}^{\min\{m,n\}} \sigma_i(A)$ is called the nuclear norm of a matrix $A \in \mathbb{K}^{m\times n}$, and $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ denote the largest and smallest singular value of A, respectively. Denote by $\mathbb{S}^{m\times m}$ the $m \times m$ real symmetric matrices set, and $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ stand for the largest and smallest eigenvalues of given $A \in \mathbb{S}^{m\times m}$, respectively. For a given tube matrix $\mathcal{C} \in \mathbb{K}_p^{m\times n}$ (indeed a tensor $\mathcal{C} \in \mathbb{K}^{m\times n\times p}$), we denote the operator by $\mathrm{Vec}(\mathcal{C})$ converting $\mathcal{C}$ into a tube vector in $\mathbb{K}_p^{mn}$ by stacking columns of tube matrix $\mathcal{C} \in \mathbb{K}_p^{m\times n}$ on top of another.

Let $Q = (Q_{ij}) \in \mathbb{K}^{p\times p}$ be an orthogonal matrix. We define a mapping $\phi_Q : \mathbb{K}^p \to \mathbb{K}^p$ by $\bar{\boldsymbol{a}} := \phi_Q(\boldsymbol{a}) = Q\boldsymbol{a}$ for $\boldsymbol{a} \in \mathbb{K}^p$, and its inverse mapping $\phi_Q^{-1} : \mathbb{K}^p \to \mathbb{K}^p$ is defined by $\phi_Q^{-1}(\boldsymbol{a}) := Q^{-1}\boldsymbol{a}$ for $\boldsymbol{a} \in \mathbb{K}^p$. For given $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{K}^p$, their product with respect to Q is the tubal scalar given by $\boldsymbol{a} \odot_Q \boldsymbol{b} = \phi_Q^{-1}(\phi_Q(\boldsymbol{a}) \circ \phi_Q(\boldsymbol{b}))$, where $\circ$ is the Hadamard product of vectors.

Definition 2.1. Let $Q \in \mathbb{K}^{p\times p}$ be an orthogonal matrix. The transformed T-product of $\mathcal{A} = (\boldsymbol{a}_{il}) \in \mathbb{K}_p^{m\times s}$ and $\mathcal{B} = (\boldsymbol{b}_{lj}) \in \mathbb{K}_p^{s\times n}$, denoted by $\mathcal{A} \circledast_Q \mathcal{B}$, is a tensor $\mathcal{C} \equiv (\boldsymbol{c}_{ij}) \in \mathbb{K}_p^{m\times n}$, which is given by

$$\boldsymbol{c}_{ij} = \sum_{l=1}^{s} \boldsymbol{a}_{il} \odot_Q \boldsymbol{b}_{lj}, \quad i \in [m] \text{ and } j \in [n].$$

In particular, for $\boldsymbol{x} \in \mathbb{K}^p$ and $\mathcal{A} = (\boldsymbol{a}_{ij}) \in \mathbb{K}_p^{m\times n}$, we have $(\boldsymbol{x} \odot_Q \mathcal{A})_{ij} = \boldsymbol{x} \odot_Q \boldsymbol{a}_{ij}$, $i \in [m]$ and $j \in [n]$.

Such a transformed T-product $\mathcal{A}\circledast_Q\mathcal{B}$ is equivalently derived by considering a third-order tensor as a matrix of tube fibers, leaving matrix multiplication untouched, but replacing scalar multiplication with the definition required to handle tube fibers. If Q is specified as the discrete Fourier transform (DFT) matrix and both $\mathcal{A}$ and $\mathcal{B}$ are complex tensors, then $\mathcal{A}\circledast_Q\mathcal{B}$ immediately reduces to the classical T-product introduced by Kilmer et al. [12]. The cosine transform product defined in [11] is also an example of a $\circledast_Q$-family product.

Let $Q\in\mathbb{K}^{p\times p}$ be orthogonal, $\mathcal{A}=(\boldsymbol{a}_{ij})\in\mathbb{K}_p^{m\times s},\mathcal{B}=(\boldsymbol{b}_{ij})\in\mathbb{K}_p^{s\times n}$. Denote $\bar{\mathcal{A}}:=\Phi_Q(\mathcal{A})=(\phi_Q(\boldsymbol{a}_{ij}))$ and $\bar{\mathcal{B}}:=\Phi_Q(\mathcal{B})=(\phi_Q(\boldsymbol{b}_{ij}))$. Then, we have $\langle\mathcal{A},\mathcal{B}\rangle=\langle\bar{\mathcal{A}},\bar{\mathcal{B}}\rangle$ and

$$\mathcal{C}=\mathcal{A}\circledast_Q\mathcal{B}\quad\Leftrightarrow\quad\bar{\boldsymbol{c}}_{ij}(k)=\sum_{l=1}^{s}\bar{\boldsymbol{a}}_{il}(l)\bar{\boldsymbol{b}}_{lj}(l),\quad k\in[p],$$

which means

$$\bar{C}_{(k)}=\bar{A}_{(k)}\bar{B}_{(k)},\qquad k\in[p].\tag{2.1}$$

Definition 2.2. Let Q be an orthogonal matrix and $\mathcal{A}\in\mathbb{K}_p^{m\times n}$. Then $\mathcal{B}=(\boldsymbol{b}_{ij})\in\mathbb{K}_p^{n\times m}$ is called the transpose of $\mathcal{A}$, and denoted as $\mathcal{A}^\top=\mathcal{B}$, if $\boldsymbol{b}_{ij}=\boldsymbol{a}_{ji}$ for $i\in[n]$ and $j\in[m]$.

From Definition 2.2, we know that $\Phi_Q(\mathcal{A}^\top)(k)=(\Phi_Q(\mathcal{A})(k))^\top$ for $k\in[p]$.

Proposition 2.3. *Let Q be an orthogonal transformation. The multiplication reversal property of the conjugate transpose holds: $(\mathcal{A}\circledast_Q\mathcal{B})^\top=\mathcal{B}^\top\circledast_Q\mathcal{A}^\top$ for $\mathcal{A}\in\mathbb{K}_p^{m\times s}$ and $\mathcal{B}\in\mathbb{K}_p^{s\times n}$.*

Proof. It follows from Definition 2.1 and Definition 2.2. □

Letting $\mathcal{A}\in\mathbb{K}_p^{m\times m}$, $\mathcal{A}$ is **f-diagonal** if each frontal slice of $\mathcal{A}$ is diagonal, and we call $\mathcal{A}$ an identity tensor under Q, denoted by $\mathcal{I}_m$, if it is f-diagonal and all of its diagonal tube fibers are $\boldsymbol{e}_Q$, where $\boldsymbol{e}_Q=\phi_Q^{-1}(\boldsymbol{e})$ and $\boldsymbol{e}=(1,1,\ldots,1)^\top\in\mathbb{K}_p$. It is obvious that $\mathcal{A}$ is an identity tensor under Q, if and only if $\bar{A}_{(k)}$ is an $m\times m$ identity matrix for every $k\in[p]$. A tensor $\mathcal{A}\in\mathbb{K}_p^{m\times m}$ is called nonsingular (invertible) under the transformed T-product $\circledast_Q$, if $\mathcal{A}\circledast_Q\mathcal{B}=\mathcal{B}\circledast_Q\mathcal{A}=\mathcal{I}_m$ for some $\mathcal{B}\in\mathbb{K}_p^{m\times m}$. In that case, we denote $\mathcal{A}^{-1}=\mathcal{B}$. Particularly, if $\mathcal{A}^{-1}=\mathcal{A}^\top$, then we call $\mathcal{A}$ a $\circledast_Q$-orthogonal tensor. If $\mathcal{A}\in\mathbb{K}_p^{m\times q}$ $(q\le m)$ satisfies $\mathcal{A}^\top\circledast_Q\mathcal{A}=\mathcal{I}_q$, we say that $\mathcal{A}$ is partially $\circledast_Q$-orthogonal. Let $\mathcal{A}\in\mathbb{K}_p^{m\times n}$ and $\mathcal{B}\in\mathbb{K}_p^{m\times s}$, we call that the tube fiber column vectors in $\mathcal{B}$, i.e., $\{\mathcal{B}_{\cdot1},\mathcal{B}_{\cdot2},\ldots,\mathcal{B}_{\cdot s}\}$, is an orthogonal basis for the tube fiber columns of $\mathcal{A}$ in the sense of $\circledast_Q$, if $\mathcal{B}$ is partially $\circledast_Q$-orthogonal, and $\mathcal{A}=\mathcal{B}\circledast_Q\mathcal{C}$ for some $\mathcal{C}\in\mathbb{K}_p^{s\times n}$, which is equivalent to $\mathcal{A}=\mathcal{B}\circledast_Q\mathcal{B}^\top\circledast_Q\mathcal{A}$.

Proposition 2.4. *Let $Q\in\mathbb{K}^{p\times p}$ be an orthogonal matrix. For any given $\mathcal{A}\in\mathbb{K}_p^{m\times n}$ and $\mathcal{B}\in\mathbb{K}_p^{n\times t}$, it holds that*

$$\min_{1\le k\le p}\sigma_{\min}\big(\bar{A}_{(k)}\big)\|\mathcal{B}\|_F\le\big\|\mathcal{A}\circledast_Q\mathcal{B}\big\|_F\le\max_{1\le k\le p}\sigma_{\max}\big(\bar{A}_{(k)}\big)\|\mathcal{B}\|_F.$$

Proof. It follows from (2.1) that

$$\|\mathcal{A}\circledast\mathcal{B}\|_F^2=\sum_{k=1}^{p}\big\|\bar{A}_{(k)}\bar{B}_{(k)}\big\|_F^2\quad\text{and}\quad\|\mathcal{B}\|_F^2=\sum_{k=1}^{p}\big\|\bar{B}_{(k)}\big\|_F^2.$$

By matrix theory, we know

$$\sigma_{\min}^2(\bar{A}_{(k)})\|\bar{B}_{(k)}\|_F^2\le\|\bar{A}_{(k)}\bar{B}_{(k)}\|_F^2\le\sigma_{\max}^2(\bar{A}_{(k)})\|\bar{B}_{(k)}\|_F^2,\quad\text{for }k=1,2,\ldots,p.$$

Consequently, by the expressions above, we obtain the desired result and complete the proof. □

Proposition 2.5. *For given orthogonal matrices $\tilde{U} \in \mathbb{K}^{m\times m}$ and $\tilde{V} \in \mathbb{K}^{n\times n}$, we have*

$$\partial\|Y\|_* = \big\{\tilde{U}S\tilde{V} \mid S \in \partial\|X\|_*\big\} \quad \text{and} \quad \partial(\mathrm{rank}(Y)) = \big\{\tilde{U}S\tilde{V} \mid S \in \partial(\mathrm{rank}(X))\big\},$$

where $\mathrm{rank}(\cdot)$ *represents the rank function,* $X \in \mathbb{K}^{m\times n}$ *and* $Y = \tilde{U}X\tilde{V}$.

