ISSN: 2664-9640

Application of Iterative Dynamic Tensor Estimation Algorithm based on Norm Minimization in Video Images

Junchen Li*

School of Mathematics and Statistics, Chongqing Technology and Business University, Chongqing 400067, China

*ljcand1412@163.com

Abstract

With the advent of the big data era, traditional data analysis tools are facing challenges in handling high-dimensional, nonlinear, and heterogeneous data. Tensor factorization models have become a hot topic of research due to their ability to deeply mine the higher-order structure of data, especially showing broad application prospects in fields such as finance, biomedicine, and social sciences. This article focuses on the robust estimation techniques of the Tucker tensor factorization model. The innovation of this method lies in the establishment of a minimization objective function by introducing an exponential squared loss function, followed by the robust estimation of model parameters through iterative optimization. The main purpose is to enhance the accuracy of estimation and the adaptability of the model to outliers. Subsequent validation was conducted through large-scale simulation experiments. Comparison results confirm that the robust estimation method proposed in this paper demonstrates significant advantages in terms of estimation accuracy and robustness, especially in dealing with data with heavier tail distributions, where its performance is particularly outstanding. Finally, we apply the method to a set of mobile video data and show that the robust approach can lead to better analysis results.

Keywords

Tensor Time Series; Exponential Square Loss; Minimum Optimization; Moving MNIST.

1. Introduction

Tensor factor models are advanced data analysis tools used for processing and analyzing multidimensional data. In reference [1], the authors were the pioneers in proposing a modeling method for tensor time series factor models, introducing two types of factor load estimation methods: TOUUP and TIPUP. [2] combined tensor decomposition with Bayesian estimation methods to analyze tensor time series data, which can predict and analyze in cases of many missing values in the tensor time series. [3] proposed a computationally efficient PCA and orthogonalization algorithm for tensor CP decomposition. [4] proposed a tensor time series factor model method, mainly using tensor CP decomposition for analyzing high-dimensional dynamic tensor time series. Since the vector column space after CP decomposition has uniqueness, it has distinctly different estimation properties from the Tucker tensor decomposition factor model. [5] proposed improved versions of the iTOPIP and iTIPUP iterative algorithms based on the combination of TOUPU and TIPUP, with greater estimation accuracy but at a higher computational cost. [6] introduced a pre-averaging method, it was compared with TOPUP and TIPUP, showing that this method has better estimation accuracy under Gaussian tensor data. [7] proposed a factor estimation method for heavy-tailed matrix time series data. However, there is a lack of research on tensor time series data of third order and above. [8] proposed the moPCA estimation method for tensor factor matrixization and derived the iterative version of the IPmoPCA method. Compared to the iTOPIP and iTIPUP iterative algorithms by [5], this method has significantly lower computational costs and notably superior estimation accuracy than the methods mentioned above. However, the IPmoPCA method is effective only for nonheavy-tailed data and essentially fails for tensor heavy-tailed data. This letter primarily introduces an exponential squared loss function into the Tucker tensor factorization model, establishing a robust factor estimation method. A new weighted iterative projection estimation method is proposed, and through random simulations, it has been confirmed that the weighted iterative projection estimation method under exponential squared loss outperforms other methods. Finally, the effectiveness of the method is validated through analysis of actual data.

2. Robust Iterative Algorithm for Tensor Factor Models

2.1. Symbol Description

For a matrix $\boldsymbol{A}, \boldsymbol{A}^\top$ is the transpose of $\boldsymbol{A}$, $\mathbf{Tr}(\boldsymbol{A})$ is the trace of $\boldsymbol{A}$. For two matrices $\boldsymbol{A} \in \mathbb{R}^{I\times J}$ and $\boldsymbol{B} \in \mathbb{R}^{K\times L}$, $\boldsymbol{A} \otimes \boldsymbol{B} = (\boldsymbol{a}_{ij}\boldsymbol{B})_{I\times J} \in \mathbb{R}^{IK\times JL}$ denotes the Kronecker product(a special form of tensor outer product) of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Let $\boldsymbol{p}_{-d} = \prod_{i\neq d}^{D} \boldsymbol{p}_i = \boldsymbol{p}_1\boldsymbol{p}_2 \cdots \boldsymbol{p}_{d-1}\boldsymbol{p}_{d+1} \cdots \boldsymbol{p}_D$ and $\boldsymbol{k}_{-d} = \prod_{i\neq d}^{D} \boldsymbol{k}_i = \boldsymbol{k}_1\boldsymbol{k}_2 \cdots \boldsymbol{k}_{d-1}\boldsymbol{k}_{d+1} \cdots \boldsymbol{k}_D$ with $\boldsymbol{d}$ being the placeholder and $\boldsymbol{A}_{[D]/\{d\}} = \boldsymbol{A}_D \otimes \cdots \otimes \boldsymbol{A}_{d+1} \otimes \boldsymbol{A}_{d-1} \otimes \cdots \otimes \boldsymbol{A}_1 \in \mathbb{R}^{p_{-d}}$. For a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{p_1\times p_2\times\cdots\times p_D}$, the mode- $\boldsymbol{d}$ unfolding matrix $\boldsymbol{\mathcal{X}}^{(d)} \in \mathbb{R}^{p_d\times p_{-d}}$ is a $\boldsymbol{p}_d \times \boldsymbol{p}_{-d}$ matrix by assembling all $\boldsymbol{p}_{-d}$ mode- $\boldsymbol{d}$ fibers as columns of the matrix, which maps a tensor $\boldsymbol{\mathcal{X}}$ to a matrix $\boldsymbol{\mathcal{X}}^{(d)} \in \mathbb{R}^{p_d\times p_{-d}}$. The $\boldsymbol{d}$-mode product of tensor $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{r_1\times\cdots\times r_D}$ and matrix $\boldsymbol{U} \in \mathbb{R}^{p_d\times r_d}$ is defined as $\boldsymbol{\mathcal{S}} \times_d \boldsymbol{U} \in \mathbb{R}^{r_1\times\cdots\times r_{d-1}\times p_d\times r_{d+1}\times\cdots\times r_D}$ with the $(\boldsymbol{i}_1, \dots, \boldsymbol{i}_D)$ -th element $(\boldsymbol{\mathcal{S}} \times_d \boldsymbol{U})_{i_1,\dots,i_D} = \sum_{j_d=1}^{r_d} \mathbf{S}_{i_1,\dots,j_d,\dots,i_D} \boldsymbol{U}_{i_d,j_d}$ and $\boldsymbol{\mathcal{Y}}^{(k)} = \boldsymbol{U}\boldsymbol{\mathcal{S}}^{(k)}$. The Frobenious norm of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}_1^{p_1\times p_2\times\cdots\times p_D}$ is defined as $\| \boldsymbol{\mathcal{X}} \|_F = \left(\sum_{i_1=1}^{p_1} \cdots \sum_{i_D=1}^{p_D} \boldsymbol{\mathcal{X}}_{i_1\cdots i_D}^2\right)^{1/2}$.

