# Structure Analysis of High Dimensional Tensor Data

Shiyuan He

School of Statistics, Renmin University of China, China, flyoverearth@163.com

Jianxin Yin*

School of Statistics, Renmin University of China, China, jyin@ruc.edu.cn

**Abstract**

Multi-way tensor data have become prevalent in many scientific areas such as genomics and biomedical imaging. We consider a K-way tensor-normal distribution, where the precision matrix for each way has a graphical interpretation. We develop an l1 penalized maximum likelihood estimation and an efficient coordinate descent-based algorithm for model selection and estimation in such tensor normal graphical models (TNGMs). When the dimensions of the tensor are fixed, we provide theoretical results on the asymptotic distributions and oracle property for the proposed estimates of the precision matrices. When the dimensions diverge as the sample size goes to infinity, we present the rates of convergence of the estimates and the sparsistency results. Simulation results demonstrate that the TNGMs can lead to better estimates of the precision matrices and better identification of the graph structures defined by the precision matrices than the standard Gaussian graphical models. We illustrate the methods with an analysis of yeast gene expression data measured over different time points and under different experimental conditions.

**Keywords:** Gaussian graphical model; Gene networks; l1 penalized likelihood; Oracle property; Tensor normal distribution.

# 1. Introduction

An increasing number of statistical and data mining problems involve the analysis of data that are indexed by more than one way. This type of data is often called multidimensional matrix, multi-way array or tensor (De Lathauwer et al., 2000). Recently high-dimensional tensor data have become prevalent in many scientific areas such as genomics, biomedical imaging, remote sensing, bibliometrics, chemometrics and Internet. Take a two-way $n \times p$ data matrix as an example, if the $n$ samples are not independent, their correlation should be taken into consideration in statistical modeling, which leads to a transposable matrix (Allen and Tibshirani, 2010). In genetical experiments, gene expression data are often collected at different time points during the cell cycle process and under varying experimental conditions, which give rise to a 3-way tensor data (Omberg et al., 2009). In social-economics studies, export of commodity $k$ from country $i$ to country $j$ at year $t$ (Hoff, 2011) defines a three-way tensor data.

Statistical methods for tenor data analysis are relatively limited. Omberg et al. (2009) developed tensor higher-order singular value decomposition for integrative analysis of DNA microarray data from different studies. Tucker and parallel factor analysis (PARAFAC) are useful methods for tensor decomposition (Kolda 2006). When modeling the high dimensional tensor data, a separable covariance matrix structure is often assumed. Such a separable structure on the covariance matrix can dramatically reduce the dimension of the parameter space. Consider a four-way tensor data, suppose that the dimensions are $m_1 = m_2 = m_3 = 100$ and $m_4 = 10$. The nonseparable model requires a joint covariance matrix of a $10^7 \times 10^7$ entries, while the separable model requires only three $100 \times 100$ matrices and one $10 \times 10$ matrix for each way. The joint covariance matrix is simply the Kronecker product of the matrices for each way. The ratio of dimension between two

models is almost of order $10^{10}$.

In this paper, we consider the sparse modeling of the precision matrices of the $K$-way tensor data assuming a separable covariance matrix structure. The corresponding precision matrices define graphical models for tensor data. In many applications, sparsity in each of the corresponding precision matrices can be assumed to facilitate the interpretation. In addition, tensor normality is a natural assumption on the distribution when the data are continuous (Hoff, 2011). With the separability assumption on the covariance matrix, the joint covariance matrix of the vectorization of the tensor can be obtained by a Kronecker product of $K$ covariance matrices.

When $K$=2, the 2-way normal tensor data is also called the matrix normal data. Yin and Li (2012) discussed the sparse model selection and estimation for matrix normal distribution using a penalized likelihood approach with Lasso and adaptive Lasso penalties. In their work, the dimensions for row and column can diverge to infinity when the sample size goes to infinity. Other related works in modeling matrix-normal data include Allen and Tibshirani (2010), Zhou (2012), Zhang and Schneider (2010), Tsiligkarids et al. (2012).

In this paper, we generalize the work by Yin and Li (2012) to $K$-way tensor data and focus our work on graphical model selection and estimation. We develop a penalized likelihood approach with the adaptive Lasso penalty. The consistency and oracle property are obtained when the dimensions hold fixed. We show that the explicit rate of convergence and sparsistency property hold when the dimensions diverge with sample size going to infinity. We further show that the effective sample size for estimating the covariance matrix in each way of the tensor is the product of the number of independent samples and the dimensions of the other $K-1$ matrices. It is worth noting that this effective sample size is usually very large, hence the convergence is quite fast and the high dimension is actually

a bless. Our simulation study demonstrates the high accuracy in estimating the precision matrices with small sample size $N$.

The rest of the paper is organized as follows. A brief summary of multi-way tensor data is presented in Section 2. Section 3 introduces the definition of array normal distribution of Hoff (2011) and its estimation in high dimensional settings. The convexity and optimization of the objective function is discussed in Section 4. In Section 5, the asymptotic properties are derived both for the case of the fixed dimensions and the case of diverging dimensions when sample size goes to infinity. A Monte Carlo simulation study is presented in Section 6. Finally, a 3-way tensor data set on gene expressions (Omberg et al., 2009) is analyzed in Section 7.

## 2. Multi-way Tensor Data Structure and Operations

We first present a brief summary of multi-way array data or high order tensor data (Hoff, 2011; De Lathauwer et al. 2000). Tensor data are higher order parallels of vectors and matrices. Entries in a vector can be indexed by a single index set, while a matrix is indexed by two sets (row and column). In the following presentation, scalers are non-bold italic letters, vectors are represented by bold-faced lower case letters, and matrices and multi-way tensor are written as bold-faced capitals. For a matrix $\mathbf{A}$, we use $\mathbf{a}(j)$ to denote its $j$-th column, $\mathbf{a}[i]$ its $i$-th row, and $A(i,j)$ its $(i,j)$-th element.

The K-way tensor is an arrangement of elements, which is indexed by $K$ sets. Suppose $\mathbf{Y}$ is a K-array tensor with dimensions $\{m_1, m_2, \cdots, m_K\}$, then the total number of elements of $\mathbf{Y}$ is $m = m_1 \times m_2 \times \cdots \times m_K$. All the elements in $Y$ are

$$\{y_{(i_1,\cdots,i_K)} : i_k = 1, 2, \cdots, m_k;\ k = 1, 2, \cdots, K\}.$$

Clearly, $\mathbf{Y}$ is a vector when $K = 1$ and a matrix when $K = 2$. We further introduce the notation $\mathbf{Y}_{(\cdots, i_k^0, \cdots)}$, which is a $(K-1)$-subarray of $\mathbf{Y}$. Specifically, $\mathbf{Y}_{(\cdots, i_k^0, \cdots)}$ has the same elements as $\mathbf{Y}$, except that its $k$-th sub-index is fixed at $i_k^0$. In other words, all the elements in $\mathbf{Y}_{(\cdots, i_k^0, \cdots)}$ are

$$\{y_{(i_1, \cdots, i_k^0, \cdots, i_K)} : i_h = 1, 2, \cdots, m_h;\ h = 1, 2, \cdots, k-1, k+1, \cdots, K\}.$$

To analyze the properties of $K$-way tensor, it is helpful to relate the tensor with vector or matrix. The vectorization of $\mathbf{Y}$ is a vector of dimension $m$,

$$\begin{aligned}\text{vec}(\ \mathbf{Y}) = \big(&y_{(1,1,1,\cdots,1)}, y_{(2,1,1,\cdots,1)}, \cdots, y_{(m_1,1,1,\cdots,1)},\\ &y_{(1,2,1,\cdots,1)}, y_{(2,2,1,\cdots,1)}, \cdots, y_{(m_1,2,1,\cdots,1)},\\ &\cdots,\\ &y_{(1,m_2,1,\cdots,1)}, y_{(2,m_2,1,\cdots,1)}, \cdots, y_{(m_1,m_2,1,\cdots,1)},\\ &\cdots,\\ &y_{(1,m_2,m_3,\cdots,m_K)}, y_{(2,m_2,m_3,\cdots,m_K)}, \cdots, y_{(m_1,m_2,\cdots,m_K)}\big)^T.\end{aligned}$$

To be explicit, $y_{(i_1,\cdots,i_K)}$ is the $j$-th element of vec$(\mathbf{Y})$ with

$$j = \sum_{k=2}^{K} \Big[(i_k - 1)\Big(\prod_{l=1}^{k-1} m_l\Big)\Big] + i_1.$$

On the other hand, the $k$-mode matrix unfolding results in a $m_k \times (m/m_k)$ matrix, $\mathbf{Y}_{(k)}$, whose $i_k^0$-th row is $[\text{vec}(\mathbf{Y}_{(\cdots, i_k^0, \cdots)})]^T$ for $i_k^0 = 1, 2, \cdots, m_k$.