Proof. It is obvious from [14, Definition 5.2] that $\|\cdot\|_*$ and $\mathrm{rank}(\cdot)$ are singular value functions. Since $\tilde{U}$ and $\tilde{V}$ are orthogonal, it holds that $\sigma(Y) = \sigma(X)$. Moreover, $\tilde{U}U\mathrm{Diag}(\sigma(X))V\tilde{V}$ is a singular value decomposition (SVD) of Y, provided that $U\mathrm{Diag}(\sigma(X))V$ is a SVD of X. Hence, the desired formulas follow from [14, Theorem 7.1]. □

3 Model and Algorithm

In this section, we introduce a new low-rank prior tensor linear regression model. Then, we propose an implementable algorithm to solve the underlying model. Finally, we further establish the convergence result for the proposed algorithm.

3.1 Low-rank prior tensor linear regression model

Given a set of training tensor samples $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_q \in \mathbb{K}_p^{m\times n}$ and a test tensor sample $\mathcal{B} \in \mathbb{K}_p^{m\times n}$, we consider $\mathcal{B}$ is a linear combination of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_q$ in the sense of $\odot_Q$-product, i.e.,

$$\mathcal{B} = \boldsymbol{x}_1 \odot_Q \mathcal{A}_1 + \boldsymbol{x}_2 \odot_Q \mathcal{A}_2 + \ldots + \boldsymbol{x}_q \odot_Q \mathcal{A}_q + \mathcal{E}, \tag{3.1}$$

where $\boldsymbol{x}_i \in \mathbb{K}^p$ $(i = 1, 2, \ldots, q)$ are tubal fiber coefficients and $\mathcal{E}$ is the representation residual, which is assumed to obey a Gaussian distribution. Recalling the definitions of operator $\mathrm{Vec}(\cdot)$ converting a tensor into a tube vector, the $\odot_Q$-product and $\circledast_Q$-product, we have

$$\begin{aligned}
&\boldsymbol{x}_1 \odot_Q \mathcal{A}_1 + \boldsymbol{x}_2 \odot_Q \mathcal{A}_2 + \cdots + \boldsymbol{x}_q \odot_Q \mathcal{A}_q \\
&\quad = \boldsymbol{x}_1 \odot_Q \mathrm{Vec}(\mathcal{A}_1) + \boldsymbol{x}_2 \odot_Q \mathrm{Vec}(\mathcal{A}_2) + \cdots + \boldsymbol{x}_q \odot_Q \mathrm{Vec}(\mathcal{A}_q) \\
&\quad = \mathcal{A} \circledast_Q \boldsymbol{X},
\end{aligned}$$

where $\mathcal{A} = \big[\mathrm{Vec}(\mathcal{A}_1), \mathrm{Vec}(\mathcal{A}_2), \ldots, \mathrm{Vec}(\mathcal{A}_q)\big] \in \mathbb{K}_p^{mn\times q}$ and $\boldsymbol{X} = [\boldsymbol{x}_1; \boldsymbol{x}_2; \ldots; \boldsymbol{x}_q] \in \mathbb{K}_p^q$. Consequently, (3.1) can be written into a compact form as follows

$$\boldsymbol{B} = \mathcal{A} \circledast_Q \boldsymbol{X} + \boldsymbol{E}, \tag{3.2}$$

where $\boldsymbol{B}$ and $\boldsymbol{E}$ are tube matrices of $\mathcal{B}$ and $\mathcal{E}$, respectively. Clearly, (3.2) is a natural extension of the matrix linear regression, which is a special case with setting $p = 1$. Unlike the model (1.1), we use vector coefficients $\boldsymbol{x}_i$ instead of scalars so that the information of each frontal slice of all training tensors is possibly to be considered with different weights. On the other hand, the appearance of transformation Q is possibly able to explore the inherent property. Therefore, our model are comparatively more flexible and reasonable than the model (1.1).

Generally speaking, it is not an ideal way to find the coefficient matrix $\boldsymbol{X}$ by directly solving (3.2). Moreover, from the practical applications (e.g., see [24, 16]), the coefficient matrix $\boldsymbol{X}$ often has a low-rank structure, which at least is what we expected. Accordingly, we are concerned with the following model:

$$\min_{\boldsymbol{X}\in\mathbb{K}_p^q} \quad \mathrm{rank}(\boldsymbol{X}) + \frac{\alpha}{2}\big\|\mathcal{A} \circledast_Q \boldsymbol{X} - \boldsymbol{B}\big\|_F^2, \tag{3.3}$$

where $\alpha > 0$ is a tuning parameter. However, solving model (3.3) is NP-hard due to the appearance of the rank function. Therefore, we directly follow the idea of [1, 2, 6, 21] by using the classical nuclear norm to approximate the rank function rank($\boldsymbol{X}$). Moreover, to avoid over-fitting, we follow the spirit of ridge regression to impose a Tikhonov regularization term on the coefficient matrix $\boldsymbol{X}$. Specifically, we consider the following doubly regularized minimization model:

$$\min_{\boldsymbol{X}\in\mathbb{K}_p^q} \quad \|\boldsymbol{X}\|_* + \frac{\alpha}{2}\big\|\mathcal{A}\circledast_Q \boldsymbol{X} - \boldsymbol{B}\big\|_F^2 + \frac{\tau}{2}\|\boldsymbol{X}\|_F^2, \tag{3.4}$$

where $\tau > 0$ is a Tikhonov regularization parameter. Revisiting the model (1.1), we notice that the low-rank promoting term $\|\mathcal{E}\|_{\mathrm{TNN}}$ requires the full size of tensor $\mathcal{E}$, which would suffer from expensive computational cost when $\mathcal{E}$ is of big size. Comparatively, the coefficient matrix $\boldsymbol{X}$ in our model usually has much smaller size than the tensor $\mathcal{E}$. Therefore, our model often enjoys lower computational cost than (1.1) when dealing with large-scale problems.

3.2 Algorithm

To find a numerical solution of (3.4), we observe that the nonsmooth nuclear norm term makes (3.4) intractable. However, the good news is that the last two terms of (3.4) are differentiable. Therefore, to separate the nonsmooth and smooth terms, we introduce an auxiliary variable $\boldsymbol{Y} \in \mathbb{K}_p^q$ and rewrite (3.4) as the following separable minimization problem:

$$\begin{array}{ll} \min\limits_{\boldsymbol{X},\boldsymbol{Y}} & \|\boldsymbol{Y}\|_* + \dfrac{\alpha}{2}\|\mathcal{A}\circledast_Q \boldsymbol{X} - \boldsymbol{B}\|_F^2 + \dfrac{\tau}{2}\|\boldsymbol{X}\|_F^2 \\ \text{s.t.} & \boldsymbol{X} = \boldsymbol{Y}. \end{array} \tag{3.5}$$

Clearly, (3.5) is a linearly constrained convex optimization problem. As we know, one of the most popular solvers to solve (3.5) is the well-known ADMM. Here, we first recall the augmented Lagrangian function associated with (3.5), which is given by

$$\mathscr{L}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{U}) = \|\boldsymbol{Y}\|_* + \frac{\alpha}{2}\|\mathcal{A}\circledast_Q \boldsymbol{X} - \boldsymbol{B}\|_F^2 + \frac{\tau}{2}\|\boldsymbol{X}\|_F^2 + \big\langle \boldsymbol{U}, \boldsymbol{X}-\boldsymbol{Y}\big\rangle + \frac{\beta}{2}\big\|\boldsymbol{X}-\boldsymbol{Y}\big\|_F^2, \tag{3.6}$$

where $\boldsymbol{U}$ is the Lagrangian multipliers and $\beta > 0$ is a penalty parameter. Below, we shall show the customized implementation of ADMM to (3.5). Since the $\boldsymbol{Y}$-part subproblem amounts to a proximal operator, which is simpler than the $\boldsymbol{X}$-part, we then update variables in order $\boldsymbol{Y} \to \boldsymbol{X} \to \boldsymbol{U}$. More specifically, for given the l-th iterate $\big(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l\big)$, the iterative scheme of ADMM for (3.5) reads as

$$\begin{cases} \boldsymbol{Y}^{l+1} = \arg\min\limits_{\boldsymbol{Y}} \mathscr{L}\big(\boldsymbol{X}^l, \boldsymbol{Y}, \boldsymbol{U}^l\big), \\ \boldsymbol{X}^{l+1} = \arg\min\limits_{\boldsymbol{X}} \mathscr{L}\big(\boldsymbol{X}, \boldsymbol{Y}^{l+1}, \boldsymbol{U}^l\big), \\ \boldsymbol{U}^{l+1} = \boldsymbol{U}^l + \beta\big(\boldsymbol{X}^{l+1} - \boldsymbol{Y}^{l+1}\big). \end{cases} \tag{3.7}$$

Below, we present the concrete updating scheme of each variable and show that our algorithm enjoys closed-form solutions for each subproblem.

The $\boldsymbol{Y}$ subproblem: For the l-th iteration, the $\boldsymbol{Y}$ subproblem is specified as

$$\boldsymbol{Y}^{l+1} = \arg\min_{\boldsymbol{Y}} \mathscr{L}\big(\boldsymbol{X}^l, \boldsymbol{Y}, \boldsymbol{U}^l\big)$$

$$= \arg\min_{\boldsymbol{Y}} \left\{ \|\boldsymbol{Y}\|_* + \frac{\beta}{2}\left\|\boldsymbol{X}^l - \boldsymbol{Y} + (1/\beta)\boldsymbol{U}^l\right\|_F^2 \right\}$$
$$= \mathbf{SVT}\left(\boldsymbol{X}^l + (1/\beta)\boldsymbol{U}^l, 1/\beta\right), \tag{3.8}$$

where $\mathbf{SVT}(\cdot,\cdot)$ is the well-known singular value shrinkage operator defined by

$$\mathbf{SVT}(M,\tau) = U\mathrm{shrink}_\tau(\Sigma)V^\top$$

with $M = U\Sigma V^\top$ being the SVD of matrix $M \in \mathbb{K}^{m\times n}$ of rank r in the reduced form, U and V are $m \times r$ and $n \times r$ matrices with orthogonal columns, respectively, $\Sigma = \mathrm{diag}\left(\{\sigma_i\}_{1\le i\le r}\right)$ is a diagonal matrix and $\mathrm{shrink}_\tau(\Sigma) = \mathrm{diag}\left(\max\{\sigma_i - \tau, 0\}\right)$ for $\tau \ge 0$.