2.2. Exponentially Squared Weighted Iterative Estimation

For the observed tensor time series $\mathcal{X}_1, \dots, \mathcal{X}_T \in \mathbb{R}^{p_1\times p_2\times\cdots\times p_D}$, consider the following TFM with a Tucker low rank structure:

$$\mathcal{X}_t = \mathcal{S}_t + \mathcal{E}_t,\ \mathcal{S}_t = \mathcal{F}_t \times_1 A_1 \times_2 \cdots \times_D A_D, \tag{1}$$

where $\mathcal{S}_t$ is an unknown low-rank signal part, $\mathcal{F}_t \in \mathbb{R}^{k_1\times k_2\times\cdots\times k_D}$ is a low-dimensional ($k_d \ll p_d$) latent tensor factor, $A_d \in \mathbb{R}^{p_d\times k_d}$ is an unknown mode- d loading matrix for $d \in [D]$,$\times_d$ is the mode- d product, and $\mathcal{E}_t \in \mathbb{R}^{p_1\times p_2\times\cdots\times p_D}$ is a noise tensor. Model (2) can be rewritten as the Tucker-TFM:

$$\mathcal{X}_t = \mathcal{F}_t \times_1 A_1 \times_2 \dots \times_D A_D + \mathcal{E}_t. \tag{2}$$

In order to ensure model (2)to be identifiable, we introduce the identifiability condition:

$$\frac{1}{p_d} A_d^\top A_d = I_{k_d} \text{ for } d = 1,2,\cdots,D. \tag{3}$$

The estimation of the load matrix can be solved by establishing a minimization objective function through the following exponential square function:

The estimation of the load matrix can be solved by establishing a minimization objective function through the following exponential square function:

$$\min_{\{\mathbf{A},\mathcal{F}_t\}} \phi_h\left(\|x_t - \mathcal{F}_t \times_{d=1}^{D} A_d\|_F\right) = \min_{\{\mathbb{A},\mathcal{F}_t\}} \frac{1}{Tp} \sum_{t=1}^{T} \left[1 - \exp\left(-\frac{\|x_t - \mathcal{F}_t \times_{d=1}^{D} A_d\|_F^2}{h}\right)\right] \\ \text{s.t. } \forall d \in \{1, \cdots, D\}, \frac{1}{p_d} A_d^\top A_d = I_{k_d}, \frac{1}{p_{-d}} A_{[D]/\{d\}}^\top A_{[D]/\{d\}} = I_{k_{-d}}, \quad (4)$$

where h is an adjustment parameter. The tuning parameter h controls the sensitivity of the estimator to outliers. By appropriately choosing h, a good balance between bias and variance can be found to improve the robustness of the estimate.

For the term $\|\mathcal{X}_t - \mathcal{F}_t \times_{d=1}^{D} A_d\|_F^2$, upon simplification, we obtain:

$$\|\mathcal{X}_t - \mathcal{F}_t \times_{d=1}^{D} A_d\|_F^2 = \left\|\mathcal{X}_t^{(d)} - (\mathcal{F}_t \times_{d=1}^{D} A_d)^{(d)}\right\|_F^2 \\ = \mathrm{Tr}\left(\mathcal{X}_t^{(d)\top} \mathcal{X}_t^{(d)}\right) - 2\mathrm{Tr}\left(\mathcal{X}_t^{(d)\top} A_d \mathcal{F}_t^{(d)} A_{[D]/\{d\}}^\top\right) + p\mathrm{Tr}\left(\mathcal{F}_t^{(d)\top} \mathcal{F}_t^{(d)}\right) \quad (5)$$

To solve the optimization problem of the objective function (4), we assume that given A_d and $A_{([D]/\{d\})}$, the value of tensor $\mathcal{F}_t$ does not affect the optimal solution of the objective function $\mathcal{L}_{\text{exponential}}$. Therefore, for any $1 \leq t \leq T$, the firstorder condition is:

$$\frac{\partial}{\partial \mathcal{F}_t^{(d)}} \left(\mathrm{Tr}\left(\mathcal{X}_t^{(d)\top} \mathcal{X}_t^{(d)}\right) - 2\mathrm{Tr}\left(\mathcal{X}_t^{(d)\top} A_d \mathcal{F}_t^{(d)} A_{[D]/\{d\}}^\top\right) + p\mathrm{Tr}\left(\mathcal{F}_t^{(d)\top} \mathcal{F}_t^{(d)}\right)\right) = 0 \quad (6)$$