The $k$-mode product of a $m_1 \times \cdots \times m_K$ K-array $\mathbf{Y}$ and a $n \times m_k$ matrix $\mathbf{A}$ is a K-array $\mathbf{Z}$ with dimensions $\{m_1, \cdots, m_{k-1}, n, m_{k+1}, \cdots, m_K\}$. The product is denoted by

$\mathbf{Y} \times_k \mathbf{A}$, and the $(i_1, \cdots, i_K)$-th element of $\mathbf{Z}$ is

$$z_{(i_1,\cdots,i_K)} = \sum_{l=1}^{m_k} a_{(i_k,\, l)} y_{(i_1,\cdots,\, i_{k-1},\, l,\, i_{k+1},\cdots,\, i_K)}.$$

Tucker product is defined based on $k$-mode product and is useful for the definition of the tensor normal distribution. For a list of matrices $\mathbf{A} = \{\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_K\}$ with $\mathbf{A}_k$ being of dimension $n_k \times m_k$, the Tucker product of a $m_1 \times \cdots \times m_K$ K-way tensor $\mathbf{Y}$ and $\mathbf{A}$ is

$$\mathbf{Y} \times \mathbf{A} = \mathbf{Y} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \cdots \times_K \mathbf{A}_K.$$

Let $\mathbf{Z} = \mathbf{Y} \times \mathbf{A}$, we have the following formula that connects the $k$-mode unfolding and Tucker product,

$$\mathbf{Z}_{(k)} = \mathbf{A}_k \mathbf{Y}_{(k)} \big(\mathbf{A}_K \otimes \cdots \otimes \mathbf{A}_{k+1} \otimes \mathbf{A}_{k-1} \otimes \cdots \otimes \mathbf{A}_1\big)^T. \tag{2.1}$$

## 3. Tensor Normal Distribution and Penalized Likelihood Estimation

Our main method builds on the tensor normal distribution introduced by Hoff (2011). Without loss of generality, we assume the mean is zero, and our focus is the estimation of covariance and precision matrices. The probability density function of a tensor normal distribution with zero mean and covariances $\mathbf{\Sigma} = \{\mathbf{\Sigma}_1, \cdots, \mathbf{\Sigma}_K\}$ is

$$p(\mathbf{Y}|\mathbf{\Sigma}_1, \cdots, \mathbf{\Sigma}_K) = (2\pi)^{-m/2} \Big( \prod_{k=1}^{K} |\mathbf{\Sigma}_k|^{-m/(2m_k)} \Big) \times \exp(-\|\mathbf{Y} \times \mathbf{\Sigma}^{-1/2}\|^2/2),$$

where $\|\mathbf{Y}\|^2 = \sum_{i_1,\cdots,i_K} y^2_{(i_1,\cdots,i_K)}$ and $\mathbf{\Sigma}^{-1/2} = \{\mathbf{\Sigma}_1^{-1/2}, \cdots, \mathbf{\Sigma}_K^{-1/2}\}$. The tensor normal distribution is denoted by $\mathbf{Y} \sim \text{anorm}(\mathbf{0}, \mathbf{\Sigma}_1 \circ \mathbf{\Sigma}_2 \circ \cdots \circ \mathbf{\Sigma}_K)$. This definition includes vector normal distribution ($K = 1$) and matrix normal distribution ($K = 2$) as special

cases. For $k = 1, 2, \cdots, K$, the inverse of $\mathbf{\Sigma}_k$ is called precision matrix or concentration matrix, denoted by $\mathbf{\Omega}_k$. For the purpose of identifiability, we assume

$$\Omega_2(1,1) = \Omega_3(1,1) = \cdots = \Omega_K(1,1) = 1, \tag{3.1}$$

which requires the $(1,1)$ entries of $\mathbf{\Omega}_2, \mathbf{\Omega}_3, \cdots, \mathbf{\Omega}_K$ to be one.

Derived from (2.1), some properties for tensor normal distribution are given below.

**Lemma 1.** *Let* $\mathbf{Z} = \mathbf{Y} \times \mathbf{\Sigma}^{-1/2}$, $\mathbf{V} = \mathbf{Y}_{(k)}(\mathbf{\Omega}_K^{1/2} \otimes \cdots \otimes \mathbf{\Omega}_{k+1}^{1/2} \otimes \mathbf{\Omega}_{k-1}^{1/2} \otimes \cdots \otimes \mathbf{\Omega}_1^{1/2})^T$, *and let* $\mathbf{v}(j)$ *be the* $j$*-th column of* $\mathbf{V}$*, then we have*

*(i)* $$\begin{aligned}\|\mathbf{Y} \times \mathbf{\Sigma}^{-1/2}\|^2 = \text{tr}\Big(\mathbf{V}^T \mathbf{\Omega}_k \mathbf{V}\Big) &= \textstyle\sum_{j=1}^{m/m_k} \mathbf{v}(j)^T \mathbf{\Omega}_k \mathbf{v}(j) \\ &= \text{vec}(\mathbf{Y})^T (\mathbf{\Omega}_K \otimes \cdots \otimes \mathbf{\Omega}_1) \text{vec}(\mathbf{Y})\end{aligned}$$

*(ii)* $\text{vec}(\mathbf{Y}) \sim \text{N}(\mathbf{0}, \mathbf{\Sigma}_K \otimes \mathbf{\Sigma}_{K-1} \otimes \cdots \otimes \mathbf{\Sigma}_1)$

*(iii)* $\mathbf{Y}$ *can be expressed as*

$$\mathbf{Y} = \mathbf{Z} \times \mathbf{\Sigma}^{1/2}$$

*with* $\mathbf{\Sigma}^{1/2} = \{\mathbf{\Sigma}_1^{1/2}, \cdots, \mathbf{\Sigma}_K^{1/2}\}$ *and* $\mathbf{Z} \sim \text{anorm}(\mathbf{0}, \mathbf{I}_1 \circ \mathbf{I}_2 \circ \cdots \circ \mathbf{I}_K)$.

Assume that we have $N$ i.i.d. observations $\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_N$ from a tensor normal distribution with zero mean, we are interested in estimating the true covariance matrices $\{\mathbf{\Sigma}_1^0, \cdots, \mathbf{\Sigma}_K^0\}$ and their corresponding true precision matrices $\{\mathbf{\Omega}_1^0, \cdots, \mathbf{\Omega}_K^0\}$. In high dimensional settings, under the sparsity assumption of the precision matrices, we propose

to estimate these $K$ precision matrices by maximizing the following penalized likelihood function,

$$\begin{aligned} & \frac{1}{N}\sum_{n=1}^{N}\log(p(\mathbf{Y}_n|\mathbf{\Omega}_1,\cdots,\mathbf{\Omega}_K)) - \sum_{k=1}^{K}\lambda_k\sum_{i\neq j}p\big(\Omega_k(i,j)\big) \qquad (3.2)\\ =& -\frac{m}{2}\log(2\pi) + \sum_{k=1}^{K}\frac{m}{2m_k}\log|\mathbf{\Omega}_k| \\ & -\frac{1}{2N}\sum_{n=1}^{N}\text{vec}(\mathbf{Y}_n)^T(\mathbf{\Omega}_K\otimes\cdots\otimes\mathbf{\Omega}_1)\text{vec}(\mathbf{Y}_n) - \sum_{k=1}^{K}\lambda_k\sum_{i\neq j}p\big(\Omega_k(i,j)\big), \end{aligned}$$

where $\Omega_k(i,j)$ is the $(i,j)$-th element of $\mathbf{\Omega}_k$ and $\lambda_k$'s are the tuning parameters. We focus on the $\ell_1$ norm or Lasso penalty $p(\cdot) = |\cdot|$ and the adaptive Lasso penalty $p(\cdot) = |\cdot|/|\widetilde{\Omega}_k(i,j)|^\gamma$ where $\widetilde{\Omega}_k(i,j)$ is a consistent estimator of $\Omega_k(i,j)$.

Maximizing (3.2) is equivalent to minimizing

$$\begin{aligned} q(\mathbf{\Omega_1},\cdots,\mathbf{\Omega}_K) &= -\sum_{k=1}^{K}\frac{m}{m_k}\log|\mathbf{\Omega}_k| + \text{tr}\big[\mathbf{S}(\mathbf{\Omega}_K\otimes\cdots\otimes\mathbf{\Omega}_1)\big] \\ &\quad + \sum_{k=1}^{K}\lambda_k\sum_{i\neq j}p\big(\Omega_k(i,j)\big) \qquad (3.3) \end{aligned}$$

where $\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}\text{vec}(\mathbf{Y}_n)\text{vec}(\mathbf{Y}_n)^T$. The optimization can now be expressed as

$$\min_{\mathbf{\Omega}_1\succ 0,\cdots,\mathbf{\Omega}_K\succ 0} q(\mathbf{\Omega_1},\cdots,\mathbf{\Omega}_K).$$

Denote its solution by $\{\widehat{\mathbf{\Omega}}_1,\cdots,\widehat{\mathbf{\Omega}}_K\}$.

## 4. Optimization

Block coordinate descent algorithm can be applied to minimize $q(\mathbf{\Omega_1},\cdots,\mathbf{\Omega}_K)$, which

leads to local optimal solutions. For $k = 1, \cdots, K$, we iteratively minimize the objective function with respect to one $\mathbf{\Omega}_k$, while keeping the other matrices $\mathbf{\Omega}_j (j \neq k)$ fixed at current values. As a result of Lemma 1, minimizing (3.3) on a specific $\mathbf{\Omega}_k$ is equivalent to minimizing

$$q_3(\mathbf{\Omega}_k) = -\log|\mathbf{\Omega}_k| + tr[\mathbf{S}_k\mathbf{\Omega}_k] + \lambda_k \cdot \frac{m_k}{m} \sum_{i \neq j} p\big(\Omega_k(i,j)\big) \tag{4.1}$$

with $\mathbf{S}_k = \frac{m_k}{N \cdot m} \sum_{n=1}^{N} \mathbf{V}_{n(k)}[\mathbf{V}_{n(k)}]^T$ and $\mathbf{V}_{n(k)}$ is the $k$-mode matrix unfolding of the tensor

$$\mathbf{V}_n = \mathbf{Y}_n \times \{\mathbf{\Omega}_1^{1/2}, \cdots, \mathbf{\Omega}_{k-1}^{1/2}, \mathbf{I}, \mathbf{\Omega}_{k+1}^{1/2}, \cdots, \mathbf{\Omega}_K^{1/2}\}.$$

The optimization problem (4.1) can be solved by the $glasso$ algorithm of Friedman et al. (2007). Through minimizing on $\mathbf{\Omega}_k$ iteratively, this procedure decreases the objective function after each iteration and eventually converges to a stationary point (Tseng, 2001).