The $\boldsymbol{X}$-subproblem: Updating $\boldsymbol{X}^{l+1}$ amounts to solving the following optimization problem:

$$\min_{\boldsymbol{X}\in\mathbb{K}_p^q} \left\{ \frac{\alpha}{2}\left\|\mathcal{A}\circledast_Q \boldsymbol{X} - \boldsymbol{B}\right\|_F^2 + \frac{\tau}{2}\left\|\boldsymbol{X}\right\|_F^2 + \frac{\beta}{2}\left\|\boldsymbol{X} - \boldsymbol{Y}^{l+1} + (1/\beta)\boldsymbol{U}^l\right\|_F^2 \right\},$$

which is equivalent to

$$\min_{\bar{X}_{(k)}} \left\{ \frac{\alpha}{2}\sum_{k=1}^p \left\|\bar{A}_{(k)}\bar{X}_{(k)} - \bar{B}_{(k)}\right\|_F^2 + \frac{\tau}{2}\sum_{k=1}^p \left\|\bar{X}_{(k)}\right\|_F^2 + \frac{\beta}{2}\sum_{k=1}^p \left\|\bar{X}_{(k)} - \bar{Y}_{(k)}^{l+1} + \frac{1}{\beta}\bar{U}_{(k)}^l\right\|_F^2 \right\}, \tag{3.9}$$

where $\bar{A}_{(k)}, \bar{B}_{(k)}, \bar{X}_{(k)}, \bar{Y}_{(k)}^{l+1}$ and $\bar{U}_{(k)}^l$ are the k-th frontal slices of $\bar{\mathcal{A}}, \bar{\boldsymbol{B}}, \bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}^{l+1}$ and $\bar{\boldsymbol{U}}^l$, respectively. Thanks to the full separable structure with respect to $\bar{X}_{(k)}$ for $k = 1, 2, \ldots, p$, solving (3.9) amounts to individually finding a solution of

$$\min_{\bar{X}_{(k)}\in\mathbb{K}_p^q} \left\{ \frac{\alpha}{2}\left\|\bar{A}_{(k)}\bar{X}_{(k)} - \bar{B}_{(k)}\right\|_F^2 + \frac{\tau}{2}\left\|\bar{X}_{(k)}\right\|_F^2 + \frac{\beta}{2}\left\|\bar{X}_{(k)} - \bar{Y}_{(k)}^{l+1} + \frac{1}{\beta}\bar{U}_{(k)}^l\right\|_F^2 \right\}. \tag{3.10}$$

By the optimality condition of (3.10), it is easy to see that for $k = 1, 2, \ldots, p$,

$$\alpha\bar{A}_{(k)}^\top\left(\bar{A}_{(k)}\bar{X}_{(k)}^{l+1} - \bar{B}_{(k)}\right) + \tau\bar{X}_{(k)}^{l+1} + \beta\left(\bar{X}_{(k)}^{l+1} - \bar{Y}_{(k)}^{l+1} + \frac{1}{\beta}\bar{U}_{(k)}^l\right) = 0, \tag{3.11}$$

which implies

$$\bar{X}_{(k)}^{l+1} = \left((\tau+\beta)I + \alpha\bar{A}_{(k)}^\top\bar{A}_{(k)}\right)^{-1}\left(\alpha\bar{A}_{(k)}^\top\bar{B}_{(k)} + \beta\bar{Y}_{(k)}^{l+1} - \bar{U}_{(k)}^l\right)$$

for $k = 1, 2, \ldots, p$. Therefore, we have

$$\boldsymbol{X}^{l+1} = \left((\tau+\beta)\mathcal{I} + \alpha\mathcal{A}^\top\circledast_Q\mathcal{A}\right)^{-1}\circledast_Q\left(\alpha\mathcal{A}^\top\circledast_Q\mathcal{B} + \beta\boldsymbol{Y}^{l+1} - \boldsymbol{U}^l\right). \tag{3.12}$$

Note that $\bar{U}_{(k)}^{l+1} = \bar{U}_{(k)}^l + \beta\left(\bar{X}_{(k)}^{l+1} - \bar{Y}_{(k)}^{l+1}\right)$ for $k = 1, 2, \ldots, p$. It then follows from (3.11) that

$$\alpha\bar{A}_{(k)}^\top\left(\bar{A}_{(k)}\bar{X}_{(k)}^{l+1} - \bar{B}_{(k)}\right) + \tau\bar{X}_{(k)}^{l+1} + \bar{U}_{(k)}^{l+1} = 0,$$

which immediately implies that, for any $l \ge 0$,

$$\boldsymbol{U}^{l+1} = \alpha\mathcal{A}^\top\circledast_Q\left(\boldsymbol{B} - \mathcal{A}\circledast_Q\boldsymbol{X}^{l+1}\right) - \tau\boldsymbol{X}^{l+1}. \tag{3.13}$$

Clearly, (3.13) provides an alternative updating formula for $\boldsymbol{U}^{l+1}$ without $\boldsymbol{Y}^{l+1}$. In this situation, the $\boldsymbol{Y}$ can be regarded as an intermediate variable.

Algorithm 1 ADMM for solving (3.5)

Input: $\alpha > 0$, $\beta > 0$, and $\tau > 0$. Choose starting points $(\boldsymbol{X}^0, \boldsymbol{U}^0)$.

1: **for** $l = 0, 1, 2, \cdots$ **do**
2: Update $\boldsymbol{Y}^{l+1}$ via (3.8);
3: Update $\boldsymbol{X}^{l+1}$ via (3.12);
4: Update $\boldsymbol{U}^{l+1}$ via the third one of (3.7) or (3.13);
5: **end for**

Output: Optimal regression coefficient matrix $\widehat{\boldsymbol{X}}$.

With the above preparations, we formally summarize the updating schemes for model (3.5) in Algorithm 1.

When implementing Algorithm 1, we shall set a stopping criterion. Actually, it is not difficult to see that the first-order optimality conditions for (3.5) can be written as the following

$$\begin{cases} \partial\|\boldsymbol{Y}\|_* - \boldsymbol{U} \ni 0, \\ \alpha\mathcal{A}^\top \circledast_Q \left(\mathcal{A} \circledast_Q \boldsymbol{X} - \boldsymbol{B}\right) + \tau\boldsymbol{X} + \boldsymbol{U} = 0, \\ \boldsymbol{X} - \boldsymbol{Y} = 0. \end{cases} \tag{3.14}$$

Moreover, we call the triple $\left(\boldsymbol{X}^*, \boldsymbol{Y}^*, \boldsymbol{U}^*\right)$ satisfying (3.14) a stationary point of (3.5). In practice, we can use the following termination criterion:

$$\max\left\{\left\|\boldsymbol{X}^{l+1} - \boldsymbol{Y}^{l+1}\right\|_F,\ \left\|\mathcal{A} \circledast_Q \boldsymbol{X}^{l+1} - \boldsymbol{B}\right\|_F\right\} \le \epsilon, \tag{3.15}$$

where $\epsilon > 0$ is a preset precision.

3.3 Convergence analysis

In this subsection, we analyze the convergence of Algorithm 1 and begin this part with the following lemma.

Lemma 3.1. *Let $\left\{\left(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l\right)\right\}_{l=0}^{\infty}$ be the sequence generated by Algorithm 1. Then, we have*

$$\mathscr{L}\left(\boldsymbol{X}^l, \boldsymbol{Y}^{l+1}, \boldsymbol{U}^l\right) \le \mathscr{L}\left(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l\right). \tag{3.16}$$

Proof. Since $\boldsymbol{Y}^{l+1}$ is the optimal solution of the $\boldsymbol{Y}$-subproblem (i.e., (3.8)), we know

$$\left\|\boldsymbol{Y}^{l+1}\right\|_* + \frac{\beta}{2}\left\|\boldsymbol{X}^l - \boldsymbol{Y}^{l+1} + \frac{1}{\beta}\boldsymbol{U}^l\right\|_F^2 \le \left\|\boldsymbol{Y}^l\right\|_* + \frac{\beta}{2}\left\|\boldsymbol{X}^l - \boldsymbol{Y}^l + \frac{1}{\beta}\boldsymbol{U}^l\right\|_F^2,$$

which, together with the definition of $\mathscr{L}$, implies the desired result.

□

Lemma 3.2. *Let $\left\{\left(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l\right)\right\}_{l=0}^{\infty}$ be the sequence generated by Algorithm 1. It holds that*

$$\mathscr{L}\left(\boldsymbol{X}^{l+1}, \boldsymbol{Y}^{l+1}, \boldsymbol{U}^l\right) \le \mathscr{L}\left(\boldsymbol{X}^l, \boldsymbol{Y}^{l+1}, \boldsymbol{U}^l\right) - \frac{\zeta}{2}\left\|\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\right\|_F^2, \tag{3.17}$$

where $\zeta = \tau + \beta + \alpha\ \bar{\lambda}_{\min}$ with $\bar{\lambda}_{\min} = \min\limits_{1\le k\le p} \lambda_{\min}\left(\bar{A}_{(k)}^\top \bar{A}_{(k)}\right)$.