We can obtain:

$$\mathcal{F}_t^{(d)} = \frac{1}{p} A_d^\top \mathcal{X}_t^{(d)} A_{[D]/[d]} \quad (7)$$

Using (5) to replace $\|\mathcal{X}_t - \mathcal{F}_t \times {}_{d=1}^{D} A_d\|_F^2$ in (4), and substituting $\mathcal{F}_t^{(d)}$ in (4) with (7), under the constraints $\frac{1}{p_d} A_d^\top A_d = I_{k_d}, \frac{1}{p-d} A_{[D]/[d]}^\top A_{[D]/[d]} = I_{k_{-d}}, d = 1, \cdots, D$, the optimization problem in (4) can be solved by minimizing the following Lagrangian function:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} \left[1 - \exp\left(-\frac{1}{h} \mathrm{Tr}\left(x_t^{(d)\top} x_t^{(d)}\right) - \frac{1}{p} \mathrm{Tr}\left(x_t^{(d)\top} A_d A_d^\top x_t^{(d)} A_{[D]/\{d\}} A_{[D]/\{d\}}^\top\right)\right)\right], \quad (8)$$

where Θ_d is parameter matrices. By derivation of the Lagrangian function constructed in (8), we can get:

$$\frac{\partial \mathcal{L}}{\partial A_d} = -\frac{2}{Tpp_{-d}} \sum_{t=1}^{T} w_{d,t} \mathcal{X}_t^{(d)} A_{[D]/\{d\}} A_{[D]/\{d\}}^\top \mathcal{X}_t^{(d)\top} A_d + \frac{2}{p_d} A_d \Theta_d = 0 \quad (9)$$

where (9) can be equivalently expressed as:

$$\frac{1}{Tpp_{-d}}\sum_{t=1}^{T} w_{d,t}\mathcal{X}_t^{(d)}A_{[D]/\{d\}}A_{[D]/\{d\}}^{\top}\mathcal{X}_t^{(d)\top}A_d = A_d\Theta_d, \tag{10}$$

where $w_{d,t}$ is the weighted weight under the exponential square loss function, which can be expressed as:

$$w_{d,t} = \frac{1}{h}\exp\left(-\frac{\mathrm{Tr}\left(x_t^{(d)\top}x_t^{(d)}\right)}{h}+\frac{1}{ph}\mathrm{Tr}\left(x_t^{(d)\top}A_dA_d^{\top}x_t^{(d)}A_{[D]/\{d\}}A_{[D]/\{d\}}^{\top}\right)\right) \tag{11}$$

Transform the left side of equation (9) and express it by M_d^w and $\mathrm{M}_{[D]/\{d\}}^w$ respectively:

$$\mathrm{M}_d^w = \frac{1}{Tpp_{-d}}\sum_{t=1}^{\mathrm{T}} w_{d,t}\mathcal{X}_t^{(d)}A_{[D]/\{d\}}A_{[D]/\{d\}}^{\top}\mathcal{X}_t^{(d)\top} \tag{12}$$

Substituting (12) into (10) results in:

$$\mathrm{M_d^w}A_d = A_d\Theta_d \tag{13}$$

Arrange the first k_d largest eigenvalues of the matrix M_d^w in descending order as $\lambda_{d,1},\dots,\lambda_{d,k_d}$, and their corresponding eigenvectors are $U_d = \left(u_{d,1}\cdots u_{d,k_d}\right)$. Then,

$$A_d^* := \sqrt{p_d}U_d \tag{14}$$

can be used as the estimate for A_d, and:

$$\Theta_d^* := \mathrm{diag}\left(\lambda_{d,1},\dots,\lambda_{d,k_d}\right) \tag{15}$$

be used as the estimate for Θ_d.

In the process of solving for A_d through (12), $A_{[D]\backslash\{d\}}$ is required as a condition for constituting M_d, thus introducing the concept of iterative estimation. The exponential square weighted iterative algorithm is shown in the following Algorithm :

Input: tensor data $\{\mathcal{X}_t\}_{t=1}^T$, factor numbers $\{k_d\}_{d=1}^D$. initial estimation of loading matrices $\left\{\hat{A}_d^{(0)}\right\}_{d=1}^D$.

Output: loading matrixs $\left\{\hat{A}_d^{(w,s+1)}\right\}_{d=1}^D$ and signal tensor $\hat{\mathcal{S}}_t^{(w,s+1)}$.