The algorithm is summarized below. Let $\{\mathbf{\Omega}_1^{(s)}, \mathbf{\Omega}_2^{(s)}, \cdots, \mathbf{\Omega}_K^{(s)}\}$ be the current estimate at the beginning of $s$-th iteration.

**Algorithm 1.**

1. $s = 0$, and $\mathbf{\Omega}_k^{(0)} = \mathbf{I}$ for $k = 1, 2, \cdots, K$

2. Repeat

3. s:=s+1

4. For $k = 1, 2, \cdots, K$

5. Compute $\mathbf{V}_n := \mathbf{Y}_n \times \mathbf{\Omega}^{(s)k}$, where $\mathbf{\Omega}^{(s)k}$ is the matrix list

$$\{[\mathbf{\Omega}_1^{(s+1)}]^{1/2}, \cdots [\mathbf{\Omega}_{k-1}^{(s+1)}]^{1/2}, \mathbf{I}, [\mathbf{\Omega}_{k+1}^{(s)}]^{1/2}, \cdots, [\mathbf{\Omega}_K^{(s)}]^{1/2}\}$$

6. Compute $\mathbf{S}_k^{(s)} = \frac{m_k}{N \cdot m} \sum_{n=1}^{N} \mathbf{V}_{n(k)} [\mathbf{V}_{n(k)}]^T$.

7. Update $\mathbf{\Omega}_k^{(s)}$ to $\mathbf{\Omega}_k^{(s+1)}$ by solving objective function (4.1).

8. End For

9. Until Convergence

10. Let $\omega_k = \Omega_k(1,1)$ and $\omega = \prod_{j>1} \omega_j$, then output

$$\{\omega \cdot \mathbf{\Omega}_1^{(s)},\ \mathbf{\Omega}_2^{(s)}/\omega_2, \cdots,\ \mathbf{\Omega}_K^{(s)}/\omega_K\}$$

Although the objective function $q(\mathbf{\Omega_1}, \cdots, \mathbf{\Omega}_K)$ is not convex, we show that as $N \to \infty$, the function is strictly quasi-convex with probability 1. To see this, as $N \to \infty$, the limit of the negative log-likelihood function in $q(\mathbf{\Omega_1}, \cdots, \mathbf{\Omega}_K)$, is

$$\begin{aligned} l(\mathbf{z}) &= -\sum_{k=1}^{K} \frac{m}{m_k} \log |\mathbf{\Omega}_k| + tr\Big((\mathbf{\Sigma}_K^0 \otimes \cdots \otimes \mathbf{\Sigma}_1^0)(\mathbf{\Omega}_K \otimes \cdots \otimes \mathbf{\Omega}_1)\Big) \\ &= -\sum_{k=1}^{K} \frac{m}{m_k} \log |\mathbf{\Omega}_k| + tr\big(\mathbf{\Sigma}_K^0 \mathbf{\Omega}_K\big) \cdots tr\big(\mathbf{\Sigma}_1^0 \mathbf{\Omega}_1\big). \end{aligned}$$

With parameters $\mathbf{z} = (\text{vec}(\mathbf{\Omega}_1)^T, \cdots, \text{vec}(\mathbf{\Omega}_K)^T)^T$, we can find its Hessian matrix $\mathbf{L} = \frac{\partial l(\mathbf{z})}{\partial \mathbf{z} \partial \mathbf{z}^T}$. Next we treat $L$ as a block matrix. For $1 \leq i, j \leq K$, the $(i,j)$-th block matrix of this Hessian matrix is

$$\mathbf{L}_{(i,j)} = \frac{\partial l(\mathbf{z})}{\partial \mathbf{z}_i \partial \mathbf{z}_j^T} = \begin{cases} \big(m/m_i\big) \times \mathbf{\Omega}_i^{-1} \otimes \mathbf{\Omega}_i^{-1}, & i = j \\ \big[\prod_{k \neq i,j} \text{tr}(\mathbf{\Sigma}_k^0 \mathbf{\Omega}_k)\big] \times \text{vec}(\mathbf{\Sigma}_i^0)\text{vec}(\mathbf{\Sigma}_j^0)^T, & i \neq j \end{cases}$$

where $\mathbf{z}_i = \text{vec}(\mathbf{\Omega}_i)$. Except at $\mathbf{z}^0 = (\text{vec}(\mathbf{\Omega}_1^0)^T, \cdots, \text{vec}(\mathbf{\Omega}_K^0)^T)^T$, this Hessian matrix cannot be guaranteed to be nonnegative definite. We can linearly transform this matrix without changing its eigenvalues. Due to the fact that the diagonal blocks of the Hessian matrix at $\mathbf{z}^0$ are positive definite and that

$$\text{vec}(\mathbf{\Omega}_i^0)^T(\mathbf{\Sigma}_i^0 \otimes \mathbf{\Sigma}_i^0)\text{vec}(\mathbf{\Omega}_i^0) = \text{tr}(\mathbf{\Sigma}_i^0 \mathbf{\Omega}_i^0 \mathbf{\Sigma}_i^0 \mathbf{\Omega}_i^0) = m_i$$

the Hessian matrix $\frac{\partial l(\mathbf{z}_0)}{\partial \mathbf{z} \partial \mathbf{z}^T}$ at $\mathbf{z}^0$ can be linearly transformed into a diagonal block matrix $\mathbf{L}' = \text{diag}\{\mathbf{L}'_{(1,1)}, \cdots, \mathbf{L}'_{(K,K)}\}$, and

$$\mathbf{L}'_{(k,k)} = \begin{cases} \big(m/m_1\big)\mathbf{\Sigma}_1^0 \otimes \mathbf{\Sigma}_1^0, & k = 1 \\ \big(m/m_k\big)\mathbf{\Sigma}_k^0 \otimes \mathbf{\Sigma}_k^0 - \big(m/m_k^2\big)\text{vec}(\mathbf{\Sigma}_k^0)\text{vec}(\mathbf{\Sigma}_k^0)^T, & k = 2, \cdots, K \end{cases}$$

Clearly, its first diagonal block $\mathbf{L}'_{(1,1)}$ is positive definite. For $k = 2, 3, \cdots, K$, its first diagonal block $\mathbf{L}'_{(k,k)}$ has eigenvalues with the following properties:

(E1) One equals 0, with eigenvector $\text{vec}(\mathbf{\Omega}_k^0)$;

(E2) The others are positive, with eigenvectors $\mathbf{v}$ satisfying $\text{vec}(\mathbf{\Omega}_k^0)^T\mathbf{v} = 0$.

Property (E1) follows from the fact that

$$\begin{aligned}
&\big(m/m_k\big)\boldsymbol{\Sigma}_k^0 \otimes \boldsymbol{\Sigma}_k^0 \text{vec}(\boldsymbol{\Omega}_k^0) - \big(m/m_k^2\big)\text{vec}(\boldsymbol{\Sigma}_k^0)\text{vec}(\boldsymbol{\Sigma}_k^0)^T\text{vec}(\boldsymbol{\Omega}_k^0) \\
=&\big(m/m_k\big)\text{vec}(\boldsymbol{\Sigma}_k^0\boldsymbol{\Omega}_k^0\boldsymbol{\Sigma}_k^0) - \big(m/m_k^2\big)\text{vec}(\boldsymbol{\Sigma}_k^0)\text{tr}(\boldsymbol{\Sigma}_k^0\boldsymbol{\Omega}_k^0) \\
=&\big(m/m_k\big)\text{vec}(\boldsymbol{\Sigma}_k^0) - \big(m/m_k^2\big)\text{vec}(\boldsymbol{\Sigma}_k^0)\cdot m_k = 0.
\end{aligned}$$

Property (E2) can be justified as follows. Suppose $\mathbf{v} \neq \mathbf{0}$ is an eigenvector of $\mathbf{L}'_{(k,k)}$ $(2 \leq k \leq K)$ satisfying $\text{vec}(\boldsymbol{\Omega}_k^0)^T\mathbf{v} = 0$, and suppose $\nu$ is its eigenvalue, then

$$\big(m/m_k\big)\boldsymbol{\Sigma}_k^0 \otimes \boldsymbol{\Sigma}_k^0\mathbf{v} - \big(m/m_k^2\big)\text{vec}(\boldsymbol{\Sigma}_k^0)\text{vec}(\boldsymbol{\Sigma}_k^0)^T\mathbf{v} = \nu\cdot\mathbf{v}$$

multiplying both sides from the left by $\mathbf{v}^T\boldsymbol{\Omega}_k^0 \otimes \boldsymbol{\Omega}_k^0$ we get

$$\big(m/m_k\big)\mathbf{v}^T\mathbf{v} - \big(m/m_k^2\big)\mathbf{v}^T\boldsymbol{\Omega}_k^0 \otimes \boldsymbol{\Omega}_k^0\text{vec}(\boldsymbol{\Sigma}_k^0)\text{vec}(\boldsymbol{\Sigma}_k^0)^T\mathbf{v} = \nu\cdot\mathbf{v}^T\boldsymbol{\Omega}_k^0 \otimes \boldsymbol{\Omega}_k^0\mathbf{v}$$

that is

$$\big(m/m_k\big)\mathbf{v}^T\mathbf{v} - \big(m/m_k^2\big)\mathbf{v}^T\text{vec}(\boldsymbol{\Omega}_k^0)\text{vec}(\boldsymbol{\Sigma}_k^0)^T\mathbf{v} = \nu\cdot\mathbf{v}^T\boldsymbol{\Omega}_k^0 \otimes \boldsymbol{\Omega}_k^0\mathbf{v}$$

Because of the fact that $\text{vec}(\boldsymbol{\Omega}_k^0)^T\mathbf{v} = 0$ and that $\boldsymbol{\Omega}_k^0 \otimes \boldsymbol{\Omega}_k^0$ is positive definite, we have

$$\nu = \frac{m}{m_k} \times \frac{\mathbf{v}^T\mathbf{v}}{\mathbf{v}^T\boldsymbol{\Omega}_k^0 \otimes \boldsymbol{\Omega}_k^0\mathbf{v}} > 0$$

Thus, (E2) is established.

