Tensor on tensor regression with tensor normal errors and tensor network states on the regression parameter

Carlos Llosa

Iowa State University, cllosa@iastate.edu

Follow this and additional works at: https://lib.dr.iastate.edu/creativecomponents

Part of the Multivariate Analysis Commons, Statistical Models Commons, and the Statistical Theory Commons

Recommended Citation

Llosa, Carlos, "Tensor on tensor regression with tensor normal errors and tensor network states on the regression parameter" (2018). Creative Components. 82.
https://lib.dr.iastate.edu/creativecomponents/82

This Creative Component is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Creative Components by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Tensor on Tensor Regression with Tensor Normal Errors and Tensor Network States on the Regression Parameter

Carlos Llosa
Department of Statistics, Iowa State University
November 22, 2018

Abstract

With the growing interest in tensor regression models and decompositions, the tensor normal distribution offers a flexible and intuitive way to model multi-way data and error dependence. In this paper we formulate two regression models where the responses and covariates are both tensors of any number of dimensions and the errors follow a tensor normal distribution. The first model uses a CANDECOMP/PARAFAC (CP) structure and the second model uses a Tensor Chain (TC) structure, and in both cases we derive Maximum Likelihood Estimators (MLEs) and their asymptotic distributions. Furthermore we formulate a tensor on tensor regression model with a Tucker structure on the regression parameter and estimate the parameters using least squares. Additionally, we find the fisher information matrix of the covariance parameters in an independent sample of tensor normally distributed variables with mean 0, and show that this fisher information also applies to the covariances in the multilinear tensor regression model [6] and tensor on tensor models with tensor normal errors regardless of the structure on the regression parameter.

1 Introduction

Multi-dimensional datasets are becoming more widely spread across multiple disciplines. Examples include multilinear relational data in political science and sociology and multi-dimensional imaging data in fields such as neurology and forestry. Multidimensional arrays have also been used in experimental design for treating $n$–factor crossed layouts, or multi-dimensional balanced split-plots [36]. Since the parameters necessary to model multi-dimensional datasets using traditional statistical methods grows exponentially with the number of dimensions, the past decade has seen a growing interest in models that consider high dimensionality. The tensor GLM framework proposed in [42] and extended in [24] offers a way to regress a univariate response using tensor-valued predictors, and this task became known as tensor regression. However, several other methods for the same task have been proposed, including Bayesian Tensor Regression [14], Tensor Envelope Partial Least-Squares Regression [40], Hierarchical Tucker Tensor regression [19] and Support Tensor Machines [37]. Several other tensor regression frameworks have been proposed recently. See [23] and [34] for the task of regressing tensor valued responses using multivariate predictors, [6] for the case when the covariates and predictors are tensors of the same number of dimensions, and [41] and [25] when the response and covariates are tensors of any dimensions.

In this paper we explore tensor on tensor regression [25] from a frequentist point of view. We model the errors using the tensor normal distribution and find asymptotic distribution of the regression parameter, which is imposed both the CP and the TT decomposition to deal with overparameterization. We also show that imposing a Tucker structure to the regression parameter leads to non-estimability under tensor normal errors and thus we estimate the model using least squares.
The contributions of this paper are multiple. First, we provide tensor algebra notation and properties used in other disciplines such as physics and machine learning that can help with the development of multilinear statistics. Second, we review the tensor normal distribution, find new properties and generalize the MLE algorithm for maximum likelihood estimation [10]. Finally, we formulate two tensor on tensor regression models and derive their asymptotic distributions when the errors follow a tensor normal distribution, and one tensor on tensor model that uses least squares to the estimation of the parameters.

The rest of the paper is divided as follows. In section 2 we review tensor network diagrams, tensor algebra properties, the tensor normal distribution and linear regression with Tucker decomposition and Least Squares. Section 3 provides the asymptotics of all the models that follow the tensor normal distribution. Section 4 derive tensor on tensor regression models with normal errors. In section 5 we formulate tensor and study tensor on tensor diagrams, tensor algebra properties, the tensor normal distribution and linear regression on matrices. Finally, we formulate two tensor on tensor regression models and derive their asymptotic distributions when the errors follow a tensor normal distribution, and one estimation [10]. Finally, we formulate two tensor on tensor regression models and derive their asymptotic distributions when the errors follow a tensor normal distribution, and one

2 Preliminaries

Throughout this paper the trace is denoted using tr(·), the transpose using (·)′, the determinant using |·|, the Moore–Penrose inverse using (·)† and the identity matrix of size n using In. The Kronecker product is denoted using ⊗ and the Khatri-Rao product is denoted using □. The vec(·) operator stacks the columns of a matrix into a vector and the commutation matrix \( K_{k,l} \in \mathbb{R}^{kl \times kl} \) matches the elements of vec(A) and vec(A′), see [26]

\[
\text{vec}(A') = K_{k,l} \text{vec}(A), \quad A \in \mathbb{R}^{k \times l}.
\]

