Writing Sample

Factor Modelling for Tensor Time Series Under Flexible Error Assumptions

Applicant: Jiayu LI

Department of Statistics 2021-22

Master's Thesis

December 13, 2022

Contents

Summary

In this paper, we intends to develop two new estimation methods for factor modeling of high-dimensional dynamic tensor time series. Compared with the existing estimation procedures, we relax the assumptions about the error term, assuming it is uncorrelated with common factors but can be both cross-correlated and weakly auto-correlated. Simulations of new methods are carried out under different experimental settings, and the results obtained are analyzed and compared with those obtained by existing methods(TOPUP method). From simulation results, it can be concluded that our proposed methods derive more accurate and stable estimators, accompanied with faster convergence rate. Finally, we apply our methods to a real dynamic transport network data for demonstration and model interpretations.

1 Introduction

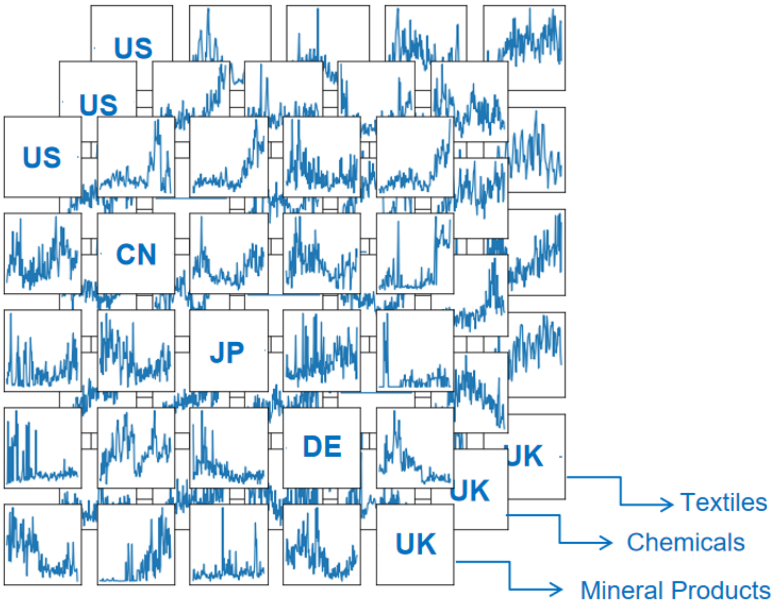


Figure 1.1: Time series of monthly imports and exports from January 2010 to December 2019.

With the leaping improvement of modern data collection capabilities and high-tech data analysis tools, multivariate time series exhibiting high dimensionality began to appear in various fields such as physics, economics and environmental science, accompanied with a large number of intensively studies. Meanwhile, the structure of multivariate time series has also developed, from vector time series and matrix time series until to the recent tensor time series. For example, a typical tensor time series can be formed by the monthly imports and exports sequence of different products categories among different countries, as shown in Figure 1.1. Specifically, each observation of the sequence can be seen as an order-3 tensor, with missing diagonal elements of each matrix slice along products categories treated as zero. We are already familiar with vector time series and panel time series analysis, which mainly focus on the co-movement of one specific fiber in the tensor(e.g. Export volume of China to USA in all commodity categories, or from another perspective, textiles export volume of China to all other countries). Additionally, arisen in recent years, studies on matrix time series deal with a matrix slice of a tensor(e.g. Imports and exports activities of one product category among different countries). However, no matter what structure is, those multivariate time series usually exhibit high dimensionality, which causes difficulties for implementing traditional time series analysis methods. While one approach to solve this problem is to apply factor modelling. As the factor model provides us a simpler-structured time series, revealing a deeper insight into the series, which makes the subsequent exploration along the time dimension easier. In this paper, We consider to develop two new estimation procedures for factor modelling of tensor-structured time series. Compared with previous estimation methods proposed by (Chen, Yang, and Zhang, 2019), (Han et al., 2020) and (Han, Chen, and Zhang, 2022), we relax the assumptions about the error term. Instead of assuming it is 'white', we allow the error series being both cross-correlated and temporally auto-correlated. Surprisingly, according to the simulation results, the estimators obtained by our methods are more accurate, more stable, and converge faster.

The rest of the paper is organized as follows. The second chapter reviews the literature on factor modeling for all kinds of high-dimensional multi-variate time series, and the mainstream model estimation procedures are summarized and classified. It should be pointed out that the methods proposed in this paper essentially belong to one of those mainstream estimation procedures. And the proposed methods can be seen as an innovative extension from matrix time series application to higher-order tensor time series application. In Chapter 3, the existing definition and interpretations of factor model for tensor time series are used. However, for model assumptions, the restriction that the error series is 'white' is relaxed, and we allow it to have a weak auto-correlation. Note that this model assumption only appears in modeling lower-order structured time series(i.e. vector, panel, matrix time series), and no such paper has considered extending it to tensor time series, to the best of our knowledge. In Chapter 4, we mainly discuss two estimation procedures; starting from introducing model identifications, we then derive two estimators of loading matrices from two kinds of statistics constructed based on model assumptions, respectively. Finally, the estimation of latent factors is given. In Chapter 5, the estimation procedures are applied to several simulation data with different statistical characteristics, and the results obtained are compared with the results obtained by existing methods. We also construct several evaluation metrics, hoping to conduct a comparative analysis on the accuracy, stability and convergence speed of the estimators. Chapter 6 carries out an application on real tensor time series data, the results obtained by our methods and an existing method are compared and further analyzed to demonstrate the factor modeling.

2 Literature Review

The core point of factor modelling for high-dimensional multi-variate time series is to estimate latent factors. While so far the proposed estimation methods, whether it is for vector time series, matrix time series or tensor time series, basically belong to the category of spectral method. The idea behind is to do eigenvalue decomposition or singular value decomposition of designed moments-type statistics. In this case, two popular types of statistics are constructed under two different assumptions about the error series of the factor model. And two types of estimators are derived from those two different statistics, respectively.

2.1 Lagged-auto-covariance-based estimator

(Lam, Yao, and Bathia, 2011) and (Lam and Yao, 2012) creatively developed a lagged-auto-covariance-based estimator for their vector factor model under the assumption that the error process is 'white'. Specifically, consider a vector time series $y_t \in \mathbb{R}^N, t = 1, \dots, T$, where y_t is an N-dimensional vector; and the corresponding vector factor model $y_t = Af_t + \epsilon_t$, which decomposes y_t into a signal Af_t and an idiosyncratic error ϵ_t. In their model, $\{f_t\}$ is an r-dimensional factor time series($r \ll N$), $A \in \mathbb{R}^{N \times r}$ is a constant loading matrix and $\{\epsilon_t\}$ is assumed to be a mean zero white noise and uncorrelated with factor series. Under their assumptions, the latent factors accommodate all dynamics of the time series. Hence we have the following equations,

$$\Sigma_y(k) = A\Sigma_f(k)A^T, \quad k = 1, 2, \cdots \tag{2.1}$$

where $\Sigma_y(k) = cov(y_{t+k}, y_t)$ and $\Sigma_f(k) = cov(f_{t+k}, f_t)$. We can see that the lagged auto-covariance of $\{y_t\}$ dose not include any auto-covariance of $\{\epsilon_t\}$ or covariance matriix between $\{f_t\}$ and $\{\epsilon_t\}$. Based on such observation, they build a lagged-auto-covariance-based statistics $L = \sum_{k=1}^{k_0} \Sigma_y(k)\Sigma_y^T(k)$, where k_0 is a hyperparameter. After performing eigenanalysis on the sample version of L, the estimated loading matrix $\hat{A}$ can be formed by taking its orthonormal eigenvectors corresponding to r largest eigenvalues and the estimated factor $\hat{f}_t = \hat{A}^T y_t$. Subsequently, (Wang, Liu, and Chen, 2019)and (Chen, Tsay, and Chen, 2019) retained the assumptions of the error term, and extended the lagged-auto-covariance-based estimator to matrix-leveled factor model. Almost at the same time,(Chen, Yang, and Zhang, 2019) extended the lagged-auto-covariance-based estimator to the tensor-leveled factor model by developing **TOPUP**(Time series Outer-Product Unfolding Procedure) and **TIPOP**(Time series Inner-Product Unfolding Procedure).

2.2 Covariance-based estimator

Seeing error process as 'white' is reasonable since the estimation results obtained by doing so usually have great sample properties. However, sometimes, adopting this assumption may encounter some problems in real applications. For example, when the observation is weakly auto-correlated or even temporally independent, the lagged(lag ≥ 1) population auto-covariance of the time series is very close to zero, which means it is difficult to do eigenvalue decomposition or singular value decomposition on sample statistics build on sample lagged-auto-covariance. Therefore, some scholars have considered more flexible assumptions on error term and constructed another type of statistics. For vector factor model,(Bai, 2003) and (Bai and Ng, 2002) firstly discussed that the zero-mean idiosyncratic error can be both cross-correlated and weakly auto-correlated. Based on their assumptions and after normalizing the covariance matrix of $\{f_t\}$ to be an identity matrix, we have $\Sigma_y = AA^T + \Sigma_\epsilon$, where Σ_y and Σ_ϵ are covariance martices of $\{y_t\}$ and $\{\epsilon_t\}$, respectively. Under the pervasiveness assumption(Stock and Watson, 2002), the covariance-based estimator for loading matrix A is developed by taking eigenvectors of Σ_y corresponding to its r largest eigenvalues. Then the asymptotic properties for consistent estimates are discussed as dimension N goes to infinity. (Chen, Fan, and Li, 2020) extended this idea to matrix-variate time series and constructed statistics based on both the first and second order moments of the time series(i.e. mean and covariance). We still classify the estimator derived from such statistics as covariance-based estimator since one of the additive part of the statistics is the covariance. Furthermore, (Forni and Hallin, 2017) raised a very interesting point, they combined both covariance and auto-covariance when building statistics for estimating a full dynamic factor model of vector time series.