Proof. For every $k = 1, 2, \ldots, p$ and $l \geq 1$, we define $\varphi_{lk} : \mathbb{K}^q \to \mathbb{K}$ by

$$\varphi_{lk}(\bar{X}_{(k)}) = \frac{\alpha}{2}\|\bar{A}_{(k)}\bar{X}_{(k)} - \bar{B}_{(k)}\|_F^2 + \frac{\tau}{2}\|\bar{X}_{(k)}\|_F^2 + \frac{\beta}{2}\|\bar{X}_{(k)} - \bar{Y}_{(k)}^{l+1} + \frac{1}{\beta}\bar{U}^l\|_F^2.$$

It is obvious that $\varphi_{lk}(\cdot)$ is strongly convex with modulus at least $\zeta_k := \tau+\beta+\alpha\lambda_{\min}\big(\bar{A}_{(k)}^\top\bar{A}_{(k)}\big)$ on $\mathbb{K}^q$, which implies

$$\varphi_{lk}(\bar{X}'_{(k)}) \geq \varphi_{lk}(\bar{X}_{(k)}) + \big\langle\nabla\varphi_{lk}(\bar{X}_{(k)}), \bar{X}_{(k)} - \bar{X}'_{(k)}\big\rangle + \frac{\zeta_k}{2}\|\bar{X}'_{(k)} - \bar{X}_{(k)}\|_F^2$$

for any $\bar{X}'_{(k)}, \bar{X}_{(k)} \in \mathbb{K}^q$. Since $\bar{X}_{(k)}^{l+1}$ minimizes $\varphi_{lk}(\bar{X}_{(k)})$, it holds that $\nabla\varphi_{lk}\big(\bar{X}_{(k)}^{l+1}\big) = 0$. Consequently, it holds that

$$\varphi_{lk}\big(\bar{X}_{(k)}^{l}\big) \geq \varphi_{l}\big(\bar{X}_{(k)}^{l+1}\big) + \frac{\zeta_k}{2}\big\|\bar{X}_{(k)}^{l+1} - \bar{X}_{(k)}^{l}\big\|_F^2,$$

which implies

$$\mathscr{L}\big(\boldsymbol{X}^l, \boldsymbol{Y}^{l+1}, \boldsymbol{U}^l\big) \geq \mathscr{L}\big(\boldsymbol{X}^{l+1}, \boldsymbol{Y}^{l+1}, \boldsymbol{U}^l\big) + \frac{\zeta}{2}\big\|\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\big\|_F^2,$$

where $\zeta = \min\{\zeta_k \mid k = 1, 2, \ldots, p\}$. We obtain the desired result and complete the proof. □

Proposition 3.3. *Let $\big\{\big(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l\big)\big\}_{l=0}^{\infty}$ be a sequence generated by Algorithm 1. Let $c_0 := \mathscr{L}\big(\boldsymbol{X}^0, \boldsymbol{Y}^0, \boldsymbol{U}^0\big)$ and $\bar{\sigma}_{\max} = \max\limits_{1\leq k\leq p}\sigma_{\max}\big((\bar{A}_{(k)})^\top\bar{A}_{(k)}\big)$. We have the following conclusions:*

(a). *If $\beta\big(\tau + \beta + \alpha\ \bar{\lambda}_{\min}\big) - 2\big(\tau + \alpha\ \bar{\sigma}_{\max}\big)^2 > 0$, then the sequence $\big\{\big(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l\big)\big\}_{l=0}^{\infty}$ satisfies the following formula*

$$\mathscr{L}\big(\boldsymbol{X}^{l+1}, \boldsymbol{Y}^{l+1}, \boldsymbol{U}^{l+1}\big) + \xi\big\|\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\big\|_F^2 \leq \mathscr{L}\big(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l\big)$$

for any $l \geq 1$, where $\xi = 1/(2\beta)\Big\{\beta\big(\tau + \beta + \alpha\ \bar{\lambda}_{\min}\big) - 2\big(\tau + \alpha\ \bar{\sigma}_{\max}\big)^2\Big\}$.

(b). *If $\beta\big(\tau + \beta + \alpha\bar{\lambda}_{\min}\big) - 2\big(\tau + \alpha\bar{\sigma}_{\max}\big)^2 > 0$ and $\tau\beta - 2\left(\tau + \alpha\bar{\sigma}_{\max}\right)^2 > 0$, then the sequence $\big\{\big(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l\big)\big\}_{l=0}^{\infty}$ is bounded.*

Proof. First, by the update formula of Lagrangian multiplier in (3.7), we know that for any $l \geq 0$,

$$\mathscr{L}\big(\boldsymbol{X}^{l+1}, \boldsymbol{Y}^{l+1}, \boldsymbol{U}^{l+1}\big) = \mathscr{L}\big(\boldsymbol{X}^{l+1}, \boldsymbol{Y}^{l+1}, \boldsymbol{U}^l\big) + \frac{1}{\beta}\big\|\boldsymbol{U}^{l+1} - \boldsymbol{U}^l\big\|_F^2,$$

which, together with Lemmas 3.1 and 3.2, implies that, for any $l \geq 0$,

$$\mathscr{L}\big(\boldsymbol{X}^{l+1}, \boldsymbol{Y}^{l+1}, \boldsymbol{U}^{l+1}\big) \leq \mathscr{L}\big(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l\big) - \frac{\zeta}{2}\big\|\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\big\|_F^2 + \frac{1}{\beta}\big\|\boldsymbol{U}^{l+1} - \boldsymbol{U}^l\big\|_F^2. \quad (3.18)$$

On the other hand, it follows from (3.13) that

$$\begin{aligned}\big\|\boldsymbol{U}^{l+1} - \boldsymbol{U}^l\big\|_F^2 &= \big\|\big(\tau\mathcal{I} + \alpha\mathcal{A}^\top \circledast_Q \mathcal{A}\big) \circledast_Q \big(\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\big)\big\|_F^2 \\ &\leq \max_{1\leq k\leq p}\sigma_{\max}^2\big(\tau I + \alpha\bar{A}_{(k)}^\top\bar{A}_{(k)}\big)\big\|\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\big\|_F^2\end{aligned}$$

$$= \big(\tau + \alpha\ \bar{\sigma}_{\max}\big)^2 \big\|\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\big\|_F^2, \tag{3.19}$$

where the inequality comes from Proposition 2.4. Consequently, it holds from (3.18) and (3.19) that

$$\begin{aligned}
&\mathscr{L}\big(\boldsymbol{X}^{l+1}, \boldsymbol{Y}^{l+1}, \boldsymbol{U}^{l+1}\big) \\
&\quad \le \mathscr{L}\big(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l\big) - \left(\frac{\zeta}{2} - \frac{\big(\tau + \alpha\ \bar{\sigma}_{\max}\big)^2}{\beta}\right) \big\|\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\big\|_F^2 \\
&\quad = \mathscr{L}\big(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l\big) - \frac{\beta\big(\tau + \beta + \alpha\ \bar{\lambda}_{\min}\big) - 2\big(\tau + \alpha\ \bar{\sigma}_{\max}\big)^2}{2\beta} \big\|\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\big\|_F^2.
\end{aligned}$$

Therefore, we obtain the conclusion (a).

Now we prove (b). By (3.13), it holds that

$$\begin{aligned}
\big\|\boldsymbol{U}^l\big\|_F^2 &\le \big(\big\|\alpha\mathcal{A}^\top \circledast_Q \boldsymbol{B}\big\|_F + \big\|(\tau\mathcal{I} + \alpha\mathcal{A}^\top \circledast_Q \mathcal{A}) \circledast_Q \boldsymbol{X}^l\big\|_F\big)^2 \\
&\le 2\alpha^2\big\|\mathcal{A}^\top \circledast_Q \boldsymbol{B}\big\|_F^2 + 2\big(\tau + \alpha\bar{\sigma}_{\max}\big)^2\big\|\boldsymbol{X}^l\big\|_F^2,
\end{aligned}$$

where the second inequality comes from Proposition 2.4. As a consequence, we have

$$\begin{aligned}
\mathscr{L}\big(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l\big) &= \big\|\boldsymbol{Y}^l\big\|_* + \frac{\alpha}{2}\big\|\mathcal{A} \circledast_Q \boldsymbol{X}^l - \boldsymbol{B}\big\|_F^2 + \frac{\tau}{2}\big\|\boldsymbol{X}^l\big\|_F^2 \\
&\quad + \frac{\beta}{2}\big\|\boldsymbol{X}^l - \boldsymbol{Y}^l + \frac{1}{\beta}\boldsymbol{U}^l\big\|_F^2 - \frac{1}{2\beta}\big\|\boldsymbol{U}^l\big\|_F^2 \\
&\ge \big\|\boldsymbol{Y}^l\big\|_* + \frac{\alpha}{2}\big\|\mathcal{A} \circledast_Q \boldsymbol{X}^l - \boldsymbol{B}\big\|_F^2 + \frac{\tau}{2}\big\|\boldsymbol{X}^l\big\|_F^2 \\
&\quad + \frac{\beta}{2}\big\|\boldsymbol{X}^l - \boldsymbol{Y}^l + \frac{1}{\beta}\boldsymbol{U}^l\big\|_F^2 - \frac{\alpha^2}{\beta}\big\|\mathcal{A}^\top \circledast_Q \boldsymbol{B}\big\|_F^2 - \frac{1}{\beta}\big(\tau + \alpha\bar{\sigma}_{\max}\big)^2\big\|\boldsymbol{X}^l\big\|_F^2 \\
&= \big\|\boldsymbol{Y}^l\big\|_* + \frac{\alpha}{2}\big\|\mathcal{A} \circledast_Q \boldsymbol{X}^l - \boldsymbol{B}\big\|_F^2 + \left(\frac{\tau}{2} - \frac{1}{\beta}\big(\tau + \alpha\bar{\sigma}_{\max}\big)^2\right)\big\|\boldsymbol{X}^l\big\|_F^2 \\
&\quad + \frac{\beta}{2}\big\|\boldsymbol{X}^l - \boldsymbol{Y}^l + \frac{1}{\beta}\boldsymbol{U}^l\big\|_F^2 - \frac{\alpha^2}{\beta}\big\|\mathcal{A}^\top \circledast_Q \boldsymbol{B}\big\|_F^2.
\end{aligned} \tag{3.20}$$

By conclusion (a), we know that $\mathscr{L}\big(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l\big) \le c_0$ for any $l \ge 0$, which, together with (3.20), implies that

$$\begin{aligned}
&\big\|\boldsymbol{Y}^l\big\|_* + \frac{\alpha}{2}\big\|\mathcal{A} \circledast_Q \boldsymbol{X}^l - \boldsymbol{B}\big\|_F^2 \\
&\qquad + \left(\frac{\tau}{2} - \frac{1}{\beta}\big(\tau + \alpha\bar{\sigma}_{\max}\big)^2\right)\big\|\boldsymbol{X}^l\big\|_F^2 + \frac{\beta}{2}\big\|\boldsymbol{X}^l - \boldsymbol{Y}^l + \frac{1}{\beta}\boldsymbol{U}^l\big\|_F^2 \\
&\qquad\qquad \le c_0 + \frac{\alpha^2}{\beta}\big\|\mathcal{A}^\top \circledast_Q \boldsymbol{B}\big\|_F^2 := c_1.
\end{aligned} \tag{3.21}$$

By (3.21) and the given condition, we have $\big\|\boldsymbol{X}^l\big\|_F \le \sqrt{2\beta c_1/\rho}$ with $\rho := \tau\beta - 2\,(\tau + \alpha\bar{\sigma}_{\max})^2$ and $\big\|\boldsymbol{Y}^l\big\|_* \le c_1$ for any $l \ge 0$. Hence, we know $\big\|\boldsymbol{Y}^l\big\|_F \le \big\|\boldsymbol{Y}^l\big\|_* \le c_1$ for any $l \ge 0$. Consequently, by using (3.21) again, it holds that

$$\big\|\frac{1}{\beta}\boldsymbol{U}^l\big\|_F \le \big\|\boldsymbol{X}^l - \boldsymbol{Y}^l + \frac{1}{\beta}\boldsymbol{U}^l\big\|_F + \big\|\boldsymbol{X}^l\big\|_F + \big\|\boldsymbol{Y}^l\big\|_F \le \sqrt{2c_1/\beta} + \sqrt{2c_1\beta/\rho} + c_1,$$

which implies $\big\|\boldsymbol{U}^l\big\|_F \le \sqrt{2c_1\beta} + \sqrt{2c_1\beta^3/\rho} + \beta c_1$. We obtain the conclusion (b). The proof is completed. □

We are now ready to prove our global convergence result for Algorithm 1.