The half vectorization, denoted using vech, vectorizes the lower triangular side of a symmetric matrix. The duplication matrix, denoted using \( D_k \), is the full column rank matrix that matches the vectorization and half vectorization of a symmetric matrix

\[
D_k \text{ vech}(A) = \text{vec}(A), \quad A \in \mathbb{R}^{k \times k} \text{ is symmetric}.
\]

The following matrix identities are useful for dealing with tensors

Properties 2.1. Suppose \( A_1, \ldots, A_p \) are matrices of any size and \( \Sigma_1, \ldots, \Sigma_p \) are square matrices of any size. Then

\begin{align*}
\text{a. } & | \bigotimes_{i=p} \Sigma_i | = \prod_{i=1}^p | \Sigma_i |^{m_{-i}}, \quad m_{-i} = \prod_{i=1}^{j-1} \text{rank}(\Sigma_j). \\
\text{b. } & ( \bigotimes_{i=p} \Sigma_i )^{-1} = \left( \bigotimes_{i=p} \Sigma_i^{-1} \right), \quad ( \bigotimes_{i=p} A_i )^\dagger = \left( \bigotimes_{i=p} A_i^\dagger \right) \quad \text{and} \quad ( \bigotimes_{i=p} A_i )^\dagger = \left( \bigotimes_{i=p} A_i^\dagger \right). \\
\text{c. } & \bigotimes_{i=1}^l A_i = \left( \bigotimes_{i=1}^l A_i \right) \bigotimes \left( \bigotimes_{i=l+1}^p A_i \right), \quad l = 1, \ldots, p. \\
\text{d. } & \left( \bigotimes_{i=p} A_i \right) \left( \bigotimes_{i=p} B_i \right) = \bigotimes_{i=p} (A_i B_i), \quad \text{where } B_i \text{ are matrices such that } A_i B_i \text{ can be formed.} \\
\text{e. } & K_{m,n}' = K_{m,n}^{-1} = K_{n,m} \quad \text{and} \quad K_{n,m} K_{m,n} = I_{nm}. \\
\text{f. } & K_{p,m} (A_1 \otimes A_2) K_{n,q} = A_2 \otimes A_1, \\
\text{g. } & \text{vec}(A_1 \otimes A_2) = R_{A_1} \text{vec}(A_2), \quad R_{A_1} = (I_n \otimes K_{q,m}) (\text{vec } A \otimes I_q).
\end{align*}
2.1 Tensor network diagrams

Tensor network diagrams are useful for visualizing tensor manipulations. They were originally introduced in quantum physics to describe the Hilbert space interactions that occur in many-body problems (see [31]) and in high energy physics to represent invariants of quantum states (see [2]). They have been adapted to tensor models in machine learning in the past years (see [4] and [5]). These diagrams are critical in this paper because they allow us to identify tensor expressions that make estimation and formulation possible.

Each node corresponds to a tensor and the number of lines coming from the node represents a dimension, or mode. A node with no lines is a scalar, a node with one line is a vector, a node with two lines is a matrix and a node with \( p \) lines is a \( p \)-th dimensional tensor (see Figure 1a-d). The contraction between two same-sized modes from (possibly) distinct tensors sums over all elements in the modes being contracted while leaving the other modes intact. For instance, the matrix product \( AB \) contracts the rows of \( A \) with the columns of \( B \), and the resulting columns and rows of \( AB \) result from the columns and rows of \( A \) and \( B \) respectively (see Figure 1e). Contractions are represented in tensor network diagrams by merging the lines (modes) being contracted. A self contraction is a contraction between two lines coming from the same tensor, and is analogous to the trace (see Figure 1f).

2.2 Tensor notation and properties

Tensors are multidimensional arrays of numbers. The number of dimensions or indices of a tensor is called its order. Vectors are first order tensors and matrices are second order tensors. Following the notation in [4] we refer to scalars using lower case letters (ie. \( x \)), vectors using bold lower case italicized letters (ie. \( \mathbf{x} \)), matrices using capital letters (ie. \( X \)) and higher order tensors using bold underlined capital letters (ie. \( \underline{X} \)). The \((i_1, \ldots, i_p)\)th element of a tensor \( \underline{X} \) is denoted \( \underline{X}(i_1, \ldots, i_p) = x_{i_1, \ldots, i_p} \). Similarly, subtensors are obtained by fixing some of the indexes of the tensor. For instance mode \(-k\) fibers results from fixing all but the \( k \)th index of the tensor (ie: \( \underline{X}(i_1, ::, i_3, :) \)), and slices result by fixing all but two indexes (ie: \( \underline{X}(i_1, ::, i_3, :) \)). The simplest way of generating a \( p \)th order tensor is via the
vector outer product ($\circ$). For $p = 2$ we have $a_1 \circ a_2 = a_1 a_2'$, and for the general case

$$
\frac{p}{\bigcirc} a_j = \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} \left[ \prod_{j=1}^{p} a_j(i_j) \right] \frac{p}{\bigcirc} e_{i_j}^{m_j}, \quad a_j \in \mathbb{R}^{m_j}, \quad j = 1, \ldots, p,
$$

(2.1)

where $e_i^m \in \mathbb{R}^m$ is a unit basis vector with 1 as the $i$th element and 0 elsewhere.