Inspired by (Chen, Fan, and Li, 2020), we extend their flexible assumptions about the error process from matrix time series factor modeling to tensor time series factor modeling and propose two new covariance-based statistics. Specifically, we assume that the error process is uncorrelated with factors but can be both cross-correlated and weakly auto-correlated. Since one of the proposed statistics is of tensor structure, we need to operate tensor decomposition to obtain the estimators of loading matrices. In this paper, we adopt the Tucker decomposition(Kolda and Bader, 2009) formation and apply HOOI(**H**igher **O**rder **O**rthogonal **I**teration) algorithm (Sheehan and Saad, 2007) to approximating decomposition. Note that this method still belongs to the regime of spectral method since the Tucker decomposition can be seen as the realization of singular value decomposition for higher-order tensor.

3 Tensor Factor Model

Before introducing the factor model for tensor time series, let's briefly do some review of tensors. A tensor itself is a multidimensional array. The 'mode' of a tensor is the analogue of the dimension of an array and the 'fibers' of a tensor can be obtained by fixing all except for one of the modes. Following (Chen, Yang, and Zhang, 2019), an order-K tensor is denoted by $\mathcal{X} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$; a typical element of $\mathcal{X}$ is $\mathcal{X}_{i_1,\ldots,i_K}$ or $\mathcal{X}(i_1, i_2, \ldots, i_K), 1 \leq i_k \leq d_k$. An inner product defined on tensors is

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{d_1} \sum_{i_2=1}^{d_2} \cdots \sum_{i_K=1}^{d_K} \mathcal{X}_{i_1,\ldots,i_K} \mathcal{Y}_{i_1,\ldots,i_K}$$

which induces the Frobenius norm $\|\mathcal{X}\|_F = \langle \mathcal{X}, \mathcal{X} \rangle^{1/2}$. The mode-k product of $\mathcal{X}$ with a matrix $A \in \mathbb{R}^{\tilde{d_k} \times d_k}$ is an order-K tensor of size $d_1 \times \cdots d_{k-1} \times \tilde{d_k} \times d_{k+1} \times \cdots \times d_K$ and will be denoted by $\mathcal{X} \times_k A$. Elementwise,

$$(\mathcal{X} \times_k A)_{i_1, i_{k-1}, j, i_{k+1}, \cdots, i_K} = \sum_{i_k=1}^{d_k} \mathcal{X}_{i_1, \cdots, i_k, \cdots, i_K} A_{j, i_k}$$

Let $d = \prod_{k=1}^{K} d_k$ and $d_{-k} = d/d_k$. A $d_k \times d_{-k}$ matrix is the result of unfolding tensor $\mathcal{X}$ along mode-K and can be denoted as $mat_k(\mathcal{X})$, the columns of this matrix are composed of all mode-k fibers of $\mathcal{X}$ in a certain order. Additionally, tensor $\mathcal{X}$ can also be vectorized by stacking all its mode-1 fibers in the order from mode 2 to mode K and the result of vectorization can be denoted as $vec(\mathcal{X}) \in \mathbb{R}^d$.

3.1 Model Definition

Following (Chen, Yang, and Zhang, 2019), consider a tensor time series $\mathcal{X}_t \in \mathbb{R}^{d_1 \times \cdots \times d_K}, t = 1, \ldots, T$, where $\mathcal{X}_t$ is an order-K tensor. We firstly introduce the following decomposition,

$$\mathcal{X}_t = \mathcal{S}_t + \varepsilon_t \quad t = 1, \ldots, T \tag{3.1}$$

where $\mathcal{S}_t$ and ε_t are known as signal and idiosyncratic error, respectively. Here we assume that $\mathcal{S}_t$ has a lower-rank component and can be represented in a certain multilinear decomposition formation. Furthermore, the existing lower-rank component in this decomposition involves all randomness and dynamics of the signal part. We call it common factors, denoted by $\mathcal{F}_t, t = 1, \ldots, T$. While other components within this decomposition are assumed to be constant and will be called the loading matrices. In terms of the error term, we assume that ε_t is uncorrelated with common factors but can be both cross-correlated and weakly auto-correlated.

For more detailed decomposition, we also adopt (Chen, Yang, and Zhang, 2019)'s formation

$$\mathcal{X}_t = \mathcal{F}_t \times_1 A_1 \times_2 \ldots \times_K A_K + \varepsilon_t \quad t = 1, \ldots, T \tag{3.2}$$

where $\{\mathcal{F}_t\}$ is the factor times series of dimension $r_1 \times \cdots \times r_K$ with $r_k \ll d_k$ and A_k are $d_k \times r_k$ loading matrices of rank r_k.

Remark: Here we briefly provide some interpretations of this model. Firstly denote $\mathcal{F}_t \times_{i=1}^{k-1} A_i \times_{i=k+1}^{K} A_i$ by $f_t^{(k)}$, and it is easy to know $\mathcal{X}_t = f_t^{(k)} \times_k A_k + \varepsilon_t$. Then for all mode-k fibers of $\mathcal{X}_t$, we have

$$\begin{aligned} \mathcal{X}_t(i_1, \ldots, i_{k-1}, :, i_{k+1}, \ldots, i_K) = A_k f_t^{(k)}&(i_1, \ldots, i_{k-1}, :, i_{k+1}, \ldots i_K) \\ &+ \varepsilon_t(i_1, \ldots, i_{k-1}, :, i_{k+1}, \ldots, i_K) \end{aligned} \tag{3.3}$$

Now consider mode-1 fibers,

$$\mathcal{X}_t(:, i_2, \ldots, i_K) = A_1 f_t^{(1)}(:, i_2, \ldots, i_K) + \varepsilon_t(:, i_2, \ldots, i_K)$$

by stacking all those fibers in the order from mode 2 to mode K, we have

$$vec(\mathcal{X}_t) = KP(I_{d_{-1}}, A_1)vec(f_t^{(1)}) + vec(\varepsilon_t)$$

where $KP(I_{d_{-1}}, A_1) \in \mathbb{R}^{d \times d_{-1} r_1}$ denotes the Kronecker product between $I_{d_{-1}}$ and A_1, $I_{d_{-1}} \in \mathbb{R}^{d_{-1} \times d_{-1}}$ is the identity matrix. If we continue to decompose $f_t^{(1)}$ until getting the latent factor $\mathcal{F}_t$, the final equation will be derived as

$$vec(\mathcal{X}_t) = KP(A_K, \ldots, A_1)vec(\mathcal{F}_t) + vec(\varepsilon_t) \tag{3.4}$$

where $KP(A_K, \ldots, A_1) \in \mathbb{R}^{d \times r}$, $d = \prod_{k=1}^K d_k$, $r = \prod_{k=1}^K r_k$. Hence, from the whole perspective, $\mathcal{F}_t \to \mathcal{X}_t$ is a linear mapping from $\mathbb{R}^{r_1 \times \cdots \times r_K}$ to $\mathbb{R}^{d_1 \times \cdots \times d_K}$, and a vectorized $\mathcal{X}_t$ can be represented as a block matrix operating on vectorized $\mathcal{F}_t$. Note that if we see $vec(\mathcal{X}_t) \in \mathbb{R}^{d^2}, t = 1, 2, \ldots, T$ as a panel time series, then equation(3.4) is a special case of the vector factor model since it has a specially structured loading matrix. The Kronecker expression only has $d_1 r_1 + \ldots + d_K r_K$ parameters to be estimated, while for a traditional vector factor model, the number of parameters to be estimated can be $dr = d_1 r_1 \ldots d_K r_K$. And that is a very significant dimension reduction.

3.2 Model Assumptions

Here we state the necessary model assumptions on which our estimations are derived. The whole assumption part is an extension of the matrix factor model assumptions suggested by (Chen, Fan, and Li, 2020) to the tensor factor model. Let's first clarify the notations. Denote the centralized factor by $\tilde{\mathcal{F}}_t = \mathcal{F}_t - \frac{1}{T}\sum_{t=1}^T \mathcal{F}_t$ and the mode-k unfolding tensor $mat_k(\varepsilon_t)$ by $e_t^{(k)}$; the $ij-th$ element of $e_t^{(k)}$ is $e_{t,ij}^{(k)}$, $i \in [d_k]$ and $j \in [d_{-k}]$. For a matrix X, we use $\|X\|_\infty$ and $\|X\|_2$ to denote l_∞ and l_2 norm, respectively. For order-K tensors $X, \mathcal{Y} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$, we use $\bigodot$ to denote the tensor product. Specifically, $\mathcal{X} \bigodot \mathcal{Y}$ is an order-2K tensor of size $d_1 \times d_1 \times d_2 \times d_2 \times \ldots \times d_K \times d_K$ with element $(\mathcal{X} \bigodot \mathcal{Y})_{i_1 j_1 i_2 j_2 \ldots i_K j_K} = \mathcal{X}_{i_1 i_2 \ldots i_K} \mathcal{Y}_{j_1 j_2 \ldots j_K}$.

Assumption A. Latent Factor. Assume $\mathbb{E}[\|\mathcal{F}_t\|_F^2] \leq c < \infty$, and as $T \to \infty$, we have $\frac{1}{T}\sum_{t=1}^{T} F_t \xrightarrow{\mathcal{P}} \mu_F$,

$$\frac{1}{T}\sum_{t=1}^{T} mat_k(\tilde{\mathcal{F}}_t)mat_k^T(\tilde{\mathcal{F}}_t) \xrightarrow{\mathcal{P}} \mathbf{U}_k \quad \text{and} \quad \frac{1}{T}\sum_{t=1}^{T} \tilde{\mathcal{F}}_t \bigodot \tilde{\mathcal{F}}_t \xrightarrow{\mathcal{P}} \mathbf{V}_F \tag{3.5}$$

where $\mathbf{U}_k \in \mathbb{R}^{r_k \times r_k}, k \in [K]$ are some positive definite matrices and an order-2K tensor $\mathbf{V}_F$ is of size $r_1 \times r_1 \times \ldots \times r_K \times r_K$.

Assumption B. Loading Matrices. For each row of $A_k, k \in [K]$, $\|A_{k,i\cdot}\|_2 \leq c < \infty$. And as $d_k \to \infty$, we have $\|d_k^{-1}A_k^T A_k - \Omega_k\|_2 \to 0$ for some $r_k \times r_k$ positive definite matrices Ω_k. Assumption B is an extension of the pervasive assumption (Stock and Watson, 2002) to the tensor structured data. Under such assumption, no fiber of the latent factor $\mathcal{F}_t$ contributes negligibly to the whole variance of the observation $\mathcal{X}_t$. Therefore, the eigenvalues obtained by our estimations are extremely larger than the eigenvalues of the error term so that the latter can be completely ignored.