From this, we know that $\frac{\partial l(\mathbf{z}_0)}{\partial\mathbf{z}\partial\mathbf{z}^T}$ is non-negative definite. As a result, the negative likelihood function is a convex function although not strictly convex. Since the Lasso the penalty function is strictly quasi-convex, we have the following lemma.

**Lemma 2.** *As $N \to \infty$, the limit of the objective function (3.3) with parameters $\{\mathbf{\Omega}_1, \mathbf{\Omega}_2, \cdots, \mathbf{\Omega}_K\}$ is strictly quasi-convex with probability one at global optimizer $\{\mathbf{\Sigma}_1^0, \cdots, \mathbf{\Sigma}_K^0\}$.*

## 5. Asymptotic Results

This section discusses the asymptotic behavior for the optimizer of (3.3). Theorem 1 and Theorem 2 assume that the dimensions $(m_1, m_2, \cdots, m_K)$ are fixed, while Theorem 3 and Theorem 4 allow the dimensions $(m_1, m_2, \cdots, m_K)$ to diverge with sample size $N$. For both scenarios, a fast rate of convergence can be guaranteed and the true sparsity pattern of each concentration matrix can be recovered by using the adaptive Lasso penalty with probability tending to 1.

For multi-way tensor normal distribution, the effective sample size for estimating $\mathbf{\Omega}_k^0$ is asymptotically $m/m_k \cdot N$, which is larger than $N$. In fact, if $\mathbf{\Omega}_l^0 (l \neq k)$'s are known, the correlation on the $l(\neq k)$-th mode can be removed, the columns of the $k$-mode matrix unfolding can be treated as the i.i.d. samples from the corresponding vector normal distribution, and these column vectors can be pooled together to estimate $\mathbf{\Omega}_k^0$. This can be stated precisely in the following lemma. It helps to explain the fast convergence rate in Theorem 2 and Theorem 3 and is used in the proofs.

**Lemma 3.** *Let $\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_N$ be $N$ i.i.d. observations from tensor normal distribution* anorm$(\mathbf{0}, \mathbf{\Sigma}_1^0 \circ \mathbf{\Sigma}_2^0 \circ \cdots \circ \mathbf{\Sigma}_K^0)$*, and suppose $\{\mathbf{\Sigma}_1^0, \cdots, \mathbf{\Sigma}_{k-1}^0, \mathbf{\Sigma}_{k+1}^0, \cdots, \mathbf{\Sigma}_K^0\}$ are known and $\mathbf{\Sigma}_k^0$ is unknown, then the columns $\mathbf{v}_n^{0k}(j)$ of*

$$\mathbf{V}_n^{0k} = \mathbf{Y}_{n(k)} \Big[ \left(\mathbf{\Omega}_K^0\right)^{1/2} \otimes \cdots \left(\mathbf{\Omega}_{k+1}^0\right)^{1/2} \otimes \left(\mathbf{\Omega}_{k-1}^0\right)^{1/2} \otimes \cdots \otimes \left(\mathbf{\Omega}_1^0\right)^{1/2} \Big]$$

*are i.i.d samples from $m_k$-vector normal distribution $N(\mathbf{0}, \mathbf{\Sigma}_k^0)$, and*

$$\mathbf{S}_k = \frac{m_k}{N \cdot m} \sum_{n=1}^{N} \mathbf{V}_n^{0k} \Big[\mathbf{V}_n^{0k}\Big]^T = \frac{m_k}{N \cdot m} \sum_{n=1}^{N} \sum_{j=1}^{m/m_k} \mathbf{v}_n^{0k}(j) \big[\mathbf{v}_n^{0k}(j)\big]^T$$

*estimates $\mathbf{\Sigma}_k^0$ with sample size $(m \cdot N)/m_k$.*

Theorem 1 shows the consistency of estimators from (3.3) with Lasso penalty when the dimensions $(m_1, m_2, \cdots, m_K)$ are fixed. The tuning parameters may change with sample size $N$, but we omit the subscript $N$ for simplicity.

**Theorem 1.** ***(Consistency)*** *For $k = 1, 2, \cdots, K$, assume $\sqrt{N}\lambda_k \to \lambda_{0k}$ for some constants $\lambda_{0k} \geq 0$, and $\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_N$ are $N$ i.i.d. observations from tensor normal distribution* anorm$(\mathbf{0}, \mathbf{\Sigma}_1^0 \circ \mathbf{\Sigma}_2^0 \circ \cdots \circ \mathbf{\Sigma}_K^0)$*, then there exists local optimizer $\{\widehat{\mathbf{\Omega}}_1, \cdots, \widehat{\mathbf{\Omega}}_K\}$ of (3.3) with the $\ell_1$ norm penalty such that:*

$$\sqrt{N}\{(\widehat{\mathbf{\Omega}}_1, \cdots, \widehat{\mathbf{\Omega}}_K) - (\mathbf{\Omega}_1^0, \cdots, \mathbf{\Omega}_K^0)\} \to_d \text{argmin}_{(\mathbf{U}_1, \cdots, \mathbf{U}_K)}\, g(\mathbf{U}_1, \cdots, \mathbf{U}_K)$$

*where*

$$\begin{aligned} g(\mathbf{U}_1, \cdots, \mathbf{U}_K) =& \frac{1}{2} \sum_{k=1}^{K} \frac{m}{m_k} \text{tr}(\mathbf{U}_k \mathbf{\Sigma}_k^0 \mathbf{U}_k \mathbf{\Sigma}_k^0) + \sum_{i<j} \frac{m}{m_i m_j} \text{tr}(\mathbf{U}_i \mathbf{\Sigma}_i^0) \text{tr}(\mathbf{U}_j \mathbf{\Sigma}_j^0) \\ &+ \sigma \cdot W + \sum_{k=1}^{K} \lambda_{0k} \Big( U_k(i,j) \text{sign}(\mathbf{\Omega}_k(i,j)) I\{\mathbf{\Omega}_k(i,j) \neq 0\} \\ &\qquad + |U_k(i,j)| I\{\mathbf{\Omega}_k(i,j) = 0\} \Big), \end{aligned}$$

*W is subject to standard normal distribution* $\mathrm{N}(0,1)$ *and*

$$\sigma^2 = \sum_{k=1}^{K} \frac{2m}{m_k} tr(\mathbf{U}_k \mathbf{\Sigma}_k^0 \mathbf{U}_k \mathbf{\Sigma}_k^0) + \sum_{i \neq j} \frac{2m}{m_i m_j} tr(\mathbf{U}_i \mathbf{\Sigma}_i^0) tr(\mathbf{U}_j \mathbf{\Sigma}_j^0).$$

With a slight modification of the proof of Theorem 1, we can show that the consistency also holds for the solutions of (3.3) with adaptive Lasso penalty. The adaptive penalty is introduced for selecting the non-zero entries in the concentration matrix and achieving optimal efficiency for them. For $k = 1, 2, \cdots, K$, define the active sets $\mathcal{A}_k = \{(i,j) : \Omega_k^0(i,j) \neq 0\}$ as the set of indices corresponding to non-zero entries in $\mathbf{\Omega}_k^0$.