A $p$-th order tensor $X$ has rank 1 if it can be written as the outer product of $p$ vectors. In general, the rank of a tensor is the minimum number of rank one tensors that add up to it. Since finding the rank of a higher order tensor is an NP-hard problem [20], most tensor manipulations choose the rank a priori based on the precision needed. Any $p$th order tensor $X \in \mathbb{R}^{m_1 \times \cdots \times m_p}$ with $(i_1, \ldots, i_p)$th element $x_{i_1,\ldots,i_p}$ can be expressed as

$$
X = \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} x_{i_1,\ldots,i_p} \left( \frac{p}{\bigcirc} e_{i_q}^{m_q} \right).
$$

(2.2)

Flattening or matricizing tensors allow us to use matrix algebra properties and efficient algorithms. The mode–$k$ matricization sets the mode–$k$ fibers as the columns of the resulting matrix

$$
X_{(k)} = \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} x_{i_1,\ldots,i_p} e_{i_k}^{m_k} \left( \bigotimes_{q=k}^{p} e_{i_q}^{m_q} \right)'.
$$

(2.3)

and the mode–$k$ canonical matricization maps the first $k$ modes to the rows and the rest of the modes to the columns of the resulting matrix

$$
X_{<k>} = \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} x_{i_1,\ldots,i_p} \left( \bigotimes_{q=k}^{p} e_{i_q}^{m_q} \right) \left( \bigotimes_{q=p}^{k+1} e_{i_q}^{m_q} \right)'.
$$

(2.4)

Another useful reshaping is the vectorization, which stacks the mode–1 fibers

$$
\text{vec}(X) = \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} x_{i_1,\ldots,i_p} \left( \bigotimes_{q=p}^{1} e_{i_q}^{m_q} \right).
$$

(2.5)

Note that in equation (2.5) we defined tensor vectorization in reverse lexicographic or column major order to avoid an inconsistency with matrix vectorization. Column major order is the convention used in languages such as R and Matlab. Since this convention leads to multiple Kronecker products in reverse order, we can also use the left Kronecker product ($\otimes_L$), which simply reverses the order of the Kronecker product (ie: $B \otimes_A = A \otimes_L B$).

As an example consider the third order tensor $Y \in \mathbb{R}^{3 \times 4 \times 2}$ shown in figure 2. This tensor was used also used in [21]. The fibers of $Y$ are

$$
Y = \begin{bmatrix}
1 & 13 & 16 & 19 & 22 \\
14 & 17 & 7 & 20 & 10 & 23 \\
2 & 15 & 5 & 18 & 8 & 21 & 11 & 24 \\
3 & 6 & 9 & 12 &
\end{bmatrix}
$$

Figure 2: A third order tensor with elements 1 through 24.
Mode 1 fibers
\[ \begin{bmatrix} 1 & 13 & 22 \\ 2 & 14 & 23 \\ 3 & 15 & 24 \end{bmatrix} \]

Mode 2 fibers
\[ \begin{bmatrix} 1 & 13 & 15 \\ 4 & 16 & 18 \\ 7 & 19 & 21 \\ 10 & 22 & 24 \end{bmatrix} \]

Mode 3 fibers
\[ \begin{bmatrix} 1 & 13 & 15 \\ 4 & 16 & 18 \\ 7 & 19 & 21 \\ 10 & 22 & 24 \end{bmatrix} \]

Some of the slices of \( Y \) are

\[ Y(:, :, 1) = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \quad Y(:, 3, :) = \begin{bmatrix} 7 & 19 \\ 8 & 20 \\ 9 & 21 \end{bmatrix}, \quad Y(1, :, :) = \begin{bmatrix} 1 & 13 \\ 4 & 16 \\ 7 & 19 \\ 10 & 22 \end{bmatrix} \]

The \( \text{mode-k} \) matricizations are

\[ X_{(1)} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix}, \quad X_{(2)} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{bmatrix}, \]

\[ X_{(3)} = \begin{bmatrix} 1 & 2 & \cdots & 12 \\ 13 & 14 & \cdots & 24 \end{bmatrix} \]

and the \( \text{mode-k} \) canonical matricizations are

\[ Y_{<1>} = X_{(1)}, \quad Y_{<2>} = X'_{(3)}, \quad Y_{<3>} = \text{vec}(Y) = \begin{bmatrix} 1 \\ \vdots \\ 24 \end{bmatrix} \]