Assumption C. Error Process. In this paper, we adopt relatively flexible assumptions about the error term. Assuming that the error process is uncorrelated with the latent factors but can be both cross-correlated and weakly auto-correlated with at most geometrically decaying auto-covariance. Specifically, for cross-correlation of $\{\varepsilon_t\}$, we have:

1. For all $(i_1, i_2, \ldots, i_K)$, $\quad \mathbb{E}[\varepsilon_{t,i_1\ldots i_K}] = 0$ and $\mathbb{E}|\varepsilon_{t,i_1\ldots i_K}|^4 \leq c < \infty$

2. For all $k \in [K]$, let $W_k = \mathbb{E}[\frac{1}{d_{-k}T}\sum_{t=1}^{T}(e_t^{(k)})' e_t^{(k)}]$, we assume $\|W_k\|_\infty \leq c < \infty$.

3. For all $k \in [K]$ and $j, h \in [d_{-k}]$, we assume $\sum_{i \neq l} |Cov[e_{t,ij}^{(k)} e_{t,ih}^{(k)}, e_{t,lj}^{(k)} e_{t,lh}^{(k)}]| \leq c < \infty$.

Assumption C.2 implies that, for all $k \in [K]$ and $j \in [d_{-k}]$,

$$\sum_{l=1}^{d_{-k}} |\mathbb{E}[\frac{1}{d_{-k}T}\sum_{t=1}^{T}\sum_{i=1}^{d_k} e_{t,il}^{(k)} e_{t,ij}^{(k)}]| \leq c < \infty$$

and assumption C.3 holds through relatively sparse $W_k, k \in [K]$.

4 Estimation procedures

Since we only observe $\{\mathcal{X}_t\}$ and everything on the right side of the model (3.2) is unknown to us, our first task is to make sure those components are separately identifiable. While according to our pervasiveness assumption(Assumption B) on loading matrices and the bounded covariance assumption(Assumption C) on the idiosyncratic error, the signal part $\mathcal{S}_t = \mathcal{F}_t \times_{k=1}^{K} A_k$ and the error ε_t can be easily separated by ignoring error covariance when applying statistics decomposition. In terms of loading matrices, we learn from the Tensor decomposition that they can be estimated up to an affine transformation as for some non-singular matrices $H_k \in \mathbb{R}^{r_k \times r_k}, k = 1, \ldots, K$, $(\mathcal{F}_t \times_{k=1}^{K} H_k^{-1}) \times_{k=1}^{K} (A_k H_k)$ is equivalent to $\mathcal{F}_t \times_{k=1}^{K} A_k$ under model (3.2). Thus, without loss of generality, it is a good practice to restrict estimators $\widehat{A_k}, k = 1, \ldots, K$ to orthogonal matrices. According to previous researches conducted by (Chen, Fan, and Li, 2020) and (Sheehan and Saad, 2007), knowing linear transformation $\mathcal{F}_t \times_{k=1}^{K} H_k^{-1}$ and $A_k H_k, k = 1, \ldots, K$ is as good as knowing the ground truth $\mathcal{F}_t$ and $A_k, k = 1, \ldots, K$ for many analysis purposes such as dimension reduction and regression analysis.
Note that in order to simplify the expression, here we assume $\{\mathcal{F}_t\}$ has mean zero and $\{\mathcal{X}_t\}$ are centered.

4.1 Covariance Matrix Based Statistics

Note that the mode-k unfolding of $\mathcal{X}_t$ has the following equivalent expressions under model (3.2).

$$\begin{aligned} mat_k(\mathcal{X}_t) &= mat_k(\mathcal{F}_t \times_{i=1}^{K} A_i) + mat_k(\varepsilon_t) \\ &= A_k mat_k(f_t^{(k)}) + mat_k(\varepsilon_t) \end{aligned} \tag{4.1}$$

Recall that we have denoted $\mathcal{F}_t \times_{i=1}^{k-1} A_i \times_{i=k+1}^{K} A_i$ by $f_t^{(k)}$. So it is a natural thought to develop the second order moment of such matricized $\mathcal{X}_t$,

$$\begin{aligned}&\mathbb{E}[mat_k(\mathcal{X}_t)mat_k^T(\mathcal{X}_t)]\\&\quad= A_k\mathbb{E}[mat_k(f_t^{(k)})mat_k^T(f_t^{(k)})]A_k^T + \mathbb{E}[mat_k(\varepsilon_t)mat_k^T(\varepsilon_t)]\end{aligned} \tag{4.2}$$

where the error process $\{\varepsilon_t\}$ is assumed to be uncorrelated with the latent factors. Furthermore, note that the matrix $\mathbb{E}[mat_k(f_t^{(k)})mat_k^T(f_t^{(k)})]$ is of dimension $r_k \times r_k$ and the term $\mathbb{E}[mat_k(\varepsilon_t)mat_k^T(\varepsilon_t)]$ can be ignored according to the pervasiveness assumption(Assumption B). Hence, A_k can be derived as the top r_k eigenvectors of $\mathbb{E}[mat_k(\mathcal{X}_t)mat_k^T(\mathcal{X}_t)]$.

In practice, the consistent estimator of $\mathbb{E}[mat_k(\mathcal{X}_t)mat_k^T(\mathcal{X}_t)]$ is derived as $\widehat{\sum}_k = \frac{1}{T}\sum_{t=1}^{T} mat_k(\mathcal{X}_t)mat_k^T(\mathcal{X}_t)$ based on the assumption A. And the estimated loading matrices $\widehat{A_k}, k = 1, \ldots, K$ are obtained as the top r_k orthonormal eigenvectors of $\widehat{\sum}_k$ stacked in descending order by corresponding eigenvalues.

Remark: As we know, a commonly used heuristic to operate factor analysis on tensor time series is to unfold tensor into a matrix and apply matrix factor modelling, because the matrix factor model has been extensively studied in the literature. For example, consider a factor model in the form of

$$X_t = RF_tC^T + E_t, \quad t = 1, \ldots, T \tag{4.3}$$

where X_t is a $d_k \times d_{-k}$ matrix; $F_t \in \mathbb{R}^{r_k \times r_{-k}}$ is the common factor; R is a $d_k \times r_k$ front loading matrix and C is a $d_{-k} \times r_{-k}$ back loading matrix. If we assume $d_{-k} = r_{-k}$ and $C = I_{d_{-k}}$, then only focusing on one dimension in tensor factor modeling, let's say the k-th mode of the observed tensor $\mathcal{X}_t$, will be equivalent to considering matrix factor modelling on $mat_k(\mathcal{X}_t)$ under such model assumptions. Hence, the first estimation method essentially can be seen as performing matrix factor analysis sequentially by transforming tensorial observation into a matrix through mode-k unfolding, $k = 1, \ldots, K$.

4.2 Covariance Tensor Based Statistics

Here we firstly define the covariance tensor $Cov[\mathcal{X}_t] \in R^{d_1 \times d_1 \times \cdots \times d_K \times d_K}$, with its element $Cov[X_t]_{i_1 j_1 \cdots i_K j_K} = Cov[\mathcal{X}_{t,i_1 \cdots i_K}, \mathcal{X}_{t,j_1 \cdots j_K}]$. By definition, we have

$$Cov[\mathcal{X}_t] = Cov[\mathcal{F}_t] \times_1^2 A_1 \times_3^4 A_2 \cdots \times_{2K-1}^{2K} A_K + Cov[\varepsilon_t] \tag{4.4}$$

where the error process $\{\varepsilon_t\}$ is assumed to be uncorrelated with the latent factors. After ignoring the term $Cov[\varepsilon_t]$ based on the assumption B and C, the decomposition (4.4) can be viewed as a constrained Tucker decomposition problem: each loading matrix appears twice in sequence.

Based on the assumption A, the consistent estimator of $Cov[\mathcal{X}_t]$ is derived as $\widehat{\sum} = \frac{1}{T} \sum_{t=1}^{T} \mathcal{X}_t \bigodot \mathcal{X}_t$, where $\bigodot$ represents the tensor product defined in Section 3.2. In terms of the estimated loading matrices $\widehat{A_k}, k = 1, \ldots, K$, we adopt Higher Order Orthogonal Iteration(HOOI) algorithm to compute those approximations. According to (Sheehan and Saad, 2007), the basic idea is to solve the $Rank - \{r_1, r_2, \ldots, r_K\}$ optimization problem such that

$$\min_{A_1, A_2, \ldots, A_K} \|\mathcal{X} - \mathcal{F} \times_1^2 A_1 \times_3^4 A_2 \cdots \times_{2K-1}^{2K} A_K\|_F^2 \tag{4.5}$$

where $\mathcal{X}$ is the consistent estimator $\widehat{\sum}$ and the optimal $\mathcal{F}$ is given by $\mathcal{F} = \mathcal{X} \times_1^2 A_1^T \times_3^4 A_2^T \cdots \times_{2K-1}^{2K} A_K^T$.

Here we only state HOOI algorithm for an order-6 tensor and the extension to higher order is straightforward. Specifically, let $\mathcal{X}_t \in \mathbb{R}^{d_1 \times d_2 \times d_3}, t = 1, \ldots, T$, then $\widehat{\sum} \in \mathbb{R}^{d_1 \times d_1 \times d_2 \times d_2 \times d_3 \times d_3}$; besides, the same as Section 4.1, we also set $\widehat{A_i}^T \widehat{A_i} = I_{r_i}, i = 1, 2, 3$ for given r_1, r_2 and r_3. As shown in Algorithm 1.

Algorithm 1 Higher Order Orthogonal Iteration(HOOI) Algorithm

Input: $\widehat{\sum}, r_1, r_2, r_3$

Output: $\widehat{A_1} \in \mathbb{R}^{d_1 \times r_1}$, $\widehat{A_2} \in \mathbb{R}^{d_2 \times r_2}$, $\widehat{A_3} \in \mathbb{R}^{d_3 \times r_3}$;

Choose initial $\widehat{A_2}$ and $\widehat{A_3}$ with orthonormal columns.