Theorem 3.4. *Let* $\{(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l)\}_{l=0}^{\infty}$ *be a sequence generated by Algorithm 1. If* $\beta(\tau + \beta + \alpha\ \bar{\lambda}_{\min}) - 2(\tau + \alpha\ \bar{\sigma}_{\max})^2 > 0$ *and* $\tau\beta - 2\,(\tau + \alpha\bar{\sigma}_{\max})^2 > 0$*, then we have the following conclusions:*

(a). $\lim_{l\to\infty} \|\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\|_F + \|\boldsymbol{Y}^{l+1} - \boldsymbol{Y}^l\|_F + \|\boldsymbol{U}^{l+1} - \boldsymbol{U}^l\|_F = 0.$

(b). *Any cluster point* $(\boldsymbol{X}^*, \boldsymbol{Y}^*, \boldsymbol{U}^*)$ *of* $\{(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l)\}_{l=1}^{\infty}$ *is a stationary point of* (3.5).

Proof. With the given conditions, by Proposition 3.3, we immediately obtain the boundedness of the sequence $\{(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l)\}_{l=1}^{\infty}$, which implies a cluster point exists.

Now, we first prove statement (a). Suppose that $(\boldsymbol{X}^*, \boldsymbol{Y}^*, \boldsymbol{U}^*)$ is a cluster point of the sequence $\{(\boldsymbol{X}^l, \boldsymbol{Y}^l, \boldsymbol{U}^l)\}_{l=1}^{\infty}$ generated by Algorithm 1, and let $\{(\boldsymbol{X}^{l_i}, \boldsymbol{Y}^{l_i}, \boldsymbol{U}^{l_i})\}_{i=1}^{\infty}$ be a convergent subsequence such that $\lim_{i\to\infty}(\boldsymbol{X}^{l_i}, \boldsymbol{Y}^{l_i}, \boldsymbol{U}^{l_i}) = (\boldsymbol{X}^*, \boldsymbol{Y}^*, \boldsymbol{U}^*)$. By the statement (a) of Proposition 3.3, it holds that

$$\xi \sum_{l=1}^{l_i} \|\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\|_F^2 \le \mathscr{L}(\boldsymbol{X}^0, \boldsymbol{Y}^0, \boldsymbol{U}^0) - \mathscr{L}(\boldsymbol{X}^{l_i}, \boldsymbol{Y}^{l_i}, \boldsymbol{U}^{l_i}).$$

Consequently, by letting $l_i \to \infty$, we know

$$\xi \sum_{l=1}^{\infty} \|\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\|_F^2 \le \mathscr{L}(\boldsymbol{X}^0, \boldsymbol{Y}^0, \boldsymbol{U}^0) - \mathscr{L}(\boldsymbol{X}^*, \boldsymbol{Y}^*, \boldsymbol{U}^*) < \infty. \tag{3.22}$$

Hence, $\lim_{l\to\infty} \|\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\|_F = 0$. By (3.19) in the proof of Proposition 3.3, we know $\lim_{l\to\infty} \|\boldsymbol{U}^{l+1} - \boldsymbol{U}^l\|_F = 0$. Moreover, by the updating formula of $\boldsymbol{U}$ in (3.7), we have

$$\boldsymbol{Y}^{l+1} - \boldsymbol{Y}^l = \boldsymbol{X}^{l+1} - \boldsymbol{X}^l - \frac{1}{\beta}(\boldsymbol{U}^{l+1} - \boldsymbol{U}^l) + \frac{1}{\beta}(\boldsymbol{U}^l - \boldsymbol{U}^{l-1}),$$

which implies

$$\|\boldsymbol{Y}^{l+1} - \boldsymbol{Y}^l\|_F \le \|\boldsymbol{X}^{l+1} - \boldsymbol{X}^l\|_F + \frac{1}{\beta}\|\boldsymbol{U}^{l+1} - \boldsymbol{U}^l\|_F + \frac{1}{\beta}\|\boldsymbol{U}^l - \boldsymbol{U}^{l-1}\|_F.$$

Hence, we have $\lim_{l\to\infty} \|\boldsymbol{Y}^{l+1} - \boldsymbol{Y}^l\|_F = 0$. The conclusion (a) holds.

Hereafter, we prove (b). From the first optimality condition of $\boldsymbol{Y}$ subproblem (3.8) at the l_i-th iteration, we know

$$0 \in \partial\|\boldsymbol{Y}^{l_i}\|_* + \beta(\boldsymbol{Y}^{l_i} - \boldsymbol{X}^{l_i-1} - \frac{1}{\beta}\boldsymbol{U}^{l_i-1}) = \partial\|\boldsymbol{Y}^{l_i}\|_* - \boldsymbol{U}^{l_i} + \beta(\boldsymbol{X}^{l_i} - \boldsymbol{X}^{l_i-1}), \tag{3.23}$$

where the equality is due to the third expression in (3.7).

On the other hand, by the optimality condition of (3.10) at the l_i-th iteration, it is easy to see that for $k = 1, 2, \ldots, p$,

$$\alpha\bar{A}_{(k)}^{\top}\left(\bar{A}_{(k)}\bar{X}_{(k)}^{l_i} - \bar{B}_{(k)}\right) + \tau\bar{X}_{(k)}^{l_i} + \beta\left(\bar{X}_{(k)}^{l_i} - \bar{Y}_{(k)}^{l_i} + \frac{1}{\beta}\bar{U}_{(k)}^{l_i-1}\right) = 0,$$

which, together with the update formula of $\boldsymbol{U}$ (i.e., $\bar{U}_{(k)}^{l_i} = \bar{U}_{(k)}^{l_i-1} + \beta\big(\bar{X}_{(k)}^{l_i} - \bar{Y}_{(k)}^{l_i}\big)$ for $k = 1, 2, \ldots, p$), implies

$$\alpha \bar{A}_{(k)}^{\top}\left(\bar{A}_{(k)}\bar{X}_{(k)}^{l_i} - \bar{B}_{(k)}\right) + \tau\bar{X}_{(k)}^{l_i} + \bar{U}_{(k)}^{l_i} = 0. \tag{3.24}$$

By (3.24), we know

$$\alpha \boldsymbol{\mathcal{A}}^{\top} \circledast_Q \left(\boldsymbol{\mathcal{A}} \circledast_Q \boldsymbol{X}^{l_i} - \boldsymbol{B}\right) + \tau \boldsymbol{X}^{l_i} + \boldsymbol{U}^{l_i} = 0. \tag{3.25}$$

Consequently, by letting $l_i \to \infty$ in (3.23) and (3.25), it holds that

$$\begin{cases} \partial\big\|\boldsymbol{Y}^*\big\|_* - \boldsymbol{U}^* \ni 0, \\ \alpha \boldsymbol{\mathcal{A}}^{\top} \circledast_Q \left(\boldsymbol{\mathcal{A}} \circledast_Q \boldsymbol{X}^* - \boldsymbol{B}\right) + \tau \boldsymbol{X}^* + \boldsymbol{U}^* = 0, \end{cases} \tag{3.26}$$

since $\big\|\boldsymbol{X}^{l_i} - \boldsymbol{X}^{l_i-1}\big\|_F \to 0$ as $l_i \to \infty$.

Finally, since $\boldsymbol{U}^{l_i} - \boldsymbol{U}^{l_i-1} = \beta\big(\boldsymbol{X}^{l_i} - \boldsymbol{Y}^{l_i}\big)$ and $\lim\limits_{l_i\to\infty}\big\|\boldsymbol{U}^{l_i} - \boldsymbol{U}^{l_i-1}\big\|_F = 0$, we know $\boldsymbol{X}^* - \boldsymbol{Y}^* = 0$, which, together with (3.26), implies that $\big(\boldsymbol{X}^*, \boldsymbol{Y}^*, \boldsymbol{U}^*\big)$ is a stationary point of (3.5). □

Remark 3.5. In Proposition 3.3 and Theorem 3.4, there are two assumptions $\beta\big(\tau + \beta + \alpha\bar{\lambda}_{\min}\big) - 2\big(\tau + \alpha\bar{\sigma}_{\max}\big)^2 > 0$ and $\tau\beta - 2\left(\tau + \alpha\bar{\sigma}_{\max}\right)^2 > 0$, where the first condition is used to ensure that the Lagrange function (3.6) is nonincreasing, and the second one guarantees the boundedness of the sequence $\{X^l\}$. In practice, these two assumptions can be easily satisfied. Note that the tensor $\boldsymbol{\mathcal{A}}$ is known, we can directly compute $\bar{\lambda}_{\min}$ and $\bar{\sigma}_{\max}$. Moreover, τ and α are regularization parameters in optimization problem (3.4), we could simply choose the penalty parameter β as large as possible to satisfy the assumptions.

4 Numerical Experiments

In this section, we apply the proposed method to color face classification and traffic flow data regression. All codes were written by MATLAB 2021b and all experiments were conducted on a laptop computer with Intel (R) core (TM) i7-7500 CPU @ 2.70GHz and 8GB memory.