The \( k \)-mode product \( (\times_k) \) between \( X \in \mathbb{R}^{m_1 \times \cdots \times m_p} \) and \( A \in \mathbb{R}^{d_k \times m_k} \) multiplies every \( \text{mode-k} \) fiber of \( X \) with \( A \), resulting in \( X \times_k A \in \mathbb{R}^{m_1 \times \cdots \times m_{k-1} \times d_k \times m_{k+1} \times \cdots \times m_p} \) with elements

\[ (X \times_k A)(i_1, \ldots, i_p) = \sum_{j_k=1}^{m_k} x_{i_1, \ldots, i_{k-1}, j_k, i_{k+1}, \ldots, i_p} a_{i_k, j_k}. \quad (2.6) \]

The \( k \)-th mode product applied to every mode is often called the Tucker product (see figure 3b) ([38] and [21])

\[ \{X; A_1, \ldots, A_p\} = X \times_1 A_1 \times_2 \cdots \times_p A_p = \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} x_{i_1 \ldots i_p} (\frac{p}{q=1} A_q(:, i_q)). \quad (2.7) \]

A diagonal tensor is a tensor with 0s everywhere but at the places where the indexes are all the same. They are represented in tensor network diagrams with nodes with a diagonal inside (see Figure 3a). If we let \( I \in \mathbb{R}^{R \times \cdots \times R} \) be a \( p \)th order diagonal tensor with 1s at the diagonal and 0s elsewhere

\[ I = \sum_{i=1}^{R} (\frac{p}{q=1} e_i^R), \quad (2.8) \]

then the Tucker product applied to \( I \) with respect to \( A_i \in \mathbb{R}^{m_i \times R} \) for \( i = 1, \ldots, p \) can be expressed in canonical polyadic, or CANDECOMP/PARAFAC (CP) ([3], [17] and [21]) form (see figure 3a)

\[ \{A_1, \ldots, A_p\} = \{I; A_1, \ldots, A_p\}. \quad (2.9) \]
Figure 3: tensor network diagrams of a) the CP decomposition, b) the Tucker decomposition, c) the inner product between two order $p$ tensors of the same order and size, and d) the partial—$k$ contraction between two order $p$ tensors with the same size along their first $k$ modes.

Note that as a result of equations (2.7), (2.8) and (2.9) the CP form can be expressed as

$$\mathbb{A} = \sum_{i=1}^{R} \left( \mathbb{B}_{i} \mathbb{C}_{i} \right),$$

which means that the rank of $\mathbb{A}$ is at most $R$. The tensor in equation (2.10) is referred to as a Kruskal tensor. The following identities are important for the case when $p = 2$

$$X_{(2)} = X', \quad \left[ X; A_1, A_2 \right] = A_1 X A_2', \quad \left[ A_1, A_2 \right] = A_1 A_2'.$$

Tensor decompositions allow us to decompose a tensor using lower order tensors. The CP decomposition decomposes a tensor in a series of rank one tensors (see figure 3a)

$$X = \sum_{j=1}^{R} \left( \mathbb{B}_{j} \mathbb{A}_{j} \right) = \left[ A_1, \ldots, A_p \right], \quad \mathbb{A}_j = [a_{j1} \ldots a_{jR}],$$

and is often solved using an alternating least squares (ALS) algorithm where each step is obtained using property (2.1.j) below. The Tucker decomposition decomposes the $p$th order tensor $X$ into $\left[ \mathbb{G}; A_1, \ldots, A_p \right]$, where $\mathbb{G}$ is a smaller tensor of the same order as $X$ and $A_1, \ldots, A_p$ are the factor matrices (see figure 3b). The Tucker decomposition is often solved in the form of the higher order singular value decomposition (HOSVD) [7] or hierarchical Tucker decomposition ([13] and [16]). One can also estimate the Tucker decomposition via ALS using theorems 2.1i and 2.1c below.

The inner product or contraction between two tensors of the same order and size is the sum of the product of their entries (see figure 3c)

$$\langle X, Y \rangle = \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} x_{i_1 \ldots i_p} y_{i_1 \ldots i_p},$$

(2.13)
This inner product is invariant under reshapings (see theorem 2.1g) and is used to define the Frobenius norm of a tensor
\[ ||X||_F = \sqrt{\langle X, X \rangle}. \] (2.14)

This definition of the Frobenius norm is consistent when \( p = 2 \) and \( p = 1 \) because
\[ \langle X, X \rangle = \text{tr}(XX^t), \quad \langle x, x \rangle = x^tx. \] (2.15)