**Theorem 2.** ***(Oracle Property)*** *Consider (3.3) with adaptive Lasso penalty, and let* $\gamma > 0$ *be a constant and* $\widetilde{\mathbf{\Omega}}_k$ *be* $N^{1/2}$*-consistent estimators. When* $\sqrt{N}\lambda_k \to 0$ *and* $N^{(\gamma+1)/2}\lambda_k \to \infty$ *for* $k = 1, 2, \cdots, K$*, there exists local solutions of (3.3) satisfying the oracle property:*

*(1) For* $k = 1, 2, \cdots, K$ *and all* $(i,j) \in \mathscr{A}_k^c$, $\widehat{\Omega}_k(i,j) = 0$ *with probability tending to 1.*

*(2)For* $k = 1, 2, \cdots, K$ *and elements indexed by* $(i,j) \in \mathscr{A}_k$*,*

$$vec(\widehat{\mathbf{\Omega}}_k - \mathbf{\Omega}_k^0)_{\mathscr{A}_k} \to_d N\Big(\mathbf{0}, \frac{m_k}{m}\Big[\big(\mathbf{\Omega}_k^0 \otimes \mathbf{\Omega}_k^0\big)_{(\mathscr{A}_k,\cdot)}\Big]\big(\mathbf{\Sigma}_k^0 \otimes \mathbf{\Sigma}_k^0\big)\Big[\big(\mathbf{\Omega}_k^0 \otimes \mathbf{\Omega}_k^0\big)_{(\mathscr{A}_k,\cdot)}\Big]^T\Big)$$

*where* $\mathrm{vec}(\widehat{\mathbf{\Omega}}_k - \mathbf{\Omega}_k^0)_{\mathscr{A}_k}$ *is a sub-vector of* $\mathrm{vec}(\widehat{\mathbf{\Omega}}_k - \mathbf{\Omega}_k^0)$ *with only elements indexed by* $\mathscr{A}_k$ *reserved; and* $\big(\mathbf{\Omega}_k^0 \otimes \mathbf{\Omega}_k^0\big)_{(\mathscr{A}_k,\cdot)}$ *is a sub-matrix of* $\mathbf{\Omega}_k^0 \otimes \mathbf{\Omega}_k^0$ *with all rows corresponding to* $\mathscr{A}_k^c$ *removed. That is, the* $l$*-th row of* $\mathbf{\Omega}_k^0 \otimes \mathbf{\Omega}_k^0$ *can be reserved if and only if* $l = m_k(j-1)+i$ *for some* $(i,j) \in \mathscr{A}_k$*.*

For tensor normal distribution, the estimator of precision matrices converges much faster than the vector normal case ($K = 1$). For the tensor case, the limiting covariance

matrix for active entry estimator is

$$\frac{m_k}{m}\left[\left(\mathbf{\Omega}_k^0 \otimes \mathbf{\Omega}_k^0\right)_{(\mathscr{A}_k,\cdot)}\right]\left(\mathbf{\Sigma}_k^0 \otimes \mathbf{\Sigma}_k^0\right)\left[\left(\mathbf{\Omega}_k^0 \otimes \mathbf{\Omega}_k^0\right)_{(\mathscr{A}_k,\cdot)}\right]^T$$

while for vector normal distribution($K = 1$), the limiting covariance matrix is

$$\left[\left(\mathbf{\Omega}_k^0 \otimes \mathbf{\Omega}_k^0\right)_{(\mathscr{A}_k,\cdot)}\right]\left(\mathbf{\Sigma}_k^0 \otimes \mathbf{\Sigma}_k^0\right)\left[\left(\mathbf{\Omega}_k^0 \otimes \mathbf{\Omega}_k^0\right)_{(\mathscr{A}_k,\cdot)}\right]^T.$$

In the former case, the additional factor $m_k/m$ can be quite small when $\prod_{i\neq k} m_i$ is large. This explains the fast rate of convergence in our simulation studies.

This fast rate of convergence is also observed when the dimensions $(m_1, m_2, \cdots, m_K)$ increase with the sample size $N$. Results similar to Lam and Fan (2009) hold, and with much faster rates, as shown in Theorem 3. Again, the results are stated and proven for the $\ell_1$ norm penalty, and similar results hold for the adaptive Lasso penalty. Let $s_k = |\mathscr{A}_k| - m_k$ be the number of non-zero off-diagonal entries in $\Omega_k$, which also varies with sample size $N$. Under some conditions, the convergence in terms of Frobenius norm can be guaranteed for (3.3).

**Theorem 3.** ***(Rate of Convergence)*** *Assume* $\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_N$ *are* $N$ *i.i.d. observations from a tensor normal distribution* $\text{anorm}(\mathbf{0}, \mathbf{\Sigma}_1^0 \circ \mathbf{\Sigma}_2^0 \circ \cdots \circ \mathbf{\Sigma}_K^0)$*, with dimensions* $(m_1, m_2, \cdots, m_K)$ *diverging when sample size N goes to infinity.*

*In addition, for* $k = 1, 2, \cdots, K$ *and some constants* $\tau_{k1}, \tau_{k2}$*, assume the eigenvalues are bounded,*

$$0 < \tau_{k1} < \lambda_{min}(\mathbf{\Sigma}_k^0) \leq \lambda_{max}(\mathbf{\Sigma}_k^0) < \tau_{k2} < \infty \tag{5.1}$$

*If the following conditions on tuning parameters $\lambda_k$'s ($k = 1, \cdots, K$) are satisfied*

$$\frac{m \log m_k}{m_k N} = O(\lambda_k^2) \quad \textit{and} \quad \lambda_k^2 = O\Big((1 + \frac{m_k}{s_k + 1})\frac{m \log m_k}{m_k N}\Big) \tag{5.2}$$

*then when the Lasso penalty functions are used, there exists a local minimizer $(\widehat{\mathbf{\Omega}}_1, \widehat{\mathbf{\Omega}}_2, \cdots, \widehat{\mathbf{\Omega}}_K)$ of (3.3) such that*

$$\|\widehat{\mathbf{\Omega}}_k - \mathbf{\Omega}_k^0\|_F^2 = O_p\Big(m_k(m_k + s_k) \log m_k/(Nm)\Big).$$

Because $\|\mathbf{A}\| \leq \|\mathbf{A}\|_F$ for any matrix $\mathbf{A}$, this rate of convergence also holds for the spectral norm. The rate of convergence for tensor normal distribution is $(m_k/m)(m_k + s_k) \log m_k/N$, which is much faster than multivariate normal case where $K = 1$. The rate in the latter case is $(m_1 + s_1) \log m_1/N$, as shown in Lam and Fan (2009). Clearly, the results also hold for the adaptive Lasso penalty. Furthermore, with adaptive Lasso penalty, we can also recover the true sparsity pattern with probability tending to one, as shown in the following theorem.

**Theorem 4.** ***(Sparsistency)*** *Given the conditions in Theorem 3, for $k = 1, \cdots, K$, suppose $\widetilde{\mathbf{\Omega}}_k$ is $f_k$-consistent estimator for $\mathbf{\Omega}_k^0$ in the sense that*

$$f_k \|\widetilde{\mathbf{\Omega}}_k - \mathbf{\Omega}_k^0\|_\infty = O_p(1)$$

*If $\{\widehat{\mathbf{\Omega}}_1, \widehat{\mathbf{\Omega}}_2, \cdots, \widehat{\mathbf{\Omega}}_K\}$ is a local minimizer of (3.3) with adaptive Lasso penalty satisfying*

1. $\|\widehat{\mathbf{\Omega}}_k - \mathbf{\Omega}_k^0\|_F^2 = O_p\{m_k(m_k + s_k)\log m_k/(Nm)\}$*; and*

2. $\|\widehat{\mathbf{\Omega}}_k - \mathbf{\Omega}_k^0\|^2 = O_p(\eta_n)$ *for a sequence* $\eta_n \to 0$

*and if the tuning parameters satisfy*

$$f_k^{-2\gamma}\frac{m^2}{m_k^2}\Big(\frac{m_k \log m_k}{mN} + \eta_n + \sum_{l\neq k}\frac{\tau_{l,2}^2}{mN}(m_l + s_l)\log m_l\Big) = O(\lambda_k^2)$$

*then with probability tending to one, we have* $\widehat{\Omega}_k(i,j) = 0$ *for all* $(i,j) \in \mathscr{A}_k^c$ *and* $k = 1, 2, \cdots, K$.

Similar to Yin and Li (2012), the sparsistency results require condition (5.2) to impose both a lower and upper bound on the rates of the regularization parameters $\lambda_k$'s in order to control the model sparsity and estimation biases.

## 6. Monte Carlo Simulation Studies

### *6.1 Comparison candidates and measurements*

We evaluate the performance of penalized likelihood method for tensor normal data and compare this to two naive methods using simulations.

The first naive method is an approximate maximum likelihood estimation, which is the MLE without penalty when the effective sample size is larger than the dimensions $m_k$'s and the $\ell_1$ penalized estimate otherwise. Statistical tests are used to select edges when the effective sample size is large. Specifically, for $k = 1, 2, \cdots, K$, the effective sample size for estimating of $\mathbf{\Omega}_k^0$ is approximately $N_k = Nm/m_k$, where $N$ is the real sample size. In Algorithm 1 of Section 4, if $N_k > m_k$, the inverse of $\mathbf{S}_k$ is directly used to update the estimation of $\mathbf{\Omega}_k$ in the Step 7, which corresponds to the MLE procedure. However, when

$N_k <= m_k$, we update the estimation of $\mathbf{\Omega}_k$ through (4.1) with an $\ell_1$ Lasso penalty. When $N_k > m_k$, hypothesis tests are also performed to select edges after estimation. Let $\rho_{ij}$ denote the partial correlation between $X_i$ and $X_j$ adjusting for the remaining elements and $\hat{\rho}_{ij}$ denote its MLE estimator, then

$$\frac{1}{2}\log\Big[\frac{1+\hat{\rho}_{ij}}{1-\hat{\rho}_{ij}}\Big] \to \mathbf{N}(0, 1/(n-p-5)).$$

Based on this result, for $k = 1, 2, \cdots, N$ and $i < j$, let

$$\hat{\rho}_{ij}^k = -\frac{\widehat{\Omega}_k(i,j)}{\sqrt{\widehat{\Omega}_k(i,i)\widehat{\Omega}_k(j,j)}}$$

and we set $\widehat{\Omega}_k(i,j) = \widehat{\Omega}_k(j,i) = 0$ whenever $N_k > m_k$ and

$$\Big|\frac{1}{2}\log\Big[\frac{1+\hat{\rho}_{ij}^k}{1-\hat{\rho}_{ij}^k}\Big]\Big| < \frac{z_{\alpha/2}}{\sqrt{N_k - m_k - 5}}$$

where $z_\beta$ is the upper $\beta \times 100\%$ quantile of standard normal distribution. We choose $\alpha = 0.1$.