Initial count $m = 0$.

Until $\widehat{A_1}$, $\widehat{A_2}$, $\widehat{A_3}$ Converge Do:

$Q = \widehat{\sum} \times_3^4 \widehat{A_2}^T \times_5^6 \widehat{A_3}^T$

m = m+1

if m is odd **then**

$\widehat{A_1} = SVD[r_1, mat_1(Q)]$;

else

$\widehat{A_1} = SVD[r_1, mat_2(Q)]$;

end if

$U = \widehat{\sum} \times_1^2 \widehat{A_1}^T \times_5^6 \widehat{A_3}^T$

if m is odd **then**

$\widehat{A_2} = SVD[r_2, mat_3(U)]$;

else

$\widehat{A_2} = SVD[r_2, mat_4(U)]$;

end if

$V = \widehat{\sum} \times_1^2 \widehat{A_1}^T \times_3^4 \widehat{A_2}^T$

if m is odd **then**

$\widehat{A_3} = SVD[r_3, mat_5(V)]$;

else

$\widehat{A_3} = SVD[r_3, mat_6(V)]$;

end if

End Do

return $\widehat{A_1}$, $\widehat{A_2}$, $\widehat{A_3}$

Remark I: $SVD[r, M]$ used in Algorithm 1 means to compute singular value decomposition of matrix M and the output of $SVD[r, M]$ are the r largest left singular vectors of M, stacked in descending order by corresponding singular values.

Remark II: In factor analysis, Tucker decomposition is often used to model tensor-structured data in the form of a lower-rank core tensor being multiplied (or transformed) by a matrix along each mode. Specifically, for an order-2K tensor $\mathcal{Y} \in \mathbb{R}^{d_1 \times d_1 \times \cdots \times d_K \times d_K}$, Tucker Decomposition can be denoted as $\mathcal{Y} = \Upsilon \times_{k=1}^{K} U_k \times_{k=K+1}^{2K} U_k$, where the core tensor $\Upsilon \in \mathbb{R}^{r_1 \times r_1 \times \cdots \times r_K \times r_K}$ is an order-2K tensor that contains mode-k(k=1,. . . ,2K) singular values of $\mathcal{Y}$, which are defined as the Frobenius norm of the mode-k slices of tensor $\mathcal{Y}$. $U_k, k = 1, \ldots, 2K$ are orthogonal matrices in $\mathbb{R}^{d_k \times r_k}$. Then it can be easily found that the decomposition (4.4) is of constrained Tucker decomposition formation such that $U_{2k-1} = U_{2k}, k = 1, \ldots, K$. Hence, the second estimation method is a combination of spectral aggregation and tensor decomposition, processing the entire tensor time series simultaneously.

4.3 Estimation of the factor and the signal tensor

After estimating loading matrices, we obtain an estimator of $\widehat{\mathcal{F}}_t = \mathcal{X}_t \times_{k=1}^K \widehat{A}_K^T$ and the corresponding signal part $\widehat{\mathcal{S}}_t = \mathcal{X}_t \times_{k=1}^K \widehat{A}_K^T \times_{k=1}^K \widehat{A}_K$ based on the model definition (3.1). The above estimation procedures assume that the latent dimensions $r_k, k = 1, \ldots, K$ are known. Whereas in real applications we need to estimate them as well. Here we adopt the ratio-based estimator introduced by (Lam and Yao, 2012). For all $k \in [K]$, let $\widehat{\lambda}_{k,1} \geq \widehat{\lambda}_{k,2} \geq \cdots \geq \widehat{\lambda}_{k,m} \geq 0$ be the eigenvalues of $\widehat{\sum}_k$. Then the estimator of r_k is defined as

$$\widehat{r_k} = \underset{1 \leq j \leq k_{max}}{\arg\max} \frac{\widehat{\lambda}_{k,j}}{\widehat{\lambda}_{k,j+1}} \tag{4.6}$$

where k_{max} is a given upper bound and in practice we may take $k_{max} = \lceil d_k/2 \rceil$ or $k_{max} = \lceil d_k/3 \rceil$ according to (Lam and Yao, 2012).

5 Simulation

In this section, some tensor time series are generated under specific settings, in order to enable numerical analysis of the convergence rates and the comparison of estimation accuracy among different methods. Specifically, we present the following results obtained by three estimation methods (TOPUP proposed by (Chen, Yang, and Zhang, 2019), Method 1 and Method 2 introduced in this paper) under different experimental settings in the subsequent subsections.

Result 1 (Convergence of $\widehat{A}_k, k = 1, \dots, K$.) We first introduce the column space distance between the true loading matrices and the estimated loading matrices. That is for any rank r_k matrices $\widehat{A}, A \in \mathbb{R}^{d_k \times r_k}$, the column space distance is defined as:

$$\mathcal{D}(A, \widehat{A}) = \|\widehat{A}(\widehat{A}^T\widehat{A})^{-1}\widehat{A}^T - A(A^TA)^{-1}A^T\|_2 \tag{5.1}$$

This evaluation metric can not only be used to measure the estimation accuracy, but also by cooperating with different settings, it can be used to numerically analyze and compare the convergence rates of the estimators of $A_k, k = 1, \dots, K$ retrieved from three estimation methods. Additionally, in order to demonstrate the stability of the estimation procedures and ensure the randomness of the generated data, we repeated our simulations 10 times under each setting and report box plots of the space distances.

Result 2 (Convergence of $\widehat{\mathcal{S}}_t$.) Although it is very tempting to directly measure the estimation error of the common factor, it is not that meaningful to calculate the difference between the estimates $\widehat{\mathcal{F}}_t$ and the ground truth $\mathcal{F}_t$, noting that the estimator we obtain is a transformation of the true factor. Therefore, here we choose to calculate the estimation error of the signal part in order to analyze and compare the estimation accuracy of different procedures.

Note that we have already described the calculation of the estimated signals in Section 4.3, and the estimation error can be obtained by $\mathcal{S}_t - \widehat{\mathcal{S}}_t$. Then, we adopt the Frobenius norm of the estimation error (i.e. $\|\mathcal{S}_t - \widehat{\mathcal{S}}_t\|_F$) to conveniently represent the distance between the estimated signal and the ground truth.

Hence, here we introduce the following metric to measure, from an overall perspective, the estimation accuracy of the methods. Suppose for each experimental setting we repeat simulation N times and every simulated time series is of fixed length T. Then we define the averaged distance $\mathcal{D}_{Ave}(\mathcal{S}, \widehat{\mathcal{S}}) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\|\mathcal{S}_t^{(i)} - \widehat{\mathcal{S}}_t^{(i)}\|_F$ as the evaluation metric towards the accuracy of estimating signals.

5.1 Settings

Throughout, the order-3 tensor observations $\mathcal{X}_t \in \mathbb{R}^{d_1 \times d_2 \times d_3}, t = 1, \ldots, T$ are generated according to model 3.2. The latent factor $\mathcal{F}_t$ is of fixed size $3 \times 2 \times 2$ and the rest including the dimension of the tensor (d_1, d_2, d_3), the length of the time series (T) will vary to facilitate our exploration. The elements of the true loading matrices $A_1 \in \mathbb{R}^{d_1 \times 3}, A_2 \in \mathbb{R}^{d_2 \times 2}, A_3 \in \mathbb{R}^{d_3 \times 2}$ are independently sampled from the uniform distribution $\mathcal{U}(-2, 4)$. The latent factors and the error terms are allowed to be correlated across all fibers or time dimension, in different experimental settings we specified later.

Setting I: [Error process is set to be weakly cross-correlated but temporally uncorrelated (Known as 'white').]

As briefly discussed in Section 2, the lagged-auto-covariance-based estimator may converge at a relatively low speed or even unable to converge when the time series has a very weak auto-correlation, while the covariance-based estimator has no such problem. That is why the papers introducing the former estimators, such as (Lam and Yao, 2012) and (Chen, Yang, and Zhang, 2019), have to set some restrictions on the auto-correlation of the signal part in model assumptions. The auto-correlation strength of the signal part is referred as the "signal strength" in those papers and we continue to use that name in our paper. In Setting I, we mainly focus on the convergence rate of the estimators obtained by three methods under different signal strengths. For convenience, we firstly set the error process 'white'.

Specifically, we first simulate $vec(\mathcal{F}_t) \in \mathbb{R}^{12}, t = 1, \ldots, T$ from the order one Vector Auto-Regressive model (VAR(1)).

$$vec(\mathcal{F}_t) = \Phi \cdot vec(\mathcal{F}_{t-1}) + \epsilon_t, \tag{5.2}$$

where the AR coefficient matrix $\Phi = \varphi \cdot I_{12}$ and $Var[\epsilon_t] = (1 - \varphi^2) \cdot I_{12}$. Thus $Var[vec(\mathcal{F}_t)] = I_{12}$. Then we reshape the vectors $\{vec(\mathcal{F}_t)\}$ into tensors $\mathcal{F}_t$. Note that by setting different values to φ, we can generate time series with different signal strengths.

The simulation of the error term is a bit complicated as we require the entries of ε_t are temporally uncorrelated but weakly cross-correlated. We first independently sample $d_2 \times d_3$ vectors from multivariate normal distribution $\mathcal{N}_{d_1}(0, I_{d_1})$ and stack them to form a $\mathbb{R}^{d_1 \times d_2 \times d_3}$ tensor $\mathcal{O}_t$. Then the error is generated by $\varepsilon_t = \mathcal{O}_t \times_1 U_E \times_2 V_E \times_3 W_E$, where the matrices $U_E \in \mathbb{R}^{d_1 \times d_1}, V_E \in \mathbb{R}^{d_2 \times d_2}, W_E \in \mathbb{R}^{d_3 \times d_3}$ all have 1's on the diagonal while have $1/d_1$'s, $1/d_2$'s and $1/d_3$'s off-diagonal, respectively.

Setting II: [Error process is set to be both cross-correlated and auto-correlated (Flexible assumptions).]