4.1 Face classification

In this part, we apply our method to color face classification. Here, we refer the reader to [3, 19] for more details on face recognition and classification. For given a set of training sample images $\boldsymbol{\mathcal{A}}_1, \ldots, \boldsymbol{\mathcal{A}}_q \in \mathbb{K}_p^{m\times n}$, we could divide them into N classes and the i-th class has q_i ($i \in [N]$) samples, i.e.,

$$\{\boldsymbol{\mathcal{A}}_1, \ldots, \boldsymbol{\mathcal{A}}_q\} = \left\{ \underbrace{(\boldsymbol{\mathcal{A}}_1)_1, \ldots, (\boldsymbol{\mathcal{A}}_1)_{q_1}}_{\text{the 1st class}}, \ldots, \underbrace{(\boldsymbol{\mathcal{A}}_N)_1, \ldots, (\boldsymbol{\mathcal{A}}_N)_{q_N}}_{\text{the } N\text{th class}} \right\}.$$

Then, for a given new face image $\boldsymbol{\mathcal{B}} \in \mathbb{K}_p^{m\times n}$, it could be linearly represented by all training sample images under the transformed T-product, which corresponds to model (3.1). Therefore, we can obtain the coefficient matrix $\boldsymbol{X}$ via solving the optimization problem (3.5) by Algorithm 1. Here, we define a function $\psi_i : \mathbb{K}_p^q \to \mathbb{K}_p^q$ ($q = \sum_{i=1}^N q_i$) being the

characteristic function that selects the coefficients associated with the i-th class, i.e., for $\boldsymbol{X} = [(\boldsymbol{x}_1)_1; \ldots; (\boldsymbol{x}_1)_{q_1}; \ldots; (\boldsymbol{x}_i)_1; \ldots; (\boldsymbol{x}_i)_{q_i}; \ldots; (\boldsymbol{x}_N)_1; \ldots; (\boldsymbol{x}_N)_{q_N}] \in \mathbb{K}_p^q$, we define

$$\psi_i(\boldsymbol{X}) = [\boldsymbol{0}, \ldots, \boldsymbol{0}, (\boldsymbol{x}_i)_1, \ldots, (\boldsymbol{x}_i)_{q_i}, \boldsymbol{0}, \ldots, \boldsymbol{0}] \in \mathbb{K}_p^q.$$

Using the coefficients associated with the i-th class, we can get the reconstruction of test sample $\mathcal{B}$ in the i-th class as $\bar{\mathcal{B}}_i = \mathcal{A} \circledast_Q \psi_i(\boldsymbol{X})$, where $\mathcal{A} = [\mathrm{Vec}((\mathcal{A}_1)_1), \mathrm{Vec}((\mathcal{A}_1)_{q_1}), \ldots, \mathrm{Vec}((\mathcal{A}_N)_1), \ldots, \mathrm{Vec}((\mathcal{A}_N)_{q_N})]$. The i-th class's reconstruction error is defined by

$$\delta_i(\mathcal{B}) = \left\| \mathcal{B} - \bar{\mathcal{B}}_i \right\|_F^2. \tag{4.1}$$

Accordingly, when $\delta_j = \min_{1 \le i \le N} \delta_i(\mathcal{B})$, then $\mathcal{B}$ is assigned to class j.

Below, we conduct the numerical performance of our method on two real human faces databases, i.e., the Multiple faces datasets and the AR database. In this part, we compare our method with three efficient benchmark solvers introduced in the literature, including low-tubal-rank tensor linear regression method (TLRFR [7]), low-matrix-rank regularized regression method (LR^3 [19]) and Fast-NMR ([27]). Besides, our method is denoted by **Ours**. Since there are some parameters in model (3.4) and algorithm 1. We set $\tau = 1/2, \alpha = 1$ and $\beta = 1$ for all experiments. All parameters of the other compared algorithms were taken as the default values used in the paper. Moreover, for the fair comparison, we take (3.15) as the stopping criterion with $\epsilon = 10^{-8}$ and set the max iterations as 500 for all methods. To simulate the real situation, each experiment consists of four scenarios:

(i). train samples and test samples are all clean;

(ii). train samples are contaminated and test samples are clean;

(iii). train samples are clean and test samples are contaminated;

(iv). train samples and test samples are all contaminated.

Some examples are collected in Figure 1.

We first consider the Multipie database. The Multipie database has a total of 1515 images, each one size is $240 \times 280 \times 3$. In order to simplify the experiment, it is reduced to $60 \times 70 \times 3$. Here, we selected 792 images as experimental data and divided them into four groups (198 for each group). There are 18 subjects in each group, and each subject has 11 pictures. Here we selected 8 of them as training set and the remaining three as testing set. The face classification rate (**Acc** for short), computing time in seconds (**Time** for short) and number of iterations (**Iter** for short) results are summarized in Table 1. It is not difficult to see that our method takes less computing time and iterations to achieve higher accuracy than the other three methods in many cases.

Secondly, we consider the AR Database, which consists of 4000 images of 126 subjects. Each image is of size $165 \times 120 \times 3$ and there are 13 images for each subject. We randomly chose 50 subjects and divided them into two groups (25 for each group). Here, for each subject, seven images are chosen as training samples and three samples as test set. We list the numerical results in Table 2. Similar to MultiPie Database, our method is competitive to achieve ideal accuracy by taking less computing time and iterations than the other three methods. Therefore, the above experiment on face classification support the reliability of our approach.

MultiPIE faces data sets: http://www.flintbox.com/public/project/4742/
http://www2.ece.ohio-state.edu/ aleix/ARdatabase.html

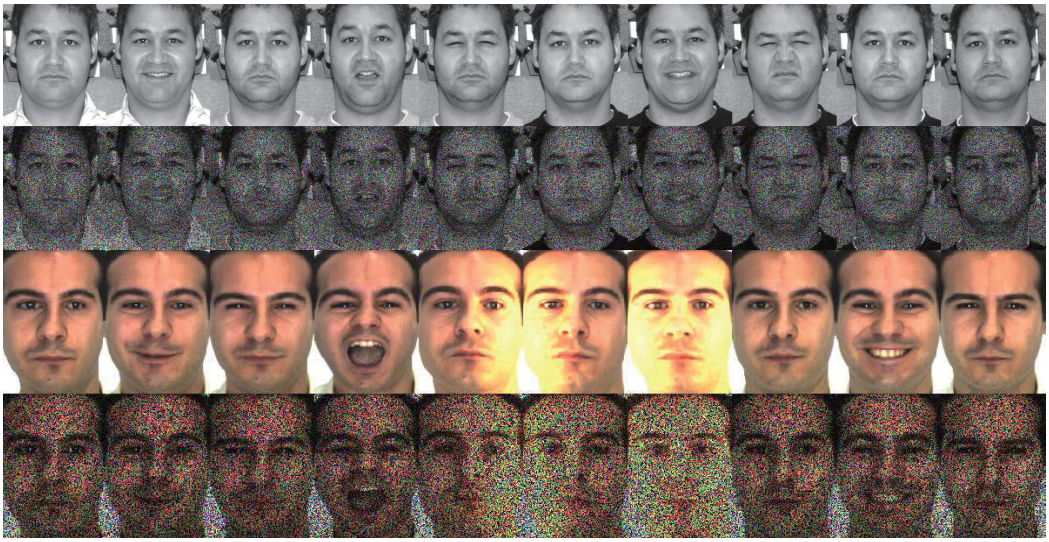

Figure 1: Some images of Multipie dataset and AR dataset under experiments. The first and third rows are the clean data without noise, the second and fourth rows are the corrupted samples images with noise.

Table 1: Computational results on four groups of MultiPie database.

Group ID	Method	scenario (i)			scenario (ii)			scenario (iii)			scenario (iv)		
		Acc	Time	Iter	Acc	Time	Iter	Acc	Time	Iter	Acc	Time	Iter
1st group	TLRFR	0.9259	1.0413	149	0.8333	3.0218	329	0.8703	1.7147	194	0.7777	1.6061	179
	LR^3	0.8518	1.0230	208	0.8333	0.2401	42	**0.8888**	0.2872	38	0.8333	**0.1845**	33
	Fast-NMR	0.9259	2.6249	500	0.8333	2.9635	500	**0.8888**	0.9473	145	0.8333	0.8903	132
	Ours	**0.9629**	**0.1868**	**10**	**0.8518**	**0.2093**	**10**	**0.8888**	**0.1919**	**10**	**0.8703**	0.2497	**10**
2nd group	TLRFR	**0.9259**	2.9667	331	0.8148	4.5087	500	0.8333	1.4016	191	0.6851	1.6823	192
	LR^3	0.8333	1.0514	157	0.8148	**0.1937**	35	**0.8518**	0.2003	39	0.7222	**0.1855**	34
	Fast-NMR	**0.9259**	2.5886	500	0.7777	2.9635	500	**0.8518**	0.5504	132	0.7037	0.6553	123
	Ours	**0.9259**	**0.2698**	**10**	**0.9074**	0.2072	**10**	**0.8518**	**0.1836**	**10**	**0.8148**	0.2135	**10**
3rd group	TLRFR	**1**	1.9891	322	0.8888	4.0145	499	0.9814	1.5208	184	0.8703	1.4222	194
	LR^3	0.7777	1.1646	192	0.9444	**0.1871**	34	**1**	0.2008	39	0.8518	**0.1774**	34
	Fast-NMR	**1**	2.643	500	0.8333	2.4977	419	**1**	0.6205	127	0.8333	0.6076	123
	Ours	**1**	**0.2059**	**10**	**0.9814**	0.2072	**10**	**1**	**0.1836**	**10**	**0.9074**	0.2060	**10**
4th group	TLRFR	**0.9259**	2.5543	341	0.8703	4.2541	500	0.8888	1.6648	191	0.7222	1.8290	196
	LR^3	0.8333	0.885	202	0.8333	**0.2017**	34	0.9074	0.2822	38	0.7592	**0.2108**	34
	Fast-NMR	0.9074	2.6622	500	0.8703	2.6970	407	0.8888	0.7373	131	0.7777	0.6981	124
	Ours	0.9074	**0.1857**	**10**	**0.9074**	0.2173	**10**	**0.9074**	**0.1952**	**10**	**0.8518**	0.2289	**10**

4.2 Traffic data prediction

In this subsection, we apply the proposed method to traffic data prediction. Given a traffic flow data $\mathcal{A}^{\text{true}} \in \mathbb{K}_t^{o_1 \times o_2 \times d}$, where o_1, o_2, d and t represent origin, destination sites, days and time intervals to record the data of each day, respectively. We simply denote $\mathcal{A}_i = \mathcal{A}^{\text{true}}(:, :, i, :)$, $i \in [d]$ in Matlab language. We firstly choose N ($N < d$) days $(\mathcal{A}_1, \ldots, \mathcal{A}_N)$ from $\mathcal{A}^{\text{true}}$ and divide them into two groups, the first q days $(\mathcal{A}_1, \ldots, \mathcal{A}_q \in \mathbb{K}_t^{o_1 \times o_2})$ and the left $N-q$ days $(\mathcal{A}_{q+1}, \ldots, \mathcal{A}_N \in \mathbb{K}_t^{o_1 \times o_2})$. We assume that $\mathcal{A}_{q+1}, \ldots, \mathcal{A}_N$ could be approximated linearly by the last q days with under the transformed T-product, i.e.,

$$\mathcal{A}_{q+i} \approx \boldsymbol{x}_1 \odot_Q \mathcal{A}_i + \cdots + \boldsymbol{x}_q \odot_Q \mathcal{A}_{q+i-1}, \quad 1 \le i \le N - q.$$