The second naive method estimates each $\mathbf{\Omega}_k$ separately with the adaptive Lasso penalty. It treats the other modes as independent, i.e., assuming $\mathbf{\Omega}_j = \mathbf{I}_j$ $(j \neq k)$ in the estimation procedure. In this case, the Step 5 of Algorithm 1 in Section 4 is not used and $\mathbf{S}_k$ in Step 6 is computed as

$$\mathbf{S}_k = \frac{m_k}{N \cdot m}\sum_{n=1}^{N} Y_{n(k)}[Y_{n(k)}]^T.$$

For the penalized maximum likelihood method, we use the adaptive Lasso penalty with the approximate MLE as the initial estimator $\widetilde{\mathbf{\Omega}}_k$. The accuracy of the estimated concentration matrix is measured by various matrix norms of $\mathbf{\Delta}_k = \mathbf{\Omega}_k^0 - \widehat{\mathbf{\Omega}}_k$, where $\mathbf{\Omega}_k^0$ is the true

matrix and $\widehat{\mathbf{\Omega}}_k$ is the estimated matrix. We use the following norms: $\|\cdot\|_F$ the Frobenius norm, $\|\cdot\|_p$ the operator norm, and $|||\cdot|||_\infty$ the entry-wise max norm. In addition, the accuracy of discovering the Gaussian graph structure is also measured. Let TP, TN, FP and FN be the numbers of true positives, true negatives, false positives and false negatives, respectively, where the true positives are the true links on the tenor normal graphs. We define specificity (SPE), sensitivity (SEN), and Matthew's correlation coefficient (MCC) as

$$\text{SPE} = \frac{\text{TN}}{\text{TN}+\text{FP}}, \qquad \text{SEN} = \frac{\text{TP}}{\text{TP}+\text{FN}}$$
$$\text{MCC} = \frac{\text{TP}\times\text{TN}-\text{FP}\times\text{FN}}{\big\{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})\big\}^{1/2}}.$$

*Models and data generation*

The sparse precision matrix $\mathbf{\Omega}_k^0$'s are generated as follows. Non-zero off-diagonal elements for the upper triangle of $\mathbf{\Omega}_k^0$ are selected independently with probability $p_k$. For non-zero elements, their values are generated from

$$\Omega_k^0(i,j) = \Omega_k^0(j,i) \sim \text{Uniform}\big([-0.8,-0.2]\cup[0.2,0.8]\big).$$

We then make $\mathbf{\Omega}_k^0$ diagonally dominant by dividing $i$-th row by $1.2\times\sum_{j\neq i}\big|\Omega_k^0(i,j)\big|$ for $i=1,2,\cdots,m_k$ and then setting all diagonal entries to be 1. We then symmetrize the matrix by letting $\mathbf{\Omega}_k^0 := \big[\mathbf{\Omega}_k^0 + (\mathbf{\Omega}_k^0)^T\big]/2$.

The following four models are considered with sample size $N=10$. These models have different dimensions and different degrees of sparsity as indicated by $p_k$. The simulations are repeated $50$ times.

1. Model 1: three-way tensor data with dimensions $(30,30,30)$ and sparsity $p_1=p_2=$

$p_3 = 0.1$

2. Model 2: three-way tensor data with dimensions $(6, 6, 500)$ and sparsity $p_1 = 0.3,\ p_2 = 0.2,\ p_3 = 0.005$

3. Model 3: four-way tensor data with dimensions $(30, 30, 30, 30)$ and sparsity $p_1 = 0.05,\ p_2 = 0.075,\ p_3 = 0.1,\ p_4 = 0.2$

4. Model 4: four-way tensor data with dimensions $(30, 40, 50, 100)$ and sparsity $p_1 = 0.2,\ p_2 = 0.125,\ p_3 = 0.1,\ p_4 = 0.075$

*Simulation results*

For all simulations, the tuning parameters are all chosen based on a validation set of sample size of 10. The results are presented in Tables 1-4. In almost all scenarios, the dimensions of the models are larger than the real sample size $N = 10$. However, we observed that the estimates of the precision matrices are still very accurate. This can be explained by the effective sample size, which is very large for each dimension of the tensor data.

For all four models considered, the proposed penalized likelihood procedure results in better estimation of the precision matrices than the other two naive methods in terms of estimation error. In terms of model selection, the penalized likelihood estimation also gives better results, although the performance of the naive method that assumes independency is comparable in certain circumstances. The effect of the effective sample sizes on precision matrix estimation is also clearly demonstrated in these tables. For Model 1, the effective sample size is $10 \times 30 \times 30 = 9000$ for each way of the tensor data. For Model 3, however, the effective sample size for each way of the data is $10 \times 30^3 = 270000$, which is 30 times larger than Model 1. It is clear from Tables 1 and 3 that the estimates for Model 3 are more accurate than these for Model 1. For Model 2, the effective sample size for estimating $\mathbf{\Omega}_3^0$

Table 1: Model 1: three-way tensor data with dimensions $(30, 30, 30)$, sample size 10 and sparsity $p_1 = p_2 = p_3 = 0.1$. For each measurement, mean and standard error over 50 replications are shown. P-MLE: penalized maximum likelihood estimates; A-MLE: approximate maximum likelihood estimates; I-MLE: penalized maximum likelihood estimates under independency assumption. $\Delta_k$ is the difference between the true and estimated precision matrix for $k = 1, 2, 3$.

| | | P-MLE | A-MLE | I-MLE |
|---|---|---|---|---|
| $\mathbf{\Omega}_1$ | $\|\Delta_1\|_F$ | 0.15(0.034) | 0.26(0.023) | 0.23(0.061) |
| | $\|\Delta_1\|_\infty$ | 0.09(0.018) | 0.20(0.036) | 0.14(0.035) |
| | $\|\Delta_1\|_2$ | 0.06(0.013) | 0.11(0.017) | 0.10(0.025) |
| | $\|\|\|\Delta_1\|\|\|_\infty$ | 0.04(0.010) | 0.05(0.010) | 0.07(0.018) |
| | SPE | 0.95(0.010) | 0.90(0.014) | 0.98(0.007) |
| | SEN | 1.00(0.000) | 1.00(0.000) | 1.00(0.000) |
| | MCC | 0.79(0.028) | 0.67(0.029) | 0.92(0.030) |
| $\mathbf{\Omega}_2$ | $\|\Delta_2\|_F$ | 0.15(0.051) | 0.25(0.037) | 0.27(0.068) |
| | $\|\Delta_2\|_\infty$ | 0.10(0.034) | 0.20(0.036) | 0.18(0.043) |
| | $\|\Delta_2\|_2$ | 0.07(0.022) | 0.11(0.020) | 0.12(0.030) |
| | $\|\|\|\Delta_2\|\|\|_\infty$ | 0.04(0.011) | 0.05(0.011) | 0.07(0.021) |
| | SPE | 0.99(0.002) | 0.90(0.016) | 0.96(0.009) |
| | SEN | 1.00(0.000) | 1.00(0.000) | 1.00(0.000) |
| | MCC | 0.99(0.007) | 0.70(0.031) | 0.87(0.031) |
| $\mathbf{\Omega}_3$ | $\|\Delta_3\|_F$ | 0.18(0.052) | 0.27(0.049) | 0.28(0.067) |
| | $\|\Delta_3\|_\infty$ | 0.11(0.027) | 0.20(0.042) | 0.18(0.035) |
| | $\|\Delta_3\|_2$ | 0.07(0.020) | 0.11(0.026) | 0.12(0.027) |
| | $\|\|\|\Delta_3\|\|\|_\infty$ | 0.05(0.013) | 0.05(0.015) | 0.08(0.022) |
| | SPE | 1.00(0.002) | 0.90(0.016) | 0.96(0.010) |
| | SEN | 1.00(0.000) | 1.00(0.000) | 1.00(0.000) |
| | MCC | 1.00(0.008) | 0.71(0.033) | 0.86(0.032) |

is $6 \times 6 \times 10 = 360$, which is smaller than its dimension of $500$, which leads to larger estimation errors.

# 7 Real Data Analysis

Omberg et. al (2009) considered the expression levels of 4270 genes of *Saccharomyces cerevisiae* during a time course of cell cycle under two different experimental conditions.