Compared to (Chen, Yang, and Zhang, 2019) and related papers where the error process is assumed to be 'white', we relax the assumptions in this paper, allowing that the error term can be both cross-correlated and auto-correlated. By doing so, we hope that such flexible assumption enables the model to cope with more complicated situations, since in real data analysis, the noise contained in data tends to have more complex structure. In Setting II, by changing the auto-correlation strength of the error series, we want to explore and compare the tolerance of three methods towards the complexity of the noise structure and its impact on convergence rate and the estimation accuracy.

Specifically, $\{\mathcal{F}_t\}$ is the same as in Setting I, with φ set to the fixed number(0.6) in model (5.2). Similarly, we simulate $\{vec(\varepsilon_t)\}$ also from VAR(1) model.

$$vec(\varepsilon_t) = \Psi \cdot vec(\varepsilon_{t-1}) + e_t, \tag{5.3}$$

where the AR coefficient matrix $\Psi = \psi \cdot I_d$, $(d = d_1 \times d_2 \times d_3)$ and $Var[e_t] = (1-\psi^2) \cdot I_d$. Then we reshape the vectors $vec(\varepsilon_t), t = 1, \ldots, T$ into $\mathbb{R}^{d_1 \times d_2 \times d_3}$ tensors in turn. Note that by setting different values to ψ, we can generate noise series with different auto-correlation strengths.

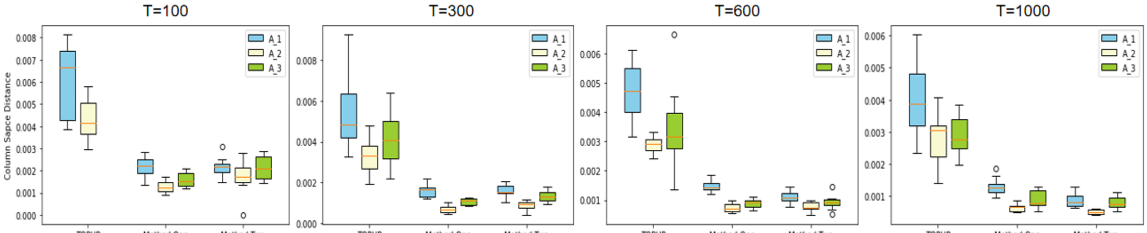
5.2 Simulation Results

In Setting I, we set $\varphi = 0.2$ and $\varphi = 0.6$ those two different signal strengths, two sets of dimensions $d_1 = d_2 = d_3 = 10$ and $d_1 = 15, d_2 = 15, d_3 = 10$, and we set the length of time series can be $T = 100, 300, 600, 1000$, respectively. The above conditions are combined to form a variety of settings, and under each setting, we operate simulations 10 times for each of the three methods. Below, Figure 5.1 reports results for loading matrices estimation under setting I with $\varphi = 0.2$ and $\varphi = 0.6$ respectively. It can be seen that the convergence of the estimators as well as their variance, have a tendency to get improved with longer length(T) of time series and larger size(d_1, d_2, d_3) of the tensor. And changes in tensor size have a more significant impact on both estimation variance and estimator convergence. Furthermore, regardless of the setting, the two methods proposed in this paper always outperform TOPUP, with better convergence and smaller estimating variance.

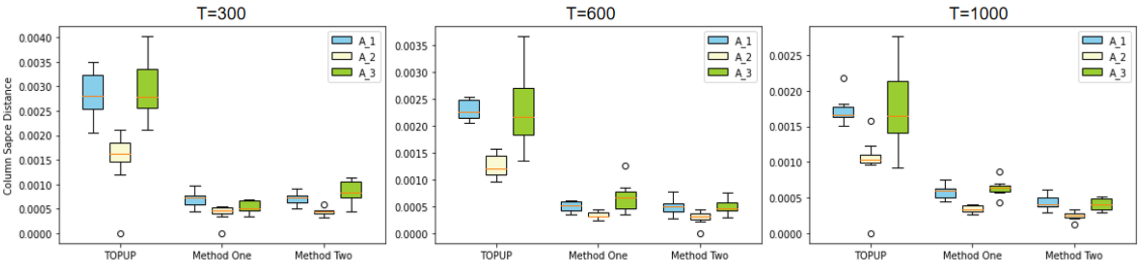
The above results of the loading matrices, unfortunately, cannot directly reflect the estimation accuracy of the model. In comparison, we are more concerned about whether the model can accurately estimate the common factors, so as to accomplish real dimension reduction. Hence, in Table 5.1, we report $\mathcal{D}_{Ave}(\mathcal{S}, \widehat{\mathcal{S}})$ (described in result 2) and the standard deviation of $\mathcal{D}(\mathcal{S}_t, \widehat{\mathcal{S}}_t)$ to intuitively show the convergence rate and estimation accuracy of the presented tensor factor model. The results obtained under settings with $d_1 = d_2 = d_3 = 10$ are present in shaded rows and results obtained from settings with $d_1 = d_2 = 15, d_3 = 10$ are recorded in highlighted rows.

It can be seen from Table 5.1 that the convergence of the signal part is consistent with

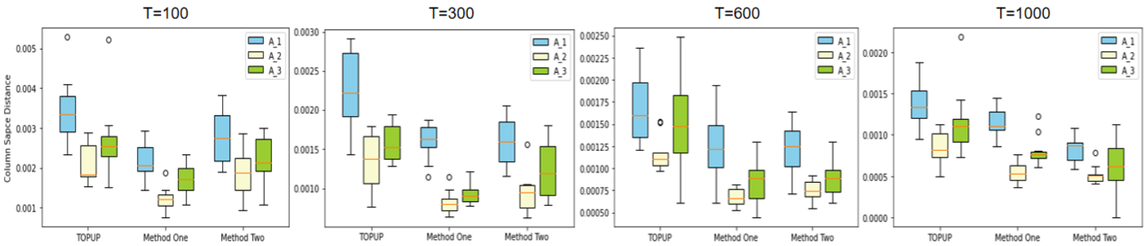
(a) Setting I, $\varphi = 0.2$, $d_1 = d_2 = d_3 = 10$



(b) Setting I, $\varphi = 0.2$, $d_1 = d_2 = 15, d_3 = 10$



(c) Setting I, $\varphi = 0.6$, $d_1 = d_2 = d_3 = 10$



(d) Setting I, $\varphi = 0.6$, $d_1 = d_2 = 15, d_3 = 10$

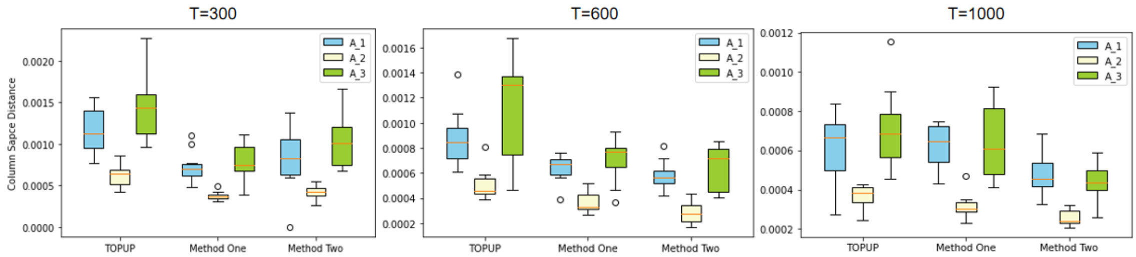


Figure 5.1: Box plots of column space distance under setting I.

the convergence of the loading matrices. Furthermore, it is observed that $\mathcal{D}_{Ave}(\mathcal{S}, \widehat{\mathcal{S}})$ obtained by Method 1 and Method 2 get stable when length(T) reaches 300, which means the signal estimators start to converge at least at length $T = 300$. While simultaneously, the convergence rate of TOPUP is much slower. Such difference is not that obvious under $\varphi = 0.6$, we can see that $\mathcal{D}_{Ave}(\mathcal{S}, \widehat{\mathcal{S}})$ obtained by TOPUP reaches the same convergence level as Method 1 and Method 2 at $T = 600$, but under $\varphi = 0.2$, it can be seen that $\mathcal{D}_{Ave}(\mathcal{S}, \widehat{\mathcal{S}})$ of TOPUP does not show convergence until $T = 1000$. This is actually in line with our expectations, since TOPUP uses lagged-auto-covariance-based estimator, when the signal strength is relatively weak, the property of the derived statistics is not good enough to operate eigenvalue decomposition, which in turn affects the convergence rate and estimation accuracy.

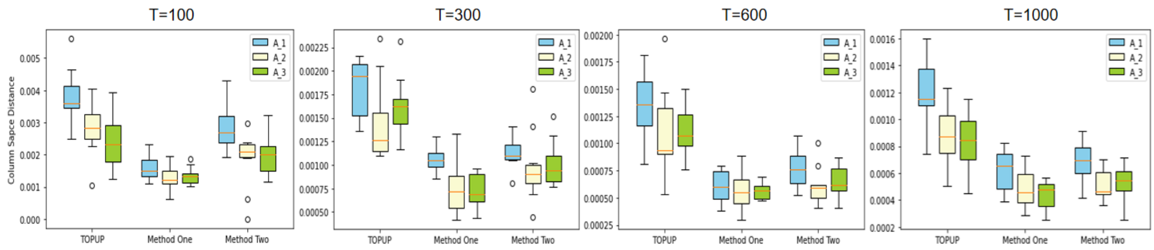
	$\varphi = 0.2$			
	T=100	T=300	T=600	T=1000
TOPUP	5.49(1.28)	5.33(1.26)	5.17(1.05)	4.92(1.14)
		5.15(1.10)	4.93(.99)	4.67(1.06)
Method One	4.66(1.03)	4.57(.99)	4.60(.98)	4.56(1.00)
		4.24(.95)	4.23(.89)	4.26(.88)
Method Two	4.73(1.10)	4.55(1.01)	4.57(.98)	4.53(.98)
		4.25(.97)	4.22(.96)	4.23(.93)
	$\varphi = 0.6$			
	T=100	T=300	T=600	T=1000
TOPUP	4.94(1.14)	4.81(1.08)	4.61(.89)	4.58(.86)
		4.45(1.01)	4.42(.92)	4.35(.91)
Method One	4.74(1.13)	4.61(1.09)	4.59(.94)	4.59(.90)
		4.34(.96)	4.36(.98)	4.35(.90)
Method Two	4.84(1.13)	4.56(1.02)	4.54(.89)	4.53(.89)
		4.36(.83)	4.34(.91)	4.34(.83)

Table 5.1: Sample mean(Standard deviation) of $\mathcal{D}(\mathcal{S}_t, \widehat{\mathcal{S}}_t)$ under Setting I.