Table 2: Computational results on two groups of AR Database

Group ID	Method	scenario (i)			scenario (ii)			scenario (iii)			scenario (iv)		
		Acc	Time	Iter	Acc	Time	Iter	Acc	Time	Iter	Acc	Time	Iter
1st group	TLRFR	**0.9866**	1.5547	498	0.8266	0.4970	157	**0.9866**	1.5789	498	0.8266	0.4777	151
	LR^3	0.7600	**0.3003**	126	0.8400	**0.0923**	**40**	0.7600	**0.2943**	126	0.8266	**0.09025**	**36**
	Fast-NMR	**0.9866**	0.9915	381	0.8533	0.3356	164	**0.9866**	0.9777	381	0.8666	0.3446	181
	Ours	**0.9866**	0.3648	**33**	**0.8666**	0.3641	**40**	**0.9866**	0.3677	**33**	**0.8933**	0.3758	39
2nd group	TLRFR	**0.9733**	1.5869	498	0.7466	0.4787	160	**0.9600**	1.5616	498	0.7066	0.4179	156
	LR^3	0.7600	**0.3154**	122	0.7466	**0.0799**	**37**	0.7600	**0.3055**	122	0.74666	**0.08216**	**37**
	Fast-NMR	0.9600	0.9810	344	0.7866	0.3719	181	**0.9600**	0.9742	344	0.7200	0.3612	181
	Ours	0.9600	0.3598	**33**	**0.8266**	0.3843	45	**0.9600**	0.3563	**33**	**0.8400**	0.3738	43

Then, we establish the following minimization model:

$$\min_{\boldsymbol{X}\in\mathbb{K}_t^q} \quad \|\boldsymbol{X}\|_* + \frac{\alpha}{2}\sum_{i=1}^{N-q}\|\boldsymbol{A}_{q+i} - \widetilde{\mathcal{A}}_{q+i}\circledast_Q \boldsymbol{X}\|_F^2 + \frac{\tau}{2}\|\boldsymbol{X}\|_F^2, \tag{4.2}$$

where $\boldsymbol{A}_{q+i} = \mathrm{Vec}(\mathcal{A}_{q+i}) \in \mathbb{K}_t^{o_1o_2}, \boldsymbol{X} = [\boldsymbol{x}_1; \boldsymbol{x}_2; \ldots; \boldsymbol{x}_q] \in \mathbb{K}_t^q$ and $\widetilde{\mathcal{A}}_{q+i} = [\mathrm{Vec}(\mathcal{A}_i), \ldots, \mathrm{Vec}(\mathcal{A}_{q+i-1})] \in \mathbb{K}_t^{o_1o_2\times q}$. It is not difficult to see that model (4.2) falls into the form of (3.4) by setting

$$\boldsymbol{B} = [\boldsymbol{A}_{q+1}; \boldsymbol{A}_{q+2}; \ldots; \boldsymbol{A}_N] \in \mathbb{K}_t^{(N-q)o_1o_2} \quad \text{and} \quad \mathcal{A} = [\widetilde{\mathcal{A}}_{q+1}; \cdots; \widetilde{\mathcal{A}}_N] \in \mathbb{K}_t^{(N-q)o_1o_2\times q}.$$

Thus, we can obtain the coefficient matrix $\boldsymbol{X}$ by solving optimization problem (3.4) via Algorithm 1. Accordingly, we can approximately predict the next two days' data $\mathcal{A}_{N+1}^{\mathrm{pre}}$ and $\mathcal{A}_{N+2}^{\mathrm{pre}}$ by

$$\mathcal{A}_{N+1}^{\mathrm{pre}} = \boldsymbol{x}_1 \odot_Q \mathcal{A}_{N+1-q} + \cdots + \boldsymbol{x}_q \odot_Q \mathcal{A}_N,$$

and

$$\mathcal{A}_{N+2}^{\mathrm{pre}} = \boldsymbol{x}_1 \odot_Q \mathcal{A}_{N+2-q} + \cdots + \boldsymbol{x}_{q-1} \odot_Q \mathcal{A}_N + \boldsymbol{x}_q \odot_Q \mathcal{A}_{N+1}^{\mathrm{pre}},$$

respectively. Certainly, we could predict the next jth day's data $\mathcal{A}_{N+j}^{\mathrm{pre}}$, $j > 2$ in the similar way. Here, we use the normalized mean absolute error (NMAE) to measure the quality of the predicted data by models and algorithms, where the NMAE is defined as follows

$$\mathrm{NMAE} := \frac{\sum_i \sum_{(j,k,l)} |(\mathcal{A}_i - \mathcal{A}_i^{\mathrm{pre}})_{jkl}|}{\sum_i \sum_{(j,k,l)} |(\mathcal{A}_i)_{jkl}|}.$$

Considering that both LR^3 ([19]) and Fast-NMR ([27]) are matrix-based methods, in our experiments, we only compare our approach with the aforementioned low-tubal-rank tensor linear regression model (i.e., TLRFR in [7]) to highlight the superiority of our approach over the tensor-based method. Moreover, we consider three synthetic datasets and three real-world traffic datasets.

Now, we first consider some structured synthetic datasets to investigate the feasibility and reliability of our approach. Here, we randomly generate three datasets, each one is of size $50 \times 50 \times 100$. Specifically, for $i, j \in [50]$ and $\boldsymbol{v} \in [100]$, the first dataset is generated by $\mathcal{A}^{\mathrm{true}}(i, j, :) = \mathrm{sqrt}(\boldsymbol{v}) + \mathrm{rand}$ in Matlab script; the second one is generated by $\mathcal{A}^{\mathrm{true}}(i, j, :) = \log(\boldsymbol{v}) + \mathrm{rand} + \mathrm{rand} * \boldsymbol{v}^2$, $\boldsymbol{v} \in [100]/100$; the third one is generated by $\mathcal{A}^{\mathrm{true}}(i, j, :) = \cos(\boldsymbol{v} * 100) + \mathrm{rand}/5 + \mathrm{rand} * \boldsymbol{v}/10$. In the experiments, we simply set $q = 2$ and $N = 3$. Here, we only show some predicted results in Figure 2. It can be seen that the visual results obtained by our approach is reliable, at least in the sense of data trend, to approximate the true data.

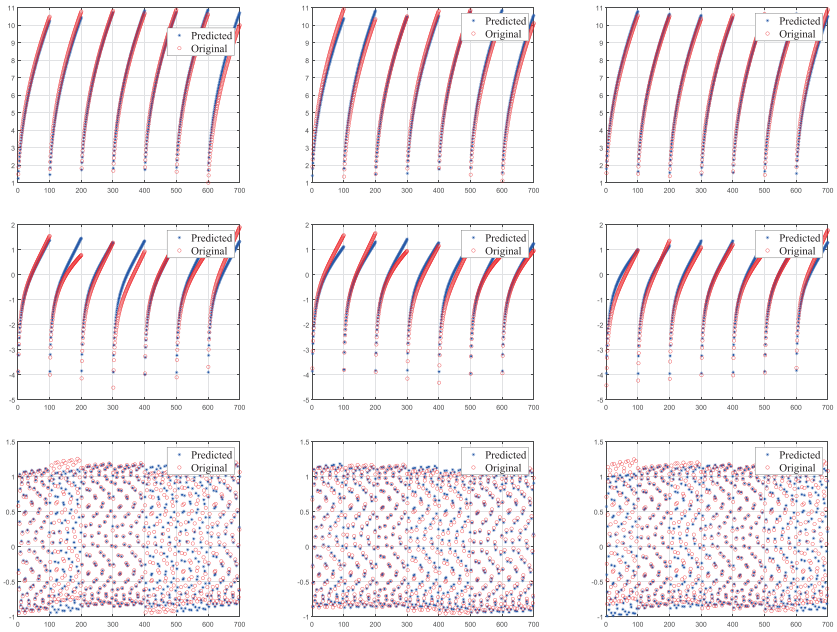

Figure 2: The visualization of the predicted results on synthetic datasets by our approach.

Hereafter, we are concerned with the numerical performance of our approach on real-world datasets. In our experiments, we consider three widely used traffic datasets including GÊANT dataset [25], Hangzhou Metro Passengers Flow, and Guangzhou urban traffic speed dataset.

- In GÊANT dataset, there are 23 routers and 529 origin and destination (OD) pairs. For each OD pair, a count of network traffic flow is recorded for every 15 minutes in a day. We also organize the data as a tensor of $23 \times 23 \times 96 \times 119$.
- Hangzhou dataset collected incoming passenger flow from 80 metro stations over 25 days (from January 1 to January 25, 2019) with a 10-minute resolution in Hangzhou, China. We discard the interval 0:00 a.m.–6:00 a.m. with no services (i.e., only consider the remaining 108 time intervals) and re-organize the raw data set into a tensor of $80 \times 25 \times 108$.
- Guangzhou dataset is collected from 214 road segments in Guangzhou, China within two months (i.e., 61 days from August 1, 2016 to September 30, 2016) at 10-min interval (144 time intervals per day). The speed data can be organized as a third-order tensor (road segment×day×time interval, with a size of $214 \times 61 \times 144$). There are about 1.29% missing values in the raw data set.

In the experiments, we performance two cases: $q = 2, N = 3$ and $q = 6, N = 7$, and we predict only one day and seven consecutive days for two cases. The numerical results are

https://tianchi.aliyun.com/competition/entrance/231 708/information
https://doi.org/10.5281/zenodo.1205229
For a third-order tensor $\mathcal{W} \in \mathbb{K}_t^{m \times d}$, it could be regarded as of size $m \times 1 \times d \times t$.