Tensors can also be contracted along individual modes. The mode-\( (l) \) product or contraction between \( X \in \mathbb{R}^{m_1 \times \cdots \times m_p} \) and \( Y \in \mathbb{R}^{n_1 \times \cdots \times n_q} \), where \( m_l = n_k \), results in a tensor of order \( p + q - 2 \) where the \( l \)-th mode of \( X \) gets contracted with the \( k \)-th mode of \( Y \)
\[
X \times_m^l Y = \sum_{i_1=1}^{m_1} \cdots \sum_{i_{l-1}=1}^{m_{l-1}} \sum_{i_{l+1}=1}^{m_{l+1}} \cdots \sum_{i_p=1}^{m_p} \sum_{j_1=1}^{n_1} \cdots \sum_{j_{k-1}=1}^{n_{k-1}} \sum_{j_{k+1}=1}^{n_{k+1}} \cdots \sum_{j_q=1}^{n_q} \left\{ \sum_{s=1}^{m_l} x_{i_1, \ldots, i_{l-1}, t, i_{l+1}, \ldots, i_p} y_{j_1, \ldots, j_{k-1}, t, j_{k+1}, \ldots, j_q} \right\} \left( \overset{p}{\underset{s=1}{\sum}} e_{i_s}^{m_s} \right) \circ \left( \overset{q}{\underset{s=1}{\sum}} e_{j_s}^{n_s} \right). \] (2.16)

The same notation can be used to contract several modes between two tensors. The mode-\( (k_1, \ldots, k_a) \) contraction between \( X \) and \( Y \) where \( m_{k_1} = n_{l_1}, \ldots, m_{k_a} = n_{l_a} \) results in a tensor of order \( p + q - 2 \times a \) where the modes indicated in the product get contracted. The partial-\( k \) contraction is a special case where the first \( k \) modes of \( X \) and \( Y \) get contracted (See figure 3d)
\[ \langle X, Y \rangle_k = X \times_{1, \ldots, k} Y \in \mathbb{R}^{m_{k+1} \times \cdots \times m_p \times n_{k+1} \times \cdots \times n_q}. \] (2.17)

An important case of the partial contraction is when the first tensor has smaller order. For instance, if \( p < q \) and \( m_i = n_i \) for \( i = 1, \ldots, p \) then
\[ \langle X, Y \rangle = X \times_{1, \ldots, p} Y \in \mathbb{R}^{m_{p+1} \times \cdots \times n_q}. \] (2.18)

Another special case is the contraction of the last mode of \( X \) along with the first mode of \( Y \), denoted using \( \times^1 \)
\[ X \times^1 Y = X \times^1_p Y. \] (2.19)

The tensor trace can be seen as a self contraction between two outer modes; it has also been defined for multiple self-contractions. Suppose \( X \in \mathbb{R}^{R \times m_1 \times \cdots \times m_p} \), then
\[ \text{tr}(X) = \sum_{i=1}^{R} X[i, \ldots, i] \in \mathbb{R}^{m_1 \times \cdots \times m_p}. \] (2.20)
These last two definitions are useful for representing the Tensor Chain (TC) decomposition [32], or Matrix Product State (MPS) with periodic boundary conditions, as it is referred to in physics [31], which decomposes a $p$-th order tensor $\mathbf{X} \in \mathbb{R}^{m_1 \times \ldots \times m_p}$ into $p$ different third order tensors $\mathbf{G}^{(i)} \in \mathbb{R}^{g_i \times m_i \times m_{i-1}}$ for $i = 1, \ldots, p$ where $g_0 = g_{p+1}$ such that

$$
\mathbf{X} = \text{tr}(\mathbf{G}^{(1)}) \times \ldots \times \text{tr}(\mathbf{G}^{(p)}).
$$

(2.21)

See figure 4 for a tensor network diagram of the TC decomposition of a fifth order tensor. The following tensor algebra properties are critical in this paper. Theorem 2.1 can be found in [4] and theorems 2.1(i and j) can be found in [21].

**Theorem 2.1.** Suppose $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{m_1 \times \ldots \times m_p}$. Then

a. $\text{vec}(\mathbf{X}) = \left[ \begin{array}{c} \text{vec}(\mathbf{X}(i, \ldots, 1, \ldots, 1)) \\ \vdots \\ \text{vec}(\mathbf{X}(i, \ldots, m_{k+1}, \ldots, m_p)) \end{array} \right]$, $k = 1, \ldots, p$.

b. $\text{vec} \left( \sum_{i=1}^{p} a_i \right) = \sum_{i=1}^{p} \text{vec}(a_i)$, where $a_1, \ldots, a_p$ are vectors of any size.

For $p=2$: $\text{vec}(a_1 a_2') = a_2 \otimes a_1$

c. $\text{vec}([\mathbf{X}; A_1, \ldots, A_p]) = \left( \bigotimes_{i=p} A_i \right) \text{vec}(\mathbf{X})$, where $A_i \in \mathbb{R}^{n_i \times m_i}$ for any $n_i \in \mathbb{N}$.