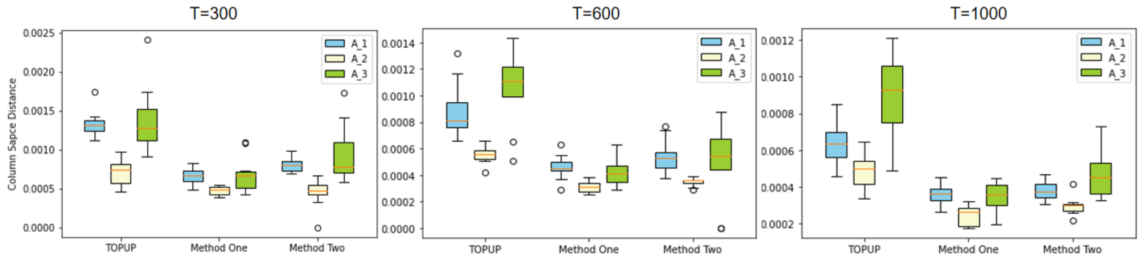
In Setting II, we set $\psi = 0.2$ and $\psi = 0.6$, two sets of dimensions $d_1 = d_2 = d_3 = 10$ and $d_1 = 15, d_2 = 15, d_3 = 10$, and we set the length of time series can be $T = 100, 300, 600, 1000$, respectively. Similar to Setting I, Figure 5.2 shows the error of loading matrices estimation under Setting II with $\psi = 0.6$ and $\psi = 0.2$. Table 5.2 shows the signal part estimation errors with the same format as in Setting I.

Comparing the results under two settings($\psi = 0.2$ and $\psi = 0.6$), we find that the convergence of TOPUP is severely affected by strong auto-correlation of the error process. As it can be seen that $\mathcal{D}_{Ave}(\mathcal{S}, \widehat{\mathcal{S}})$ obtained under $\psi = 0.6$ are much higher than that obtained under $\psi = 0.2$ at each length T. Besides, $\mathcal{D}_{Ave}(\mathcal{S}, \widehat{\mathcal{S}})$ obtained under $\psi = 0.2$ starts to converge at least at length T=600, whereas $\mathcal{D}_{Ave}(\mathcal{S}, \widehat{\mathcal{S}})$ obtained under $\psi = 0.6$ does not show convergence until T=1000 . While the Method 1 and Method 2 are barely affected by such strong auto-correlation. This is because the TOPUP is derived under the assumption that the error process is 'white'. Briefly explain, if the error is 'white', then the lagged-auto-covariance-based statistics only contains the

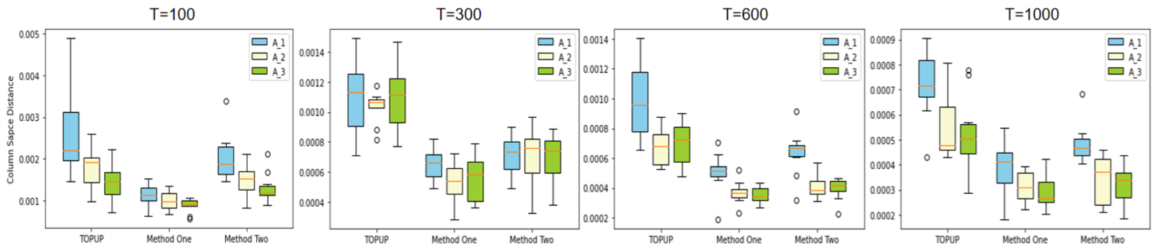
(a) Setting II, $\psi = 0.6$, $d_1 = d_2 = d_3 = 10$



(b) Setting II, $\psi = 0.6$, $d_1 = d_2 = 15, d_3 = 10$



(c) Setting II, $\psi = 0.2$, $d_1 = d_2 = d_3 = 10$



(d) Setting II, $\psi = 0.2$, $d_1 = d_2 = 15, d_3 = 10$

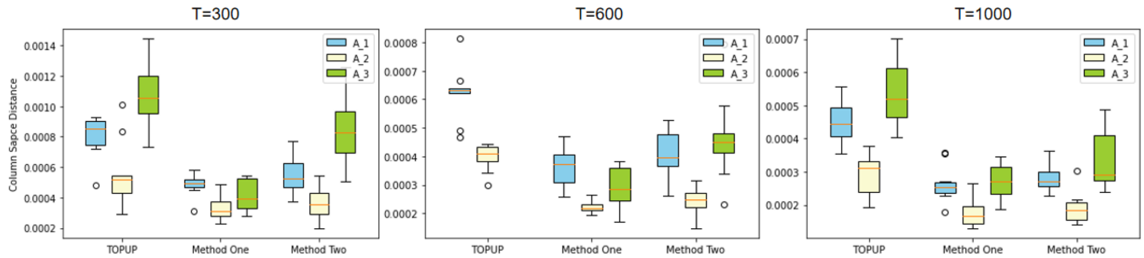


Figure 5.2: Box plots of column space distance under setting II.

information of the signal part, which is also the advantage of this statistics, quickly extracting the information from the signal part. However, if the error series has strong auto-correlation, which violates the assumption of TOPUP, the convergence of the estimator would be heavily affected due to poor sample properties of its lagged-auto-covariance-based statistics. While the covariance-based estimator, to some extent, is able to reduce the distractions caused by such auto-correlated noise.

	$\psi = 0.2$			
	T=100	T=300	T=600	T=1000
TOPUP	3.60(.71)	3.48(.68)	3.43(.76)	3.43(.72)
		3.27(.71)	3.12(.69)	3.19(.71)
Method One	3.44(.73)	3.41(.75)	3.43(.69)	3.42(.72)
		3.18(.72)	3.13(.69)	3.13(.71)
Method Two	3.51(.70)	3.42(.70)	3.43(.70)	3.42(.72)
		3.20(.72)	3.14(.69)	3.14(.71)
	$\psi = 0.6$			
	T=100	T=300	T=600	T=1000
TOPUP	3.89(.80)	3.57(.73)	3.53(.69)	3.50(.70)
		3.42(.68)	3.34(.75)	3.28(.67)
Method One	3.50(.78)	3.38(.75)	3.40(.71)	3.40(.69)
		3.20(.66)	3.11(.74)	3.09(.67)
Method Two	3.63(.79)	3.41(.75)	3.43(.70)	3.41(.70)
		3.23(.66)	3.12(.74)	3.10(.67)

Table 5.2: Sample mean(Standard deviation) of $\mathcal{D}(\mathcal{S}_t, \widehat{\mathcal{S}}_t)$ under Setting II.

6 Application

In this section, we will apply factor modelling to the trade/transport network data as illustrated in Figure 1.1. The modelling results obtained by three different estimation procedures (**TOPUP**, Method 1 and Method 2) will be presented and compared, accompanied by further analysis and interpretations on those estimated components. The data we use is obtained from UN Comtrade Database. In this study we observe the monthly export-import trade volume among the world's top 10 economies: United States(US), China(CN), Japan(JP), Germany(DE), United Kingdom(UK), India(IN), France(FR), Italy(IT), Canada(CA) and South Korea(KR). Besides, as the trade data includes Harmonized System Codes, which naturally form the commodity classifiers, we have all tradable products divided into 15 categories. Please refer to Table 6.1 for the list of detailed products categories, and their corresponding HS codes can be fund in HS Codes List. The data is collected from January 2010 to December 2019, so that the original data can be viewed as an order-3 tensor time series of size $10 \times 10 \times 15$ with length T=120. Given that no country has any import-export trade with itself, the missing diagonal elements of each matrix slice along products categories can be treated as zero. Furthermore, following (Chen, Yang, and Zhang, 2019), we also adopt a moving average smoothing technique with window length equal to 3 for original time series. Therefore, a sudden large-scale trade or a sudden reduction of trading volume due to the transportation delay can be smoothed to improve the stability of time series. Specifically, if $\mathcal{X}_t \in \mathbb{R}^{10\times10\times15}, t = 1, \ldots, 118$ is the series after data processing, its element $x_{i,j,k,t}$ represents the three-month moving average of the trading volume from export country i to import country j in product category k at month t.

According to our model definition in Section 3.1, we have the following dynamic factor model for the trade/transport network

$$\mathcal{X}_t = \mathcal{F}_t \times_1 A_1 \times_2 A_2 \times_3 A_3 + \varepsilon_t \tag{6.1}$$

where $\mathcal{X}_t \in \mathbb{R}^{10\times 10\times 15}$ is a three-month moving average of the original observation, factor $\mathcal{F}_t \in \mathbb{R}^{r_1\times r_2\times r_3}$ carries most of the dynamics of the model, $A_i \in \mathbb{R}^{d_i\times r_i}(d_1 = d_2 = 10, d_3 = 15)$ are loading matrices. The model interpretation is similar to that of (Chen and Chen, 2019) where they apply matrix factor modeling for a dynamic uni-product-category trade network, while our model just extends it to 15 product categories.
Note that in simulation part we use a ratio-based method to determine the number of latent dimensions. However, in practice, since the signal of the data is not strong enough, there often exists a 'cliff' in eigenvalues, that is, a small number of eigenvalues are extremely large, while the remaining large number of eigenvalues are very small. Under such situation, the ratio-based estimates are often much smaller than those obtained by screen plots method. Hence, in order to present and further analyze the loading matrix estimates, here we adopt screen plots method to select the number of latent dimensions, and the selected minimum number allows us to explain at least 80 percent of the variance of the corresponding statistics.