Table 3: Computational results for one day when $p = 2$ and $p = 6$

p	Method	Hangzhou			Guangzhou			GÊANT		
		NMAE	Time	Iter	NMAE	Time	Iter	NMAE	Time	Iter
2	TLRFR	0.6642	1.3812	500	0.1794	2.6184	500	0.7653	4.1803	500
	Ours	0.1933	0.1868	66	0.1393	1.4812	248	0.1702	0.2121	38
6	TLRFR	0.5299	1.8335	500	0.1616	2.918	500	0.7725	5.6809	500
	Ours	0.1639	0.2441	59	0.1045	0.8931	153	0.3109	0.1616	32

Table 4: Computational results for seven days when $p = 2$ and $p = 6$

p	Method	Hangzhou			Guangzhou			GÊANT		
		NMAE	Time	Iter	NMAE	Time	Iter	NMAE	Time	Iter
2	TLRFR	0.7427	1.3451	500	0.9229	2.0058	500	0.7973	3.5811	500
	Ours	0.4797	0.2328	64	0.1467	0.7245	327	0.3297	0.1699	37
6	TLRFR	0.6372	1.1352	500	0.9019	2.3401	500	0.7132	4.6821	500
	Ours	0.3716	0.2103	54	0.1203	0.6155	298	0.4283	0.1329	29

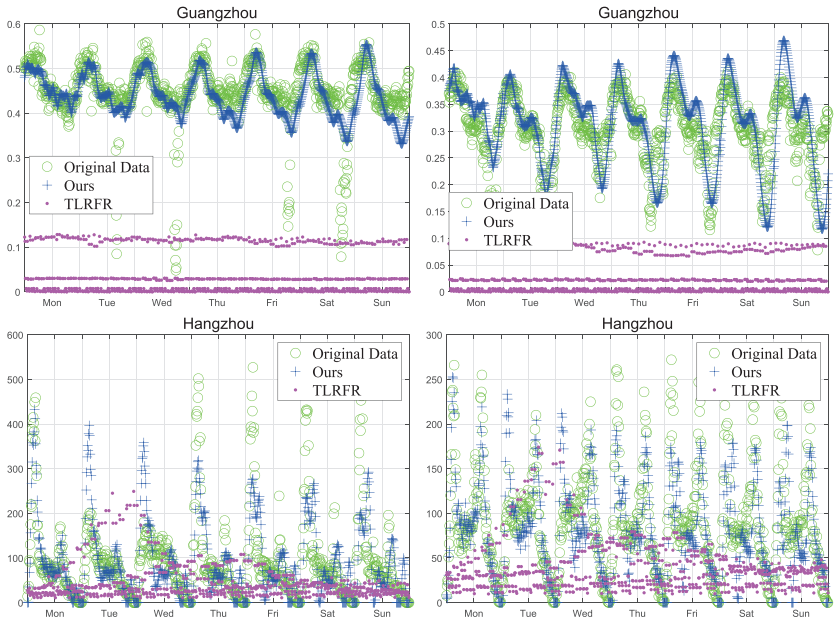

Figure 3: Prediction results of Guangzhou and Hangzhou datasets for seven consecutive days by our approach and TLRFR when $q = 2$.

reported in Tables 3 and 4. It is promising that our method works better than the TLRFR approach in terms of NMAE, computing time and iterations. Moreover, we visually show some predicted results (seven consecutive days in some sites) on Guangzhou and Hangzhou datasets in Figures 3 and 4. We can easily see that our approach can better approximate the true data than the TLRFR approach. In particular, TLRFR does not work in this application as it represents the testing data with a vector, which further verifies the power of our method.

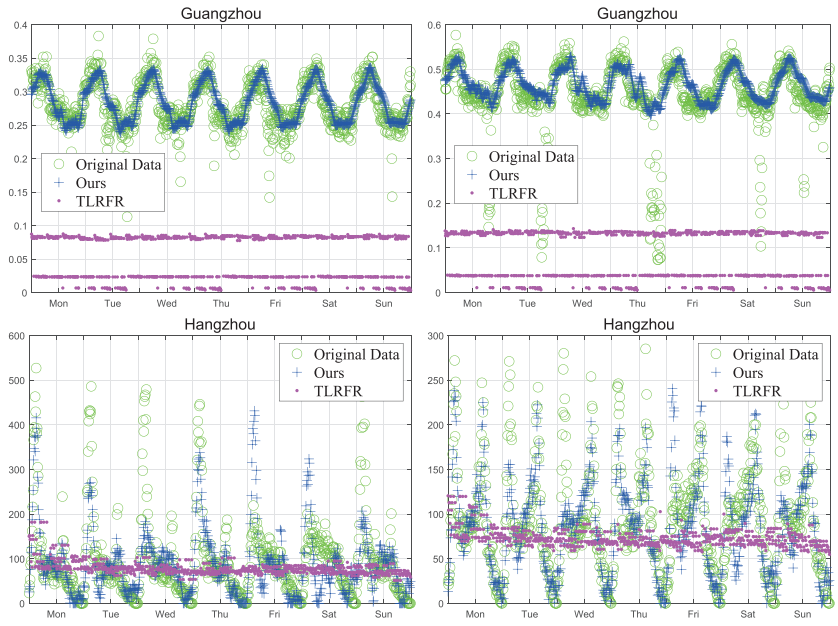

Figure 4: Prediction results of Guangzhou and Hangzhou datasets for seven consecutive days by our approach and TLRFR when $q = 6$.

5 Conclusion

In this paper, we introduced a low-rank prior tensor linear regression approach for high-dimentional data analysis, where the given test tensor data is represented as a linear combination of all training tensor samples under the transformed T-product. Due to the non-smooth nuclear norm, we accordingly introduced an auxiliary variable to make the related minimization model separable so that our proposed ADMM enjoys an easily implementable iterative scheme. A series of computational experiments on color face classification and traffic data demonstrate that out approach performs better than some existing methods.

Acknowledgments

The authors would like to thank two anonymous referees for their close reading and valuable comments, which helped us improve the presentation of this paper.

References

[1] E. Candès and T. Tao, The power of convex relaxation: near-optimal matrix completion, *IEEE Trans. Inf. Theory* 56 (2010) 2053–2080.

[2] E. J. Candès and B. Recht, Exact matrix completion via convex optimization, *Found. Comut. Math.* 9 (2009) 717–772.

[3] J. Chien and C. Wu, Discriminant waveletfaces and nearest feature classifiers for face recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (2002) 1644–1649.

[4] L. Fahrmeir, T. Kneib, S. Lang and B. Marx, *Regression: Models, Methods and Applications*, Springer, 2021.

[5] M. Fazel, *Matrix Rank Minimization with Applications*, Ph.D. thesis, Stanford University, 2002.

[6] M. Fazel, H. Hindi and S. Boyd, A rank minimization heuristic with application to minimum order system approximation, in: *Proceedings of the American Control Conference*, 2001 pp. 4734–4739.

[7] Q. Gao, J. Cheng, D. Xie, P. Zhang, W. Xia and Q. Wang, Tensor linear regression and its application to color face recognition, in: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019 pp. 523–531.

[8] H.J. He, C. Ling and W.H. Xie, Tensor completion via a generalized transformed tensor t-product decomposition without t-SVD, *J. Sci. Comput.* 93 (2022): Article No. 47.

[9] Z. Hellwig, *Linear Regression and Its Application to Economics*, Elsevier, 1963.

[10] A. Hoerl and R. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* 42 (2000) 80–86.

[11] K. Kilmer and M. Aeron, Tensor-tensor products with invertible linear transforms, *Linear Algebra Appl.* 485 (2015) 545–570.

[12] M. Kilmer and C. Martin, Factorization strategies for third-order tensors, *Linear Algebra Appl.* 435 (2011) 641–658.

[13] J. Kim, A. Mowat, P. Poole and N. Kasabov, Linear and non-linear pattern recognition models for classification of fruit from visible-near infrared spectra, *Chemometr. Intell. Lab. Syst.* 51 (2000) 201–216.

[14] A. Lewis and H. Sendov, Nonsmooth analysis of singular values, part I: Theory, *Set-Valued Anal.* 13 (2005), 213–241.

[15] S. Li and J. Lu, Face recognition using the nearest feature line method, *IEEE Trans. Neural Netw. Learn. Syst.* 10 (1999) 439–443.

[16] J. Liu, P. Musialski, P. Wonka and J. Wonka, Tensor completion for estimating missing values in visual data, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (2013) 208–220.

[17] I. Naseem, R. Togneri and M. Bennamoun, Linear regression for face recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (2010) 2106–2112.

[18] H. Pan, J. Liu, S. Zhou and Z. Niu, A block regression model for short-term mobile traffic forecasting, in: *IEEE/CIC ICCC 2015 Symposium on Next Generation Networking*, 2015 .

[19] J. Qian, J. Yang and F. Zhang, Robust low-rank regularized regression for face recognition with occlusion, in: *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014 pp. 21–26.

[20] D. Qiu, M.R. Bai, M.K. Ng and X.J. Zhang, Robust low-rank tensor completion via transformed tensor nuclear norm with total variation regularization, *Neurocomputing* 435 (2021), 197–215.

[21] B. Recht, M. Fazel and P. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, *SIAM Rev.* 52 (2010) 471–501.

[22] Z. Shan, D. Zhao and Y. Xia, Urban road traffic speed estimation for missing probe vehicle data based on multiple linear regression model, in: *Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013 pp. 6–9.

[23] A. Strehl and M. Littman, Online linear regression and its application to model-based reinforcement learning, in: *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, 2007 .

[24] M. Udell and A. Townsend, Why are big data matrices approximately low rank ? *SIAM J. Math. Data Sci.* **1** (2019) 144–160 .

[25] S. Uhlig, B. Quoitin, J. Lepropre and S. Balon, Providing public intradomain traffic matrices to the research community, *ACM Sigcomm. Comp. Com.* 36 (2006) 83–86.

[26] J. Wright, A. Yang, A. Ganesh, S. Sastry and Y. Ma, Robust face recognition via sparse representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 31 (2009) 210–227.

[27] J. Yang, L. Luo, J. Qian, Y. Tai, F. Zhang and Y. Xu, Nuclear norm based matrix regression with applications to face recognition with occlusion and ollumination changes, *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (2017) 156–171.

[28] W. Zha, R. Chellapp, P. Phillips and A. Rosenfeld, Face recognition: A literature survey, *ACM Comput. Surv.* 35 (2003) 399–458.

[29] L. Zhang, M. Yang and X. Feng, Sparse representation or collaborative representation: Which helps face recognition?, in: *2011 International Conference on Computer Vision*, 2011 pp. 471–478.

[30] Z. Zhang, G. Ely, S. Aeron, N. Hao and M. Kilmer, Novel methods for multilinear data completion and de-noising based on tensor-svd, in: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014 pp. 3842–3849.

Manuscript received 22 July 2023
revised 21 October 2023
accepted for publication 15 November 2023

Chenjian Pan
School of Mathematics and Statistics, Ningbo University
Ningbo, 315211, China
E-mail address: panchenj@163.com

Hongjin He
School of Mathematics and Statistics, Ningbo University
Ningbo, 315211, China
E-mail address: hehongjin@nbu.edu.cn

Chen Ling
Department of Mathematics, School of Science, Hangzhou Dianzi University
Hangzhou, 310018, China
E-mail address: macling@hdu.edu.cn