For $p=2$: $\text{vec}(A_1 X A_2') = (A_2 \otimes A_1) \text{vec}(X)$.

d. $\text{vec}(\mathbf{X}) = \text{vec}(\mathbf{X}(1)) = \text{vec}(\mathbf{X}_{<1>})$, $l = 1, \ldots, p$.

e. $\mathbf{X}_{<p-1>} = \mathbf{X}'_{(p)}$

f. $\text{vec}(\mathbf{X} | \mathbf{B}) = \mathbf{B}'_{<l>} \text{vec}(\mathbf{X})$, $\mathbf{X} \in \mathbb{R}^{m_1 \times \ldots \times m_l}$, $l < p$.

g. $\langle \mathbf{X}, \mathbf{Y} \rangle = (\text{vec} \mathbf{X})' (\text{vec} \mathbf{Y}) = \text{tr}(\mathbf{X}_{(k)} \mathbf{Y}'_{(k)})$, where $k = 1, \ldots, p$ and $\mathbf{Y}$ and $\mathbf{X}$ have the same order and size.

h. $\langle \mathbf{X}, [\mathbf{Y}; \Sigma_1, \ldots, \Sigma_p] \rangle = \text{vec} \left( \text{vec} \mathbf{X} \right)' \left( \bigotimes_{i=p} \Sigma_i \right) \text{vec} \left( \text{vec} \mathbf{Y} \right)$, $\Sigma_i \in \mathbb{R}^{m_i \times m_i}$, $i = 1, \ldots, p$.

For $p=2$: $\text{tr}(\mathbf{X} \Sigma_2 Y' \Sigma_1') = (\text{vec} \mathbf{X})' (\Sigma_2 \otimes \Sigma_1) (\text{vec} \mathbf{Y})$

i. $\left( [\mathbf{X}; A_1, \ldots, A_p] \right)_{(k)} = A_k \mathbf{X}_{(k)} A_{-k}'$, where $A_{-k} = \bigotimes_{i=j}^{k} A_i$ and $A_i \in \mathbb{R}^{n_i \times m_{i-1}}$ for any $n_i \in \mathbb{N}$, $k = 1, \ldots, p$.

j. $\left( [A_1, \ldots, A_p] \right)_{(k)} = A_k \left( \bigotimes_{i=j}^{k} A_i \right)'$, $k = 1, \ldots, p$ and $A_1, \ldots, A_p$ have the same number of columns.

k. $\text{vec}(\mathbf{X}_{(k)}) = K_{(k)} \text{vec}(\mathbf{X})$, $K_{(k)} = \left( \prod_{i=k+1} m_i \otimes \prod_{i=1}^{k-1} m_i, m_k \right)$.

**Proof.** See appendix 7.1. \qed
2.3 The Tensor Normal Distribution and the Generalized MLE Algorithm

The multivariate normal distribution is perhaps the most important multivariate distribution in statistics. The following results are well known and will be used to construct the tensor normal distribution. See [29] for more details.

Properties 2.2. Suppose the random vector \( \mathbf{x} \in \mathbb{R}^m \) is distributed according to the multivariate normal distribution with mean \( \mu \) and positive definite covariance matrix \( \Sigma \) (ie: \( \mathbf{x} \sim N_m(\mu, \Sigma) \)) then:

a. If \( A \in \mathbb{R}^{p \times m} \) then \( A \mathbf{x} \sim N_p(A \mu, A \Sigma A^T) \). Note that if \( A \Sigma A^T \) is not positive definite then the distribution is called singular multivariate normal.

b. The multivariate normal \( \mathbf{x} \) has the probability density function

\[
f(\mathbf{x}; \mu, \Sigma) = (2\pi)^{-m/2} |\Sigma|^{-1/2} \exp \left( -\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right)
\]

c. If \( x_i \overset{iid}{\sim} N_m(\mu_i, \Sigma) \), \( i = 1, \ldots, n \), then

\[
\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \sim N_{nm} \left( \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_n \end{bmatrix}, I_n \otimes \Sigma \right).
\]

Intuitively, the multivariate normal distribution is highly useful in statistics. The following results are well known and will be used to construct the tensor normal distribution. See [29] for more details.

Definition 2.1. A random matrix \( X \in \mathbb{R}^{m_1 \times m_2} \) follows a matrix normal distribution with mean \( M \in \mathbb{R}^{m_1 \times m_2} \) and positive definite covariance matrices \( \Sigma_1 \in \mathbb{R}^{m_1 \times m_1} \) and \( \Sigma_2 \in \mathbb{R}^{m_2 \times m_2} \) (ie. \( X \sim N_{m_1,m_2} (M, \Sigma_1, \Sigma_2) \)) if and only if \( \text{vec}(X) \sim N_{m_1 \times m_2} (\text{vec}(M), \Sigma_1 \otimes \Sigma_2) \).