Step 1: compute

$$\hat{A}_{[D]/\{d\}}^{(w,s+1)} = \hat{A}_D^{(w,s)}\otimes\cdots\otimes\hat{A}_{d+1}^{(w,s)}\otimes\hat{A}_{d-1}^{(w,s+1)}\otimes\cdots\otimes\hat{A}_1^{(w,s+1)}$$

Step 2: compute

$$\widehat{w}_{d,t}^{(s+1)} = \frac{1}{h}\exp\left(-\frac{1}{h}\left[\mathrm{Tr}\left(\mathcal{X}_t^{(d)\top}\mathcal{X}_t^{(d)}\right)-\frac{1}{p}\mathrm{Tr}\left(\mathcal{X}_t^{(d)\top}\hat{A}_d^{(w,s)}\hat{A}_d^{(w,s)\top}\mathcal{X}_t^{(d)}\hat{A}_{[D]/\{d\}}^{(w,s+1)}\hat{A}_{[D]/\{d\}}^{(w,s+1)\top}\right)\right]\right)$$

Step 3: compute

$$\widehat{\mathrm{M}}_d^{(w,s+1)} = \frac{1}{Tpp_{-d}} \sum_{t=1}^{T} \widehat{w}_{d,t}^{(s+1)} \mathcal{X}_t^{(d)} \hat{A}_{[D]/\{d\}}^{(w,s+1)} \hat{A}_{[D]/\{d\}}^{(w,s+1)\top} \mathcal{X}_t^{(d)\top}$$

Step 4: compute $\hat{A}_d^{(w,s+1)}$ as $\sqrt{p_d}$ times the matrix with columns being the first k_d eigenvectors of $\widehat{\mathrm{M}}_d^{(w,s+1)}$.

Step 5: repeat steps 1 to 4 until convergence, or up to the maximum number of iterations, as well as the core tensor and signal tensor at step $s+1$.

$$\hat{\mathcal{F}}_t^{(w,s+1)} = \frac{1}{p} \mathcal{X}_t \times_1 \hat{A}_1^{(w,s+1)\top} \times_2 \cdots \times_D \hat{A}_D^{(w,s+1)\top},$$
$$\hat{\mathcal{S}}_t^{(w,s+1)} = \hat{\mathcal{F}}_t^{(w,s+1)} \times_1 \hat{A}_1^{(w,s+1)} \times_2 \cdots \times_D \hat{A}_D^{(w,s+1)}$$

2.3. Tuning Parameter Selection

In the previous section, when introducing the exponential squared loss to establish the minimization objective function, attention was paid to the parameter value h of the function itself.

$$\begin{aligned} \mathcal{L}_{\text{exponential}} &= \min_{\{A,\mathcal{F}_t\}} L_{\text{exponential}} \left(\| \mathcal{X}_t - \mathcal{F}_t \times_{d=1}^{D} A_d \|_F \right) \\ &= \min_{\{A,\mathcal{F}_t\}} \frac{1}{Tp} \sum_{t=1}^{T} \left[1 - \exp\left(- \frac{\| \mathcal{X}_t - \mathcal{F}_t \times_{d=1}^{D} A_D \|_F^2}{h} \right) \right] \end{aligned} \tag{16}$$

In practice, when solving the minimization problem in (5), it should satisfy $h \gg \| \mathcal{X}_t - \mathcal{F}_t \times {}_{d=1}^{D} A_d \|_F$. Therefore, when about to perform estimation, the selection is $h = \arg\max_{1 \le t \le T} T \left\{ \prod_{i=1}^{D} \| \mathcal{X}_t - \hat{\mathcal{F}}_t^{(0)} \times {}_{d=1}^{D} \hat{A}_d^{(0)} \|_F^2 \right\}$.

3. Simulation Study

3.1. Data Generation

In this section we exhibit the finite sample performance of the proposed robust procedure by Monte Carlo Simulation. Let $\mathcal{X}_t$ be the observed tensor at time $t, t = 1,2,\ldots,T$. Suppose that tensor observations are generated by an order-3 tensor factor model:

$$\mathcal{X}_t = \mathcal{F}_t \times_1 A_1 \times_2 A_2 \times_3 A_3 + \mathcal{E}_t \tag{17}$$

where $\mathcal{X}_t \in R^{p_1 \times p_2 \times p_3}, \mathcal{F}_t \in R^{k_1 \times k_2 \times k_3}, A_i \in R^{p_i \times k_i}$ with the entries of A_1, A_2 and A_3 generated independently from the uniform distribution $\mathcal{U}(-1,1)$. Here assume that $k_1 = k_2 = k_3 = 2$. The core tensor factor $\mathcal{F}_t \in R^{2\times 2\times 2}$ is generated by the model:

$$\mathcal{F}_t = \phi \mathcal{F}_{t-1} + \sqrt{1-\phi^2} \mathcal{V}_t,\ t \in [T] \tag{18}$$

ISSN: 2664-9640

where $\mathrm{vec}(\mathcal{V}_t) \sim N_k(\mathbf{0}, \mathbf{I}_k), k = k_1k_2k_3 = 8$, and ϕ controls temporal correlation of tensor factor. The tensor noise $\mathcal{E}_t \in R^{p_1\times p_2\times p_3}$ is generated by the model:

$$\mathcal{E}_t = \psi\mathcal{E}_{t-1} + \sqrt{1-\psi^2}\mathcal{U}_t,\ t \in [T] \tag{19}$$

where $\mathcal{U}_t$ is a tensor normal or t distribution, and ψ controls temporal correlation of $\mathcal{E}_t$. If $\mathcal{U}_t \sim \mathcal{TN}(\mathcal{M}, \Sigma_1, \Sigma_2, \Sigma_3)$, then $\mathrm{Vec}(\mathcal{U}_t) \sim \mathcal{N}(\mathrm{Vec}(\mathcal{M}), \Sigma_3 \otimes \Sigma_2 \otimes \Sigma_1)$. If $\mathcal{U}_t$ is a tensor t distribution $t_v(\mathcal{M}, \Sigma_1, \Sigma_2, \Sigma_3)$, then $\mathrm{Vec}(\mathcal{U}_t)$ is a multivariate t distribution $t_v(\mathcal{M}, \Sigma_3 \otimes \Sigma_2 \otimes \Sigma_1)$. Here set v(degrees of freedom)=2 and $\mathcal{M} = 0$. Let $\mathbf{\Sigma}_h$ be the matrix with 1 on the diagonal and $1/p_l$ on the off-diagonal for $l = 1,2,3$.