6.1 Condensed Product Groups

The estimates of A_3 are obtained by operating TOPUP with $h_0 = 2$, Method 1 and Method 2 respectively. The latent dimension $r_3 = 4$ are selected for all three methods, and a varimax rotation is used to simplify the expression of the estimated loading matrix. Besides, in order to achieve a clearer view, all the elements are multiplied by 100 and rounded. Table 6.1 shows the simplified estimates under Method 2 and the results obtained under the other two methods are displayed in Appendix A. It can be seen from Table 6.1 that, under Method 2, Machinery and Electrical, Mineral Products, Transportation and Vegetable Products are heavily loaded on factor 1, 2, 3 and 4 respectively, and other product categories are almost excluded compared to them. Therefore, each of them can be interpreted as a factor, such as Machinery and Electrical factor, Mineral Products factor and Vegetable Products factor. In addition, since Foodstuffs has significantly higher loads on factor 3 than other factors, it also can be mixed with Transportation to jointly explain factor 3. In general, the matrix shows an obvious group structure and we can view each factor as a 'condensed product group'. According to the loading vectors of each product category, we can also carry out a hierarchical clustering to obtain a more specific grouping, as shown in Figure 6.1.

	1	2	3	4
Animal and Animal Products	0	-1	-6	1
Vegetable Products	-2	0	-2	**98**
Foodstuffs	0	0	**-16**	-1
Mineral Products	0	**100**	0	0
Chemicals and Allied Industries	-3	-2	-9	-5
Plastics and Rubbers	-7	-2	-5	-5
Raw Hides, Skins, Leather and Furs	-3	0	2	5
Wood and Wood Products	-4	1	-6	4
Textiles	-6	0	4	-5
Footwear and Headgear	-5	0	3	2
Stone and Glass	-4	0	1	1
Metals	-8	-1	5	3
Machinery and Electrical	**-94**	0	2	2
Transportation	2	0	**-96**	2
Miscellaneous	-28	4	-15	-14

Table 6.1: Simplified estimates of loading matrix A_3 based on Method 2.

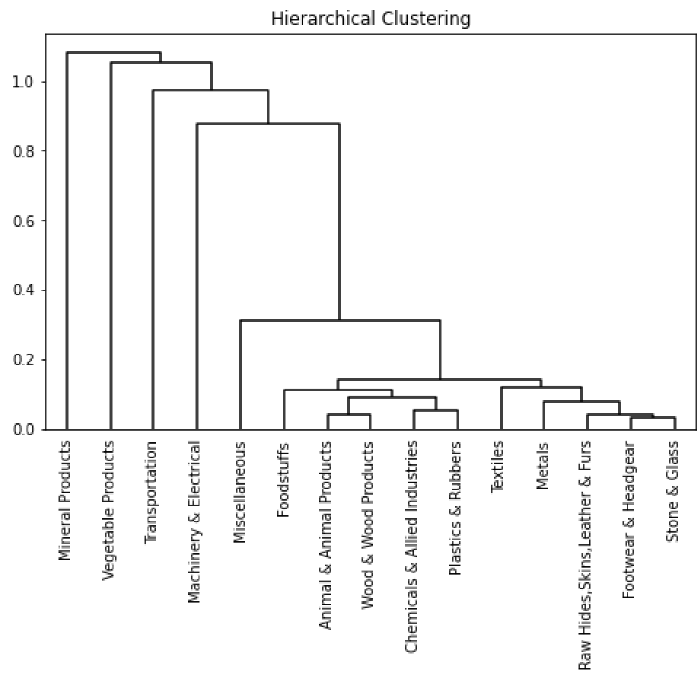


Figure 6.1: Clustering of product categories under Method 2.

Comparing the results shown in Appendix A, it can be seen that the estimate of A_3 obtained by Method 1 is very similar to that obtained by Method 2. On one hand, they share almost the same factor interpretations, that is among all product categories, Mineral Products, Transportation and Vegetable Products are exclusively dominating factor 2,3 and 4 respectively; Machinery, Electrical and Miscellaneous jointly form the factor 1. While on the other hand, the low-level clustering show some difference between Method 1 and 2. For example, Chemicals & Allied Industries, Textiles and Metals are relatively prominent for clustering analysis under Method 1 whereas under Method 2, they are grouped with other product categories, showing no such prominence.
The loading matrix obtained by TOPUP is very different from the former two. First of all, the basic interpretations of the factors is completely another one under TOPUP, where Chemicals & Allied Industries and Plastics & Rubbers have heavy loadings on factor 1 and 3, respectively; Animal Products and Wood Products jointly explain factor 2; and Mineral Products, Machinery and Electrical are used to explain factor 4. Subsequently, the hierarchical clustering constructed on its loading coefficients shows a fundamentally different grouping structure from the former two. Note that we only show the difference here and do not further study the reasons behind it and neither do we evaluate those two different results.(Maybe a follow-up research can be conducted in the future.)

6.2 Trading Hubs

Consider the following interpretation of the loading matrices A_1, A_2: Imagine that a country's exports will first go through a virtual "export hub" and then go to a virtual "import hub" before reaching another importing country. The number of columns in A_1 represents the number of such "export hubs" and the same relationship exists between the number of columns in A_2 and the number of such "import hubs". Hence, the corresponding A_1 and A_2 demonstrate how each country is connected with those import and export hubs. After varimax rotation, we normalize the simplified loading matrices so that the sum of each column of the matrices equals to 1, then we can think of each element as the percentage of each country's trade in the trading hubs. Tables 6.2 and 6.3 show the simplified estimates of A_1 and A_2 after the above processing, values are in percentage. Note that some elements are negative and it may be tricky to explain those negative values, but fortunately, there are not many negative values and their absolute values are relatively small, which does not affect our basic interpretations of the factors. Hence, here we do not further study the meaning behind those negative values.

	TOPUP		Method 1			Method 2		
	1	2	1	2	3	1	2	3
US	1	4	-1	-1	**83**	0	-1	**80**
CN	**88**	0	**91**	0	0	**91**	0	0
JP	1	5	2	8	-13	1	6	2
DE	5	13	4	12	16	4	9	10
UK	2	6	3	5	11	2	7	6
IN	2	2	2	2	7	2	2	2
FR	1	3	-1	3	-3	1	2	1
IT	1	2	2	2	1	2	2	1
CA	-1	**65**	-1	**68**	-1	0	**73**	-1
KR	1	1	-2	1	-1	-3	1	-1

Table 6.2: Estimated A_1 for the export fiber under TOPUP, Method 1 and Method 2.

	TOPUP		Method 1			Method 2		
	1	2	1	2	3	1	2	3
US	**68**	-2	**86**	0	0	**89**	0	0
CN	-6	**42**	0	0	**79**	0	**83**	0
JP	13	11	4	**33**	-12	2	-4	**33**
DE	9	10	3	**22**	-8	0	0	**27**
UK	5	8	4	7	10	4	6	7
IN	3	7	0	10	2	-1	4	12
FR	3	1	8	-6	12	5	4	-8
IT	-1	1	-1	0	1	0	3	1
CA	4	**19**	-3	**33**	14	-2	3	**26**
KR	2	3	-2	1	2	2	1	2

Table 6.3: Estimated A_2 for the import fiber under TOPUP, Method 1 and Method 2.

It can be seen that the estimates obtained by the Method 1 and 2 are very similar. In terms of exports, China, Canada and US are heavily loaded on the export hub 1, 2 and 3, respectively. In terms of imports activities, US almost contributes to the whole import hub 1, while Japan, Germany and Canada jointly form another import hub. There also exists some difference between results obtained by Method 1 and Method 2. For example, although China dominates one of the import hubs under both

Method 1 and 2, it can be seen that under Method 2, other countries except China have little effect on import hub 2 while under method 1, Japan, UK, France and Canada except China all have non-trivial contributes to import hub 3. Which means the $\widehat{A}_2$ estimated by Method 2 has more clearer structure and thus, the covariance tensor based statistics may help to identify the signal more clearly in this application. While on the other hand, TOPUP gets completely different results. Firstly, the estimated latent dimensions for A_1 and A_2 are both equal to 2. So, there are both only two hubs for imports and exports activities. In terms of export activities, US is not recognized as an export hub. In addition, for import activities, although US and China have heavy loads on hub 1 and 2, respectively, they do not take a dominant position in the hubs. The trade activities from other countries also make certain contributions to the hubs. The reason for the above difference is that as the signal of real data is relatively weak, the lagged-auto-covariance-based statistics of TOPUP often shows a poor sample property. In this case, an outstandingly large eigenvalue will appear when carrying out eigenvalue analysis on the statistics and then the estimated latent dimensions will be small no matter under ratio-based method or screen plots method.

6.3 Factors

Based on the above interpretation and analysis of each loading matrices, the common factors can be easily understood. We can think of the factor as a condensed trading network. Specifically, $\mathcal{F}_{i,j,k,t}$ can be viewed as the k-th product group being transported from the i-th export hub to the j-th import hub at time t. Figure 6.2 shows the estimated factor sequence under Method 2, wherein each "matrix slice" represents the transportation net of a condensed product group. The horizontal axis of the "matrix" represents the export hubs and the vertical axis represents the import hubs. Therefore, for trading network with a wide variety of tradable products and a large number of participating economies, we can look down at the core content of the trading network by identifying common factors. In addition, since the dimensions of the common factors are greatly reduced compared with the original observations, the modeling and analysis on the factor series become much easier. And as the factor series carries almost all the dynamics of the observation series, the subsequent analysis on observation series will be easier and clearer through separately discussing loading parts, factors and error term.