The tensor normal distribution was introduced in [1], [18], [28] and [30] as a generalization of the matrix normal distribution to multiple dimensions. It is also referred to as the array variate normal distribution and the multilinear normal distribution. To derive it consider the random order three tensor \( \mathbf{X} \in \mathbb{R}^{3 \times 2 \times 2} \) with marginally matrix normal frontal slices \( \mathbf{X}(i,:,:) \sim N_{3,2}(M_i, \Sigma_1, \Sigma_2) \) for some constants \( \sigma_i \), \( i = 1, 2 \). Then modifying equation (2.23) leads to the definition of the third order tensor normal distribution, also referred to as the trilinear normal distribution

\[
\text{vec}(\mathbf{X}) = \begin{bmatrix} \text{vec} \mathbf{X}(1,:,:) \\ \text{vec} \mathbf{X}(2,:,:) \end{bmatrix} \sim N_{3 \times 2 \times 2} \left( \begin{bmatrix} \text{vec} M_1 \\ \text{vec} M_2 \end{bmatrix}, \begin{bmatrix} \sigma_{11}^3 \Sigma_2 \otimes \Sigma_1 & \sigma_{12}^3 \Sigma_2 \otimes \Sigma_1 \\ \sigma_{21}^3 \Sigma_2 \otimes \Sigma_1 & \sigma_{22}^3 \Sigma_2 \otimes \Sigma_1 \end{bmatrix} \right). \tag{2.24}
\]
Definition 2.2. A random tensor \( \mathbf{X} \in \mathbb{R}^{m_1 \times \cdots \times m_p} \) follows a \( p \)th order tensor normal distribution with mean \( \mathbf{M} \in \mathbb{R}^{m_1 \times \cdots \times m_p} \) and positive definite mode covariance matrices \( \Sigma_i \in \mathbb{R}^{m_i \times m_i} \) for \( i = 1, \ldots, p \) (ie. \( \mathbf{X} \sim N_{m_1 \ldots m_p} (\mathbf{M}, \Sigma_1, \ldots, \Sigma_p) \)) if and only if \( \text{vec}(\mathbf{X}) \sim N_{m_1 \times \cdots \times m_p} (\text{vec}(\mathbf{M}), \otimes_{i=p}^1 \Sigma_i) \).

Note that not any multivariate normal vector can follow a tensor normal distribution if shaped into a tensor, but only multivariate normal vectors with the required Kronecker separable covariance structure. See [35] and [11] for tests on the assumption of Kronecker separability, which allows us to drastically reduce the number of free parameters required to estimate our covariance matrix. For instance consider the random tensor \( \mathbf{X} \in \mathbb{R}^{m_1 \times \cdots \times m_p} \). If \( \text{vec}(\mathbf{X}) \) follows a multivariate normal distribution then the covariance matrix has \( (\prod_{i=1}^p m_i + 1) \prod_{i=1}^p m_i / 2 \) free parameters. On the other hand if we let \( \mathbf{X} \sim N_{m_1 \ldots m_p} (\mathbf{M}, \Sigma_1, \ldots, \Sigma_p) \), then the covariance matrix of \( \text{vec}(\mathbf{X}) \) has only \( \sum_{i=1}^p (m_i + 1)m_i / 2 \) free parameters. The following results will be useful for tensor response regression with tensor normal errors.

Algorithm 1: The generalized MLE algorithm for optimizing the complete log likelihood \( l(A_1, \ldots, A_p) \) using block descent

- **Input:** Initial values \( A_2^{(0)}, \ldots, A_p^{(0)} \)
- \( k = 0 \)
- **while** convergence criteria is not met **do**
  - \( A_1^{(k+1)} \in \arg \max_{A_1} l(A_1, A_2^{(k)}, \ldots, A_p^{(k)}) \)
  - \( A_2^{(k+1)} \in \arg \max_{A_2} l(A_1^{(k+1)}, A_2, A_3^{(k)}, \ldots, \Sigma_p^{(k)}) \)
  - \( \vdots \)
  - \( A_p^{(k+1)} \in \arg \max_{A_p} l(A_1^{(k+1)}, \ldots, A_{p-1}^{(k+1)}, A_p) \)
- \( k = k + 1 \)
- **end**

Theorem 2.2. Suppose \( \mathbf{X} \sim N_{m_1 \ldots m_p} (\mathbf{M}, \Sigma_1, \ldots, \Sigma_p) \). Then

a. \( \mathbb{E} \left[ \mathbf{X} \mid A_1, \ldots, A_p \right] \sim N_{n_1 \ldots n_p} (\mathbb{E} \left[ \mathbf{M} \right], A_1 \Sigma_1 A_1', \ldots, A_p \Sigma_p A_p') \), \( A_i \in \mathbb{R}^{n_i \times m_i} \).