For the selection of dimensions for $\mathcal{X}_t$, the dimensions (p_1, p_2, p_3) of $\mathcal{X}_t$ are set to size 1: (10,10,10), size2: (20,20,50), and size 3: (20,50,100). The time series T is set to $T = 20, T = 100$. For ϕ and ψ, we consider the $\phi = 0.8$ and $\psi = 0.1$.

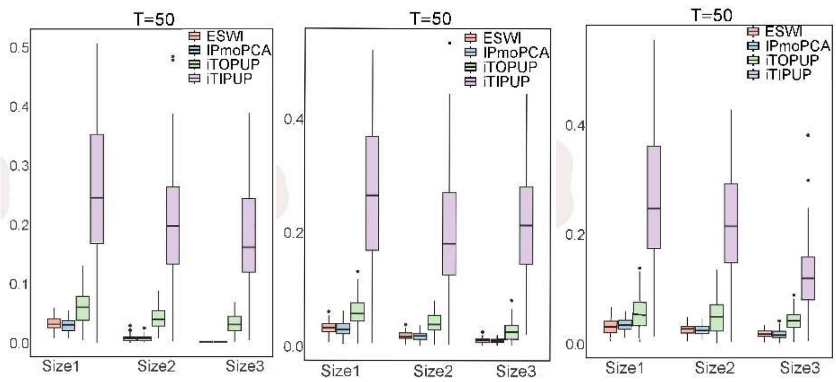
Next, it will compare the Exponential Loss Weighted Iterative Method (ESWI) proposed in this paper with other methods: IPmoPCA, iTOPUP, and iTIPUP.

3.2. Factor Loading Space Estimation

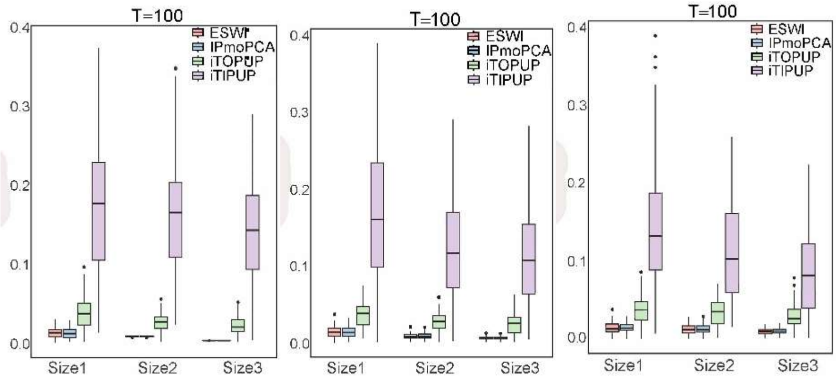
During the process of estimating the loading matrix, we utilize the comparison method proposed by [5], for the estimated matrix $\hat{A}_d$ and the actual matrix A_d. The distance between their column spaces is defined as:

$$\mathcal{D}(\hat{A}_d, A_d) = \left(1 - \frac{1}{k_d}\mathrm{Tr}(\hat{Q}_d\hat{Q}_d^\top Q_d Q_d^\top)\right)^{\frac{1}{2}}, d = 1,2,3 \tag{20}$$

where $\hat{Q}_d$ and Q_d are the left singular matrices of the estimated loading matrix $\hat{A}_d$ and the true loading matrix A_d, respectively. Table I illustrate that ESWI and IPmoPCA achieve superior accuracy and consistency in estimating under normal distribution conditions, with ESWI particularly excelling in handling heavy-tailed (t_2 distribution) data across 100 tests. ESWI stands out for its robustness in both scenarios, proving to be the most stable and reliable method among the four evaluated.

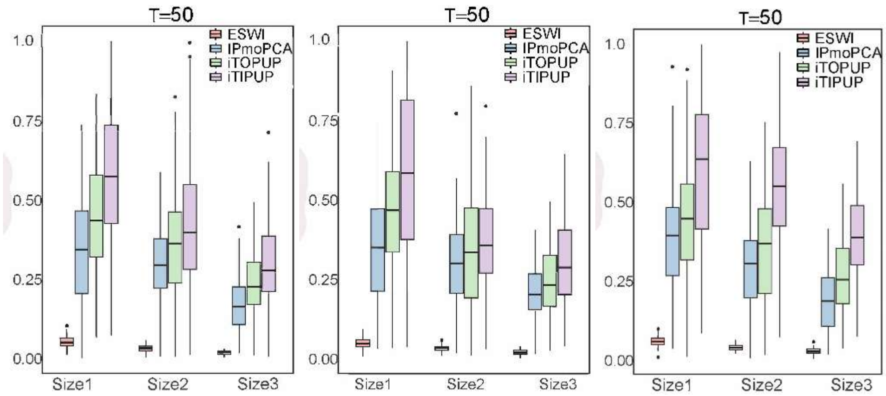


(a)$\mathcal{D}(\hat{A}_1, A_1)$ (b)$\mathcal{D}(\hat{A}_2, A_2)$ (c)$\mathcal{D}(\hat{A}_3, A_3)$