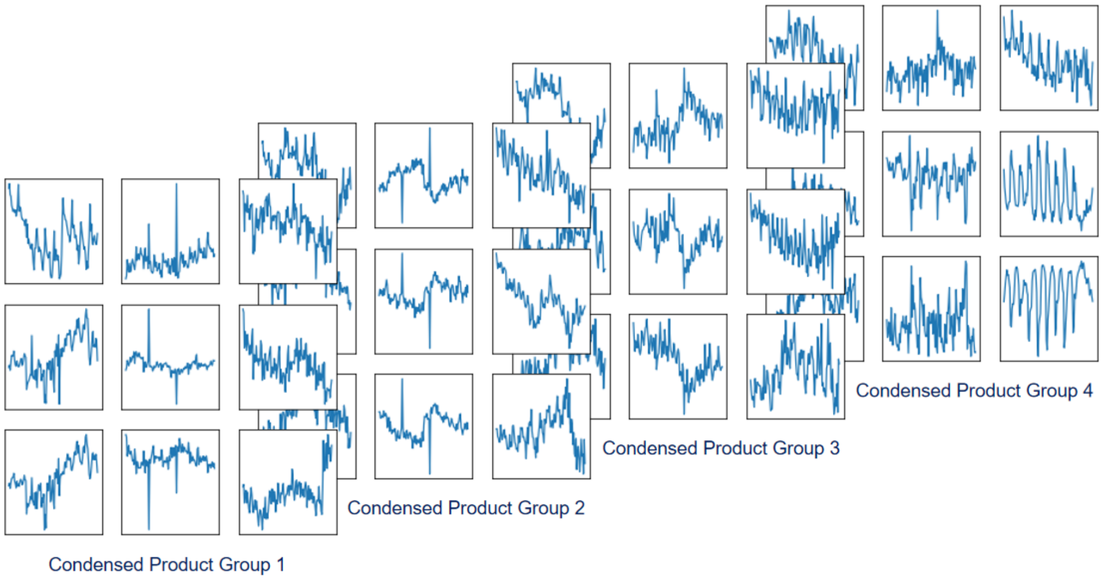


Figure 6.2: Estimated factor time series under Method 2.

7 Conclusion

Factor modeling for tensor time series has not been widely studied. Although some papers have proposed estimation methods for the common factors based on the assumption that the error term is 'white', very few literature considers more flexible assumptions, such as an auto-correlated error series or the error term can be even correlated with factors. This paper adopts the same factor model as (Chen, Yang, and Zhang, 2019) (TOPUP), but relaxes assumptions of the error term, assuming it is uncorrelated with factors but can be both cross-correlated and weakly auto-correlated. Based on the assumptions, two new estimation procedures are proposed, the obtained results are compared with that obtained by TOPUP under different simulation settings. And what's exciting is: two methods proposed in this paper get more accurate and stable estimates in simulation environment, and the estimation convergence is faster compared with TOPUP. In terms of the application, we use the transportation network data that is naturally of tensor structure as the modeling demonstration. We also interpret and analyze the estimated loading matrices and the factor series, and then follows the discussion about the significance of factor modeling for tensor time series. We hope that the two methods proposed in this paper can provide new ideas for solving tensor time series analysis.

However, the research conducted on those two new methods is not complete. Note that we only put forward the methods and preliminarily discussed the properties of estimators through obtained results in simulations. The theoretical derivations of the corresponding statistical properties and the convergence of estimators have not been done. And we hope to continue the research in the future to derive the related asymptotic properties of the estimators.

References

Bai, J. (2003). Inferential theory for factor models of large dimensions. *Econometrica*, 71(1): 135–171.

Bai, J., and Ng, S. (2002). Determining the Number of Factors in Approximate Factor Models. *Econometrica*, 70(1): 191–221.

Chen, E. Y., and Chen, R. (2019). Modeling Dynamic Transport Network with Matrix Factor Models: with an Application to International Trade Flow. *Papers*.

Chen, E. Y., Fan, J., and Li, E. (2020). Statistical Inference for High-Dimensional Matrix-Variate Factor Model.

Chen, E. Y., Tsay, R. S., and Chen, R. (2019). Constrained factor models for high-dimensional matrix-variate time series. *Journal of the American Statistical Association*.

Chen, R., Yang, D., and Zhang, C.-h. (2019). Factor Models for High-Dimensional Tensor Time Series.

Forni, M., and Hallin (2017). Dynamic factor models with infinite-dimensional factor space: Asymptotic analysis. *Journal of Econometrics*, 199(1): 74–92.

Han, Y., Chen, R., Yang, D., and Zhang, C.-H. (2020). Tensor Factor Model Estimation by Iterative Projection.

Han, Y., Chen, R., and Zhang, C.-H. (2022). Rank determination in tensor factor model. *Electronic Journal of Statistics*, 16(1).

Kolda, T. G., and Bader, B. W. (2009). Tensor Decompositions and Applications. *SIAM Review*, 51(3): 455–500.

Lam, C., and Yao, Q. (2012). FACTOR MODELING FOR HIGH-DIMENSIONAL TIME SERIES: INFERENCE FOR THE NUMBER OF FACTORS. *The Annals of Statistics*, 40(2): 694–726.

Lam, C., Yao, Q., and Bathia, N. (2011). Estimation of latent factors for high-dimensional time series. *Biometrika*, 98: 901–918.

Sheehan, B. N., and Saad, Y. (2007). Higher Order Orthogonal Iteration of Tensors (HOOI) and its Relation to PCA and GLRAM. In *SDM*.

Stock, J. H., and Watson, M. W. (2002). Forecasting Using Principal Components from a Large Number of Predictors. *Journal of the American Statistical Association*, 97(460): 1167–1179.

Wang, D., Liu, X., and Chen, R. (2019). Factor models for matrix-valued high-dimensional time series. *Journal of Econometrics*, 208(1): 231–248. Funding Information: Chen's research was supported in part by US National Science Foundation grants IIS-1741390, DMS-1503409 and DMS-1209085. Publisher Copyright: © 2018 Elsevier B.V.

A Appendices

A.1 Supplementary tables and figures

The following tables show the simplified estimates of loading matrix A_3 obtained by TOPUP and Method 1, respectively. Two matrices are rotated via varimax and then all numbers are multiplied by 100 and rounded. Table A.1 refers to TOPUP and Table A.2 refers to Method 1.

The following figures show the hierarchical clustering of product categories by their loading coefficients under TOPUP and Method 1, respectively. FigureA.1 refers to TOPUP and Figure A.2 refers to Method 1.

	1	2	3	4
Animal and Animal Products	0	**35**	-2	2
Vegetable Products	0	-7	4	-1
Foodstuffs	-2	-3	1	2
Mineral Products	-1	1	6	**-33**
Chemicals and Allied Industries	**-100**	0	0	0
Plastics and Rubbers	0	0	**99**	1
Raw Hides, Skins, Leather and Furs	-6	-2	0	-2
Wood and Wood Products	0	**-92**	0	0
Textiles	-2	-1	0	-1
Footwear and Headgear	0	-6	2	-5
Stone and Glass	-2	-9	0	0
Metals	-4	-3	-3	1
Machinery and Electrical	0	0	-1	**-94**
Transportation	-3	-6	-5	-1
Miscellaneous	-5	-4	5	7

Table A.1: Estimated loading matrix A_3 based on TOPUP.

	1	2	3	4
Animal and Animal Products	0	0	2	1
Vegetable Products	1	0	0	**99**
Foodstuffs	-1	1	6	1
Mineral Products	0	**-100**	0	0
Chemicals and Allied Industries	-5	-6	5	0
Plastics and Rubbers	-7	-2	5	1
Raw Hides, Skins, Leather and Furs	-3	0	-1	2
Wood and Wood Products	-4	0	2	4
Textiles	-13	1	-2	11
Footwear and Headgear	-6	0	-3	2
Stone and Glass	-4	-4	2	-1
Metals	-8	-4	1	6
Machinery and Electrical	**-94**	0	-1	-2
Transportation	1	0	**98**	0
Miscellaneous	**-27**	3	0	2

Table A.2: Estimated loading matrix A_3 based on Method 1.

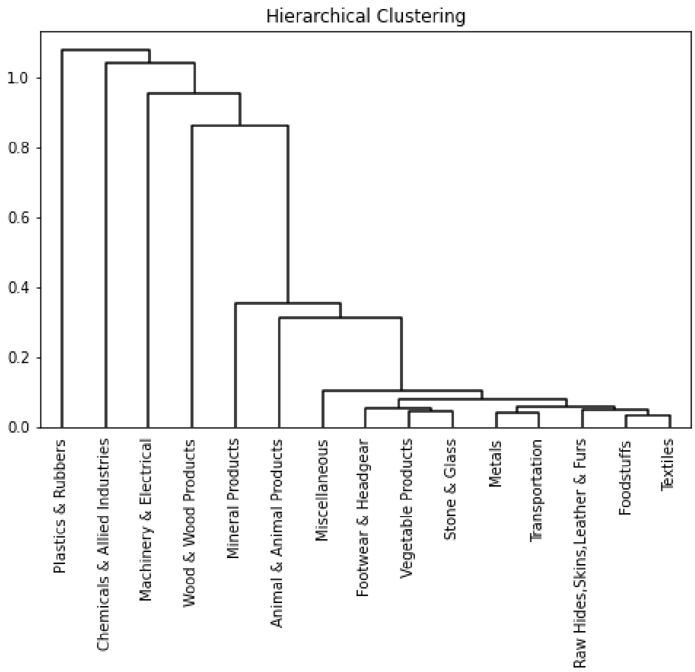


Figure A.1: Clustering of product categories under TOPUP.

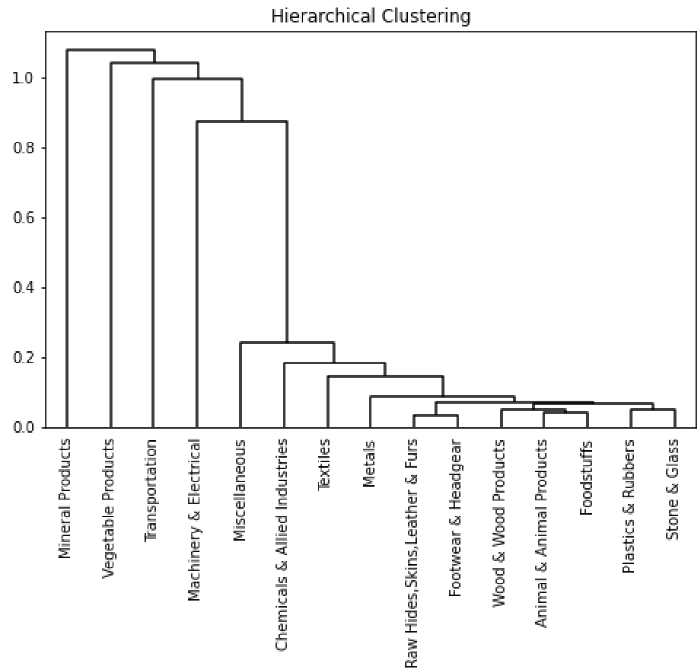


Figure A.2: Clustering of product categories under Method 1.