Table 2: Model 2: three-way tensor data with dimensions $(6, 6, 500)$, sample size 10 and sparsity $p_1 = 0.3,\ p_2 = 0.2,\ p_3 = 0.005$. For each measurement, mean and standard error over 50 replications are shown. P-MLE: penalized maximum likelihood estimates; A-MLE: approximate maximum likelihood estimates; I-MLE: penalized maximum likelihood estimates under independency assumption. $\Delta_k$ is the difference between the true and estimated precision matrix for $k = 1, 2, 3$.

| | | P-MLE | A-MLE | I-MLE |
|---|---|---|---|---|
| $\mathbf{\Omega}_1$ | $\|\Delta_1\|_F$ | 0.03(0.012) | 0.04(0.010) | 0.05(0.018) |
| | $\|\Delta_1\|_\infty$ | 0.04(0.014) | 0.04(0.012) | 0.05(0.021) |
| | $\|\Delta_1\|_2$ | 0.03(0.011) | 0.03(0.010) | 0.04(0.017) |
| | $\|\|\|\Delta_1\|\|\|_\infty$ | 0.02(0.007) | 0.02(0.006) | 0.03(0.010) |
| | SPE | 0.99(0.034) | 0.91(0.115) | 0.99(0.030) |
| | SEN | 1.00(0.000) | 1.00(0.000) | 1.00(0.000) |
| | MCC | 0.99(0.034) | 0.91(0.106) | 0.99(0.030) |
| $\mathbf{\Omega}_2$ | $\|\Delta_2\|_F$ | 0.03(0.008) | 0.03(0.011) | 0.03(0.012) |
| | $\|\Delta_2\|_\infty$ | 0.02(0.009) | 0.03(0.015) | 0.03(0.013) |
| | $\|\Delta_2\|_2$ | 0.02(0.008) | 0.03(0.009) | 0.03(0.011) |
| | $\|\|\|\Delta_2\|\|\|_\infty$ | 0.02(0.005) | 0.02(0.005) | 0.02(0.008) |
| | SPE | 1.00(0.000) | 0.92(0.082) | 0.99(0.013) |
| | SEN | 1.00(0.000) | 1.00(0.000) | 1.00(0.000) |
| | MCC | 1.00(0.000) | 0.88(0.114) | 0.99(0.021) |
| $\mathbf{\Omega}_3$ | $\|\Delta_3\|_F$ | 3.64(0.070) | 4.82(0.114) | 7.11(0.497) |
| | $\|\Delta_3\|_\infty$ | 0.95(0.100) | 1.36(0.130) | 1.82(0.158) |
| | $\|\Delta_3\|_2$ | 0.46(0.023) | 0.55(0.022) | 0.80(0.063) |
| | $\|\|\|\Delta_3\|\|\|_\infty$ | 0.27(0.0378) | 0.29(0.026) | 0.44(0.074) |
| | SPE | 1.00(0.000) | 0.98(0.001) | 0.99(0.001) |
| | SEN | 0.84(0.015) | 0.92(0.010) | 0.58(0.023) |
| | MCC | 0.63(0.010) | 0.36(0.006) | 0.37(0.019) |

Each time course was measured at 12 time points with cell cycles synchronized by $\alpha$-factor pheromone. Under the depleted condition of Cdc6 or Cdc45 (Cdc6-/Cdc45-), the DNA replication initialization is prevented without delaying cell cycle progression. The gene expressions were also measured in the presence of Cdc6 or Cdc45 (Cdc6+/Cdc45+-) without preventing DNA replication. In our analysis, the 4720 genes are averaged on observed values of different probes. After averaging and removing the genes with missing values, a total of 404 genes are used in our analysis. Among these genes, 141, 97, 62, 37 and 67 genes are regulated during the G1, G2/M, M/G1, S and S/G2 phase, respectively

Table 3: Model 3: four-way tensor data with dimensions $(30, 30, 30, 30)$ and sample size 10, $p_1 = 0.05,\ p_2 = 0.075,\ p_3 = 0.1,\ p_4 = 0.2$. For each measurement, mean and standard error over 50 replications are shown. P-MLE: penalized maximum likelihood estimates; A-MLE: approximate maximum likelihood estimates; I-MLE: penalized maximum likelihood estimates under independency assumption. $\Delta_k$ is the difference between the true and estimated precision matrix for $k = 1, 2, 3, 4$.

| | | P-MLE | A-MLE | I-MLE |
|---|---|---|---|---|
| $\mathbf{\Omega}_1$ | $\|\Delta_1\|_F$ | 0.02(0.006) | 0.04(0.004) | 0.04(0.009) |
| | $\|\Delta_1\|_\infty$ | 0.01(0.003) | 0.03(0.005) | 0.02(0.005) |
| | $\|\Delta_1\|_2$ | 0.01(0.003) | 0.02(0.002) | 0.02(0.005) |
| | $\|\|\|\Delta_1\|\|\|_\infty$ | 0.01(0.002) | 0.01(0.002) | 0.01(0.003) |
| | SPE | 1.00(0.001) | 0.90(0.017) | 1.00(0.001) |
| | SEN | 1.00(0.000) | 1.00(0.000) | 1.00(0.000) |
| | MCC | 1.00(0.010) | 0.50(0.035) | 1.00(0.007) |
| $\mathbf{\Omega}_2$ | $\|\Delta_2\|_F$ | 0.02(0.005) | 0.04(0.005) | 0.05(0.012) |
| | $\|\Delta_2\|_\infty$ | 0.01(0.004) | 0.03(0.007) | 0.03(0.007) |
| | $\|\Delta_2\|_2$ | 0.01(0.003) | 0.02(0.003) | 0.02(0.006) |
| | $\|\|\|\Delta_2\|\|\|_\infty$ | 0.01(0.002) | 0.01(0.002) | 0.01(0.004) |
| | SPE | 1.00(0.00) | 0.90(0.016) | 1.00(0.002) |
| | SEN | 1.00(0.00) | 1.00(0.000) | 1.00(0.000) |
| | MCC | 1.00(0.00) | 0.63(0.033) | 0.99(0.012) |
| $\mathbf{\Omega}_3$ | $\|\Delta_3\|_F$ | 0.03(0.004) | 0.04(0.004) | 0.05(0.010) |
| | $\|\Delta_3\|_\infty$ | 0.02(0.003) | 0.04(0.006) | 0.03(0.007) |
| | $\|\Delta_3\|_2$ | 0.01(0.002) | 0.02(0.003) | 0.02(0.004) |
| | $\|\|\|\Delta_3\|\|\|_\infty$ | 0.01(0.002) | 0.01(0.002) | 0.02(0.005) |
| | SPE | 1.00(0.001) | 0.90(0.016) | 1.00(0.002) |
| | SEN | 1.00(0.000) | 1.00(0.000) | 1.00(0.000) |
| | MCC | 1.00(0.002) | 0.72(0.032) | 0.99(0.008) |
| $\mathbf{\Omega}_4$ | $\|\Delta_4\|_F$ | 0.03(0.010) | 0.05(0.007) | 0.07(0.010) |
| | $\|\Delta_4\|_\infty$ | 0.02(0.006) | 0.04(0.006) | 0.05(0.008) |
| | $\|\Delta_4\|_2$ | 0.01(0.004) | 0.02(0.003) | 0.03(0.005) |
| | $\|\|\|\Delta_4\|\|\|_\infty$ | 0.01(0.003) | 0.01(0.002) | 0.02(0.004) |
| | SPE | 1.00(0.001) | 0.90(0.020) | 1.00(0.001) |
| | SEN | 1.00(0.000) | 1.00(0.000) | 1.00(0.000) |
| | MCC | 1.00(0.003) | 0.79(0.033) | 1.00(0.002) |

Table 4: Model 4: four-way tensor data with dimensions $(30, 40, 50, 100)$ and sample size 10, $p_1 = 0.2$, $p_2 = 0.125$, $p_3 = 0.1$, $p_4 = 0.075$. For each measurement, mean and standard error over 50 replications are shown. P-MLE: penalized maximum likelihood estimates; A-MLE: approximate maximum likelihood estimates; I-MLE: penalized maximum likelihood estimates under independency assumption. $\Delta_k$ is the difference between the true and estimated precision matrix for $k = 1, 2, 3, 4$.

| | | P-MLE | A-MLE | I-MLE |
|---|---|---|---|---|
| $\mathbf{\Omega}_1$ | $\|\Delta_1\|_F$ | 0.01(0.002) | 0.02(0.001) | 0.01(0.002) |
| | $\|\Delta_1\|_\infty$ | 0.01(0.001) | 0.01(0.002) | 0.01(0.002) |
| | $\|\Delta_1\|_2$ | 0.00(0.001) | 0.01(0.001) | 0.01(0.001) |
| | $\|\|\|\Delta_1\|\|\|_\infty$ | 0.00(0.001) | 0.00(0.001) | 0.00(0.001) |
| | SPE | 1.00(0.000) | 0.90(0.018) | 1.00(0.001) |
| | SEN | 1.00(0.000) | 1.00(0.000) | 1.00(0.000) |
| | MCC | 1.00(0.000) | 0.80(0.030) | 1.00(0.003) |
| $\mathbf{\Omega}_2$ | $\|\Delta_2\|_F$ | 0.01(0.002) | 0.03(0.002) | 0.02(0.004) |
| | $\|\Delta_2\|_\infty$ | 0.01(0.001) | 0.02(0.002) | 0.01(0.002) |
| | $\|\Delta_2\|_2$ | 0.01(0.001) | 0.01(0.001) | 0.01(0.002) |
| | $\|\|\|\Delta_2\|\|\|_\infty$ | 0.00(0.001) | 0.00(0.001) | 0.00(0.001) |
| | SPE | 1.00(0.000) | 0.90(0.010) | 1.00(0.002) |
| | SEN | 1.00(0.000) | 1.00(0.000) | 1.00(0.000) |
| | MCC | 1.00(0.000) | 0.73(0.018) | 0.99(0.007) |
| $\mathbf{\Omega}_3$ | $\|\Delta_3\|_F$ | 0.02(0.004) | 0.03(0.002) | 0.02(0.004) |
| | $\|\Delta_3\|_\infty$ | 0.01(0.002) | 0.02(0.003) | 0.01(0.002) |
| | $\|\Delta_3\|_2$ | 0.01(0.001) | 0.01(0.001) | 0.01(0.001) |
| | $\|\|\|\Delta_3\|\|\|_\infty$ | 0.00(0.001) | 0.00(0.001) | 0.01(0.001) |
| | SPE | 1.00(0.000) | 0.90(0.010) | 0.99(0.002) |
| | SEN | 1.00(0.000) | 1.00(0.000) | 1.00(0.000) |
| | MCC | 1.00(0.000) | 0.69(0.019) | 0.97(0.011) |
| $\mathbf{\Omega}_4$ | $\|\Delta_4\|_F$ | 0.04(0.007) | 0.09(0.004) | 0.06(0.008) |
| | $\|\Delta_4\|_\infty$ | 0.02(0.002) | 0.06(0.004) | 0.03(0.004) |
| | $\|\Delta_4\|_2$ | 0.01(0.002) | 0.02(0.002) | 0.01(0.002) |
| | $\|\|\|\Delta_4\|\|\|_\infty$ | 0.01(0.001) | 0.01(0.001) | 0.01(0.002) |
| | SPE | 1.00(0.000) | 0.90(0.004) | 1.00(0.000) |
| | SEN | 1.00(0.000) | 1.00(0.000) | 1.00(0.000) |
| | MCC | 1.00(0.002) | 0.63(0.008) | 1.00(0.000) |