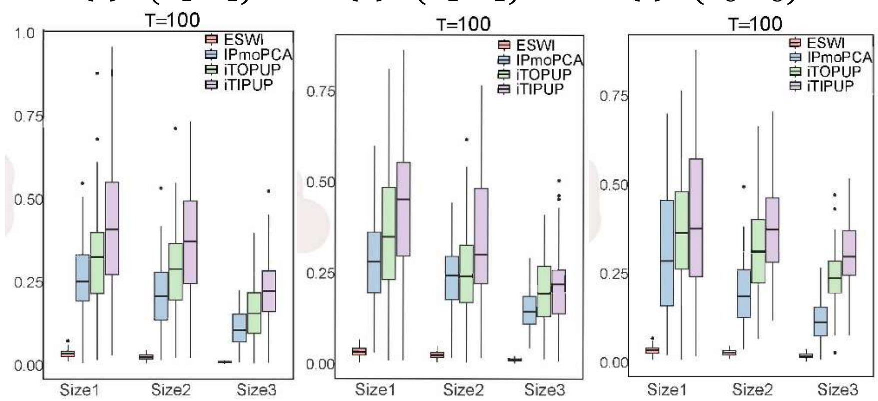


(d) $\mathcal{D}(\hat{A}_1, A_1)$ (e) $\mathcal{D}(\hat{A}_2, A_2)$ (f) $\mathcal{D}(\hat{A}_3, A_3)$

Figure 1. Under the assumption of normal distribution (non-heavy-tailed data), the calculation results of $\mathcal{D}(\hat{A}_d, A_d), d = 1,2,3$ (repeated 100 times)



(a) $\mathcal{D}(\hat{A}_1, A_1)$ (b) $\mathcal{D}(\hat{A}_2, A_2)$ (c) $\mathcal{D}(\hat{A}_3, A_3)$



(d) $\mathcal{D}(\hat{A}_1, A_1)$ (e) $\mathcal{D}(\hat{A}_2, A_2)$ (f) $\mathcal{D}(\hat{A}_3, A_3)$

Figure 2. Under the assumption of t_2 distribution(heavy-tailed data), the calculation results of $\mathcal{D}(\hat{A}_d, A_d), d = 1.2,3$ (repeated 100 times)

Figures 1 and 2 illustrate that ESWI and IPmoPCA achieve superior accuracy and consistency in estimating under normal distribution conditions, with ESWI particularly excelling in handling heavy-tailed (t_2 distribution) data across 100) tests. ESWI stands out for its robustness in both scenarios, proving to be the most stable and reliable method among the four evaluated.

3.3. Estimation of the Signal Part

This section compares various methods for estimating the common components S_t in simulated scenarios, using Mean Squared Error (MSE) between the estimated signal $\hat{S}_t$ and the true signal S_t for evaluation:

$$\text{MSE} = \frac{1}{Tp}\sum_{t=1}^{T} \|\hat{S}_t - S_t\|_F^2 \tag{21}$$

Figures 3 and 4 reveal that ESWI, IPmoPCA, and iTOPUP perform comparably in non-heavy-tailed data, but ESWI excels in heavy-tailed data with robust estimation. Overall. ESWI demonstrates superior robustness and outperforms the other methods.

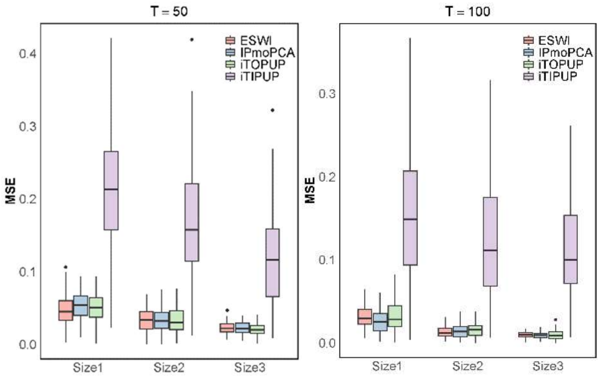


Figure 3. Estimation results of the signal part S_t under the assumption of normal distribution (non-heavy-tailed data) (repeated 100 times)

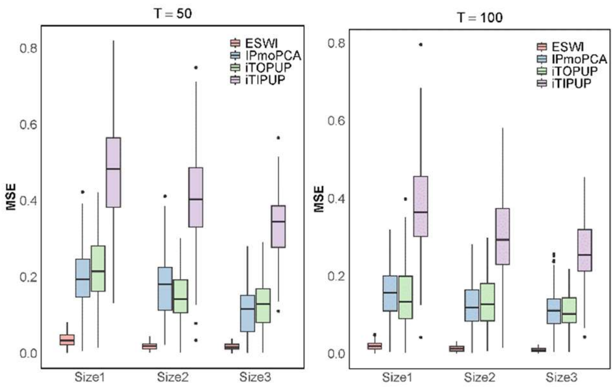


Figure 4. Estimation results of the signal part S_t under the assumption of t_2 distribution (heavy-tailed data) (repeated 100 times)

4. Application

In this section, use tensor factorization methods to a set of Moving MNIST data consisting of 100 video sequences, each comprising 20 frames. In each video sequence, two digits move independently across the frames, where each frame is an image with a spatial resolution of 64 × 64 pixels. This data can be represented as a collection of third-order tensor time series $\mathcal{X}_t \in \mathbb{R}^{64\times64\times20}$ for $t \in [100]$, where the first and second modes are the pixel row and column coordinates, respectively, and the third mode is the frame number. Figure 5 displays $\mathcal{X}_t$, for $t \in [100]$.

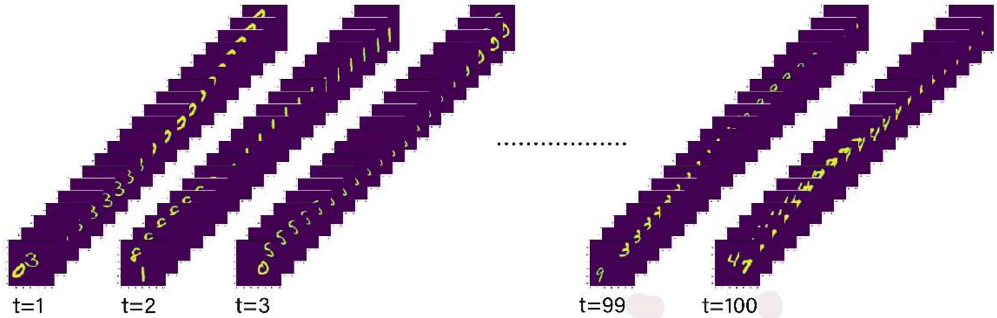


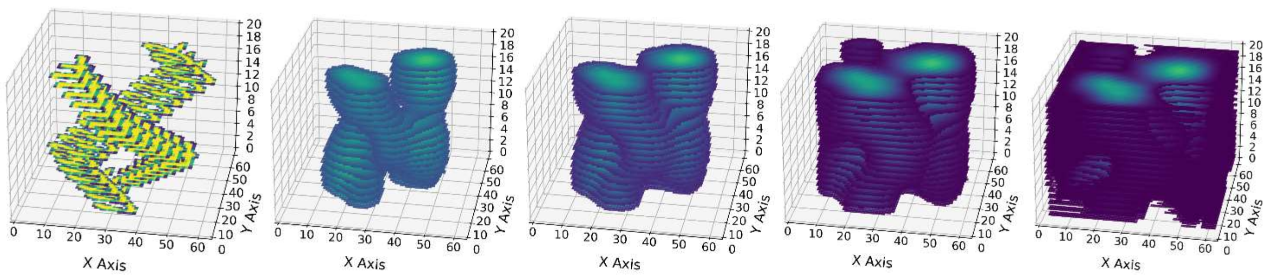
Figure 5. Third-order tensor video data $\mathcal{X}_t \in \mathbb{R}^{64\times 64\times 20}, t = 1, \cdots, 100$.

First perform tensor Tucker decomposition modeling:

$$\mathcal{X}_t = \mathcal{F}_t \times_1 A_1 \times_2 A_2 \times_3 A_3 + \mathcal{E}_t, t = 1, \cdots, 100 \tag{22}$$

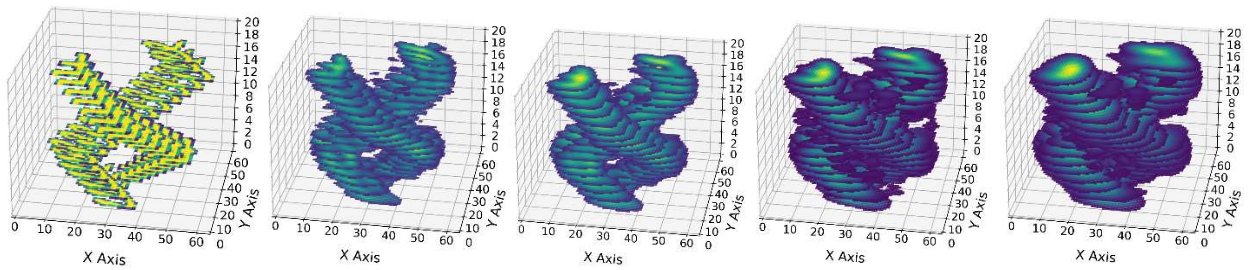
where the matrices $A_1 \in \mathbb{R}^{64\times k_1}, A_2 \in \mathbb{R}^{64\times k_2}$ and $A_3 \in \mathbb{R}^{20\times k_3}, \mathcal{F}_t \in \mathbb{R}^{k_1\times k_2\times k_3}$ are the common factor tensor. k_1, k_2 and k_3 are the numbers of factors to be determined.

Figures 6 and 7 show the real video motion and the estimated motion trajectory of the signal $\hat{S}_t$ at time $t = 15$, respectively. As the number of factors (k_1, k_2, k_3) increases, the recovery performance of the four methods becomes stronger. The ESWI method is not only better than other methods in image recovery, but also can more accurately reflect the motion patterns in the 20 -frame dynamic action of the video.



(a) Original video (b)ESWI (c)IPmoPCA (d)iTOPUP (e)iTIPUP

Figure 6. When the number of factors $k_1 = 4, k_2 = 4, k_3 = 3, t = 15$ video of $\hat{S}_t$ reconstruction for ESWI, IPmoPCA, iTOPUP, and iTIPUP.



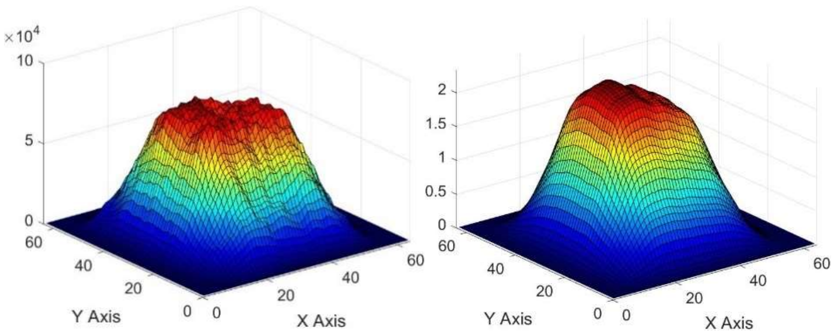
(a) Original video (b)ESWI (c)IPmoPCA (d)iTOPUP (e)iTIPUP

Figure 7. When the number of factors $k_1 = 16, k_2 = 16, k_3 = 8, t = 15$ video of $\hat{S}_t$ reconstruction for ESWI, IPmoPCA, iTOPUP, and iTIPUP.