(Spellman et al., 1998). We can treat this data set as a 3-way tensor data, where the first way is gene with $m_1 = 404$, second way is the time point with $m_2 = 12$ and the third way is the condition with $m_3 = 2$. In addition, each sample batch of Omberg et al. (2009) is treated as an independent sample for a total of $N = 4$ samples. The original expression data are log-transformed. The expression levels of each gene are scaled to zero mean and unit variance across the four samples.

We apply our penalized estimation using the adaptive Lasso penalty to estimate the precision matrices, where the initial estimates are obtained using the $\ell_1$ norm penalty. The tuning parameters are selected based on a 4-fold cross-validation. The conditional independency graph for genes that are linked is shown in Figure 1.The genes that are regulated at the same cell-cycle phases are colored with the same colors. It is interesting to note that genes that are regulated by the same cell cycle phases tend to link together.

Figure 2(a) shows the Raster plot of the eigenvectors of the correlation matrix derived from $\widehat{\mathbf{\Sigma}}_2 = \widehat{\mathbf{\Omega}}_2^{-1}$. This matrix describes the correlation among the 12 time points during the cell cycle process. Each row of Figure2(a) corresponds to an eigenvector sorted by descending eigenvalues. These eigenvectors are the x-eigengenes of Omberg et al (2009). The first x-eigengene represents a constant expression level. The second eigengene represents the contrast in gene expression between the odd and even time points. The third and forth x-eigengenes reflect the gene expression changes during the cell cycle process. In Figure 2(b), points are drawn on a plane with the third x-eigengene on the $\theta = 0$-axis and the forth on the $\theta = \pi/2$-axis, normalized together with the fifth x-eigengene, clearly showing the periodic expression patterns of genes during the cell cycle process.

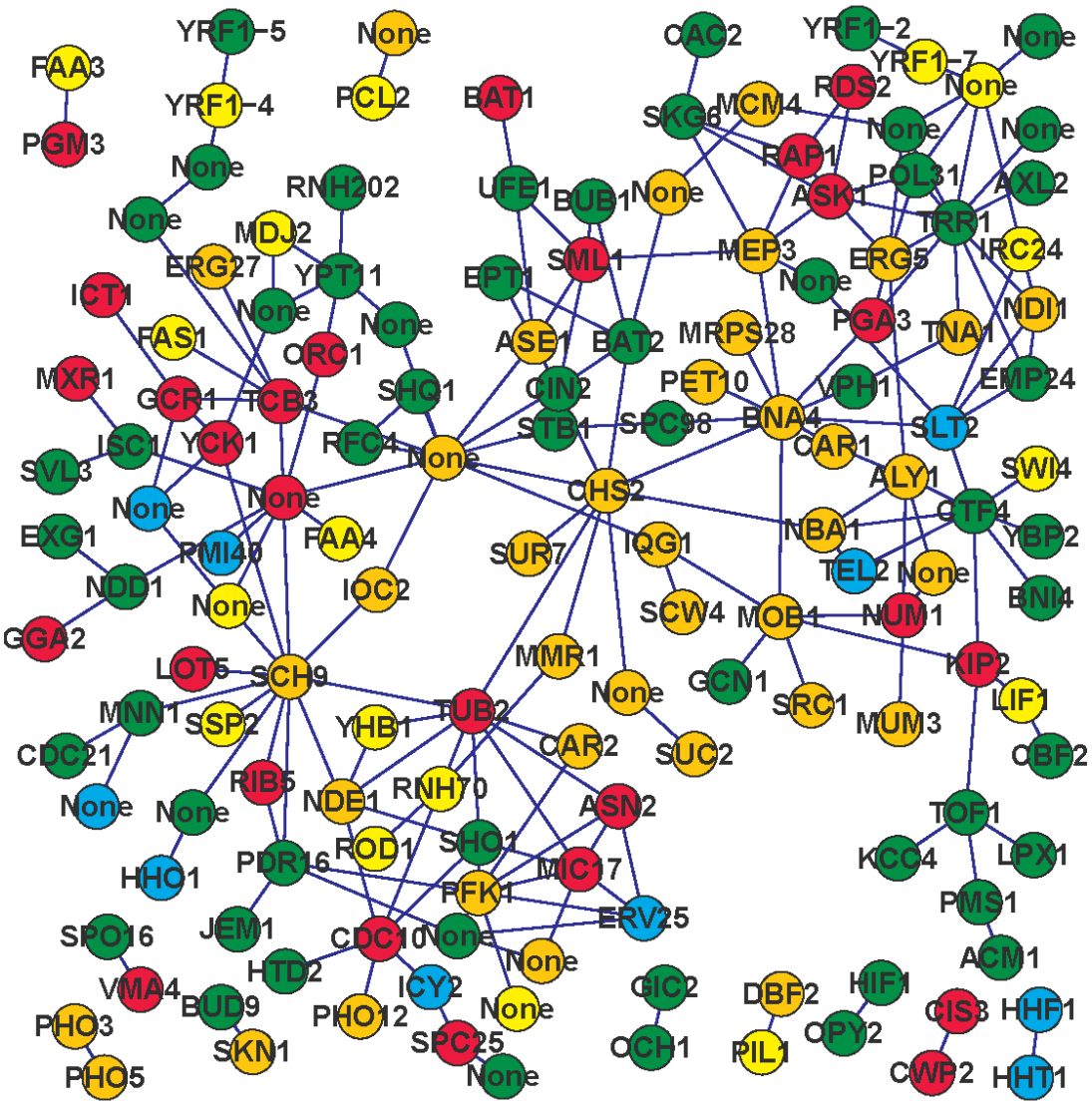
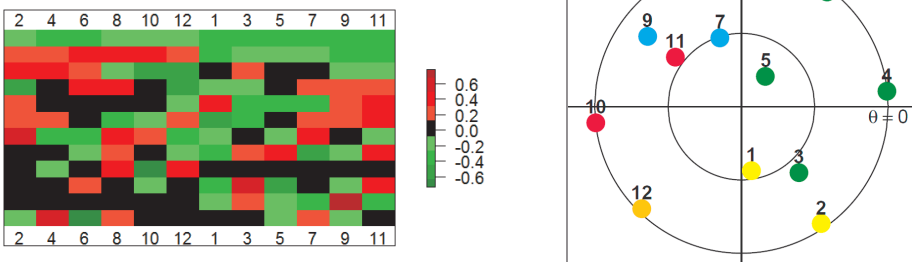



Figure 1: Gaussian graph of 150 yeast cell cycle associated genes. The colors indicate the cell-cycle phases that the genes are regulated. Green: G1 phase; orange: G2/M; yellow: M/G1; blue: S; Red: S/G2.




(a) Raster plot of the eigenvectors of the correlation matrix of the time points. Each row represents an eigenvector.

(b) Plot of the third and forth eigenvectors on the $\theta = 0$ and $\theta = \pi/2$ axes.

Figure 2: Plot of the eigenvectors of the time points correlation matrix based on the estimated time point covariance matrix $\widehat{\Sigma}_2 = \widehat{\Omega}_2^{-1}$

.

# 8 Conclusions and Discussion

Motivated by analysis of gene expression data measured at different time points and under different experimental conditions on the same set of samples, we have proposed to apply the tensor normal distribution to model the data jointly and have developed a penalized likelihood method to estimate each way's precision matrix assuming that these matrices are sparse. Our simulation results have clearly demonstrated the proposed penalized estimation method results in better estimates of the precision matrices and better identification of the corresponding graphical structures than naive alternatives. Our theoretical and numerical results shows that for the tensor data, the effective sample size for estimation of each matrix can be quite large although the independent observations are only a few. The tensor normal distribution provides a natural way of modeling the dependency of data indexed by different sets. If the underlying precision matrices are sparse, the proposed penalized likelihood estimation can lead to identification of the non-zero elements in these precision matrices. We observe that the proposed $l_1$ regularized estimation can lead to better estimates of these sparse precision matrices than the MLEs. How to extend the proposed method to non-

normal data is a future research direction.

# Acknowledgements

This research was supported by National Natural Science Foundation of China (grant No. 11201479) and NIH grants CA127334 and GM097505.