The ESWI method performs exceptionally well in the restoration of video images. For further research, we will focus solely on analyzing using the ESWI method. Regarding the final selection

ISSN: 2664-9640

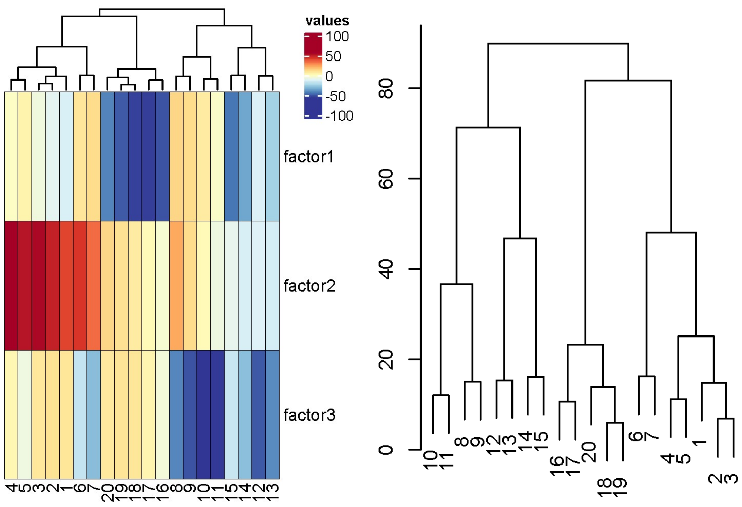
of the number of factors, to ensure that the explained variance in mode1, mode2, and mode3 is above 80%, we choose $k_1 = 4, k_2 = 4, k_3 = 3$ as the estimated number of factors. Thus the loading matrices $A_1 \in \mathbb{R}^{64\times4}, A_2 \in \mathbb{R}^{64\times4}$ and $A_3 \in \mathbb{R}^{20\times3}$. Actually, the load matrix A_1 and A_2 respectively represent the influence on the x-coordinates and y-coordinates of a 64 pixel grid. The left side of Figure 8 shows the stacked image we created from 2000 images of 64×64 pixels in $\mathcal{X}_t$; the higher the value, the more frequently that area appears in the images. It can be intuitively seen that the regions of motion in the 2000 video images primarily occur within the X-coordinate range of 15-40 and Y-coordinate range of 15-40. After multiplying the values of A_1 and A_2, the right image was plotted. It reveals that the loads A_1 and A_2 accurately reflect the main motion areas of the images, especially within the X-coordinate range of 20-35 and Y-coordinate range of 25-30, where the images appeared most frequently.



(a) Real situation (b)Estimated situation

Figure 8. Frequency diagram in video images and estimation results of load A_1 and A_2.

Figure 9 depicts the heatmap of load estimates for A_3 and the clustering results for frames 1-20. A_3 clusters frames 1-7, 8-13, and 14-20 respectively. This is because, for the motion of letters in 100 different video images, the predominant movement pattern in frames 1-7 is diagonal motion. In frames 8-13, the two letters move in opposite directions, and in frames 14-20, the two letters appear inverted and then move.



(a) Load matrix heatmap of A_3 (b)Clustering results

Figure 9. Estimated load diagram and clustering results of A_3.

5. Conclusion

This paper proposes the ESWI method for the optimization objective based on norm minimization. Through a large number of data simulations, it is verified that this method is superior to other methods. The ESWI method is then applied to a set of video data. By comparing with other methods, it can be concluded that the ESWI method can reconstruct tensor video data well, capture the information in the key pixel principal component analysis, and determine the letters in the video image. sport mode.

References

[1] R. Chen, D. Yang, and C.-H. Zhang, Factor Models for HighDimensional Tensor Time Series, Journal of the American Statistical Association, vol. 117, 2019, pp. 94-116.

[2] L. Sun and X. Chen, Bayesian Temporal Factorization for Multidimensional Time Series Prediction, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, 2019, pp. 4659-4673.

[3] Y. Han, C.-H. Zhang, and R. Chen, CP Factor Model for Dynamic Tensors, 2021,arXiv:2110.15517.

[4] Y. Han and C.-H. Zhang, Tensor Principal Component Analysis in High Dimensional CP Models, IEEE Transactions on Information Theory, vol. 69,2021 , pp. 1147-1167.

[5] Y. Han, R. Chen, D. Yang, and C.-H. Zhang, Statistical Inference for Large-dimensional Tensor Factor Model by Iterative Projections, arXiv:2206.09800.

[6] W. Chen and C. Lam, Rank and Factor Loadings Estimation in Time Series Tensor Factor Model by Pre-averaging, 2022,arXiv:2208.04012.

[7] Kong, Xinbing, Y. Liu, Long Yu and Peng Zhao. Matrix Quantile Factor Model. (2022),arXiv:2208.08 693.

[8] X. Zhang, G. Li, C. C. Liu, and J. Guo, Tucker tensor factor models: matricization and mode-wise PCA estimation, 2022,arXiv:2206.02508.