For \( p = 2 \):

\( A_1 X A_2' \sim N_{n_1 n_2} (A_1 M A_2', A_1 \Sigma_1 A_1', A_2 \Sigma_2 A_2') \)

b. \( \mathbb{E} \left[ \mathbf{X} \mid (m_k, m_{-k}) \right] \sim N_{m_k, m_{-k}} (\mathbf{M} (m_k), \Sigma_k, \Sigma_{-k}) \), \( \Sigma_{-k} = \otimes_{i=p}^{i=k} \Sigma_i \), \( m_{-k} = \prod_{i=1}^{i=k} m_i \), \( k = 1, \ldots, p \).

For \( p = k = 2 \):

\( \mathbb{E} \left[ \mathbf{X} \mid (M', \Sigma_2, \Sigma_1) \right] \)

c. \( \mathbf{X} \) has the probability density function

\[ f_X (\mathbf{X}; \mathbf{M}, \Sigma_1, \ldots, \Sigma_p) =
(2\pi)^{-\left( \prod_{i=1}^p m_i \right)} \prod_{i=1}^p |\Sigma_i|^{-\frac{m_i}{2}} \exp \left( -\frac{1}{2} \| \mathbf{X} - \mathbf{M} \|_{\Sigma_i^{-1}} \right) \]

For \( p = 2 \):

\[ f_X (X; M, \Sigma_1, \Sigma_2) =
(2\pi)^{-m_1 m_2 / 2} |\Sigma_1|^{-m_1 / 2} |\Sigma_2|^{-m_2 / 2} \exp \left( -\frac{1}{2} \text{tr} \left[ \Sigma_1^{-1}(X - M) \Sigma_2^{-1}(X - M)' \right] \right) \]
d. If $X \sim N_{p,r}(0, \Sigma_1, \Sigma_2)$ then $\mathbb{E}(X \otimes X) = \text{vec}(\Sigma_1) \text{vec}(\Sigma_2)'$.

Proof. See appendix 7.2 for a proof.

Based on an iid sample from the tensor normal distribution, the maximum likelihood estimator (MLE) of the mean is the sample mean and the MLEs of the covariance matrices have no closed form solution but depend on each other. The MLE or flip-flop algorithm [10] uses a two-step block relaxation algorithm to estimate the covariance matrices in the matrix normal model. See theorem 5.1 for novel asymptotic results for these maximum likelihood estimators. We provide the MLE algorithm [10] as any algorithm that optimizes the complete likelihood using block relaxation where each block is the profile complete likelihood (see algorithm 1). Other examples of block relaxation algorithms used in statistics include the expectation maximization (EM) algorithm and the alternating conditional expectation (ACE) algorithm. See [8] for a review on block relaxation algorithms in statistics. Note that the algorithm might converge to a local maxima and this has to be dealt with in a case-by-case manner. In many cases initializing the algorithm with $\sqrt{n}$-consistent estimators assure that the algorithm converges to estimators asymptotically equivalent to the MLE [22]. As a convergence criteria one can use the complete log likelihood or other criteria based on the change of parameters at each iteration. Examples of MLE algorithms in recent literature can be found in [6], [9], [12], [23] and [33].

2.4 Multivariate Linear Regressions and Multilinear Tensor Regression

The multivariate multiple linear regression model is critical to applications of the tensor normal distribution. We reformulate in the following way

$$Y_i = AX_i + E_i, \quad E_i \overset{iid}{\sim} N_{p,r} (0, \Sigma, I_r), \quad i = 1, \ldots, n.$$  \hfill (2.25)

Here $Y_i$ is a response matrix with predictor $X_i \in \mathbb{R}^{p \times r}$ and regression parameter $A \in \mathbb{R}^{p \times q}$. The columns of $Y_i$ are independent from each other, and therefore this is equivalent to the more common formulation of multivariate multiple linear regression

$$Y_i(:, j) = AX_i(:, j) + e_{ij}, \quad e_{ij} \overset{iid}{\sim} N_p (0, \Sigma), \quad i = 1, \ldots, n, \quad j = 1, \ldots, r.$$  \hfill (2.26)

**Theorem 2.3.** Suppose $nr - q \geq p$. Then the MLEs of $A$ and $\Sigma$ are

$$\hat{A} = \left( \frac{1}{n} \sum_{i=1}^{n} Y_i X_i' \right) \left( \frac{1}{n} \sum_{i=1}^{n} X_i X_i' \right)^{-1}, \quad \hat{\Sigma} = \frac{1}{nr} \sum_{i=1}^{n} (Y_i - \hat{A} X_i) (Y_i - \hat{A} X_i)'.$$

Proof. See appendix 7.3.1.

Another multivariate regression problem that is critical to this paper is

$$y_i = X_i \beta + e_i, \quad e_i \overset{iid}{\sim} N_p (0, \Sigma), \quad i = 1, \ldots, n.$$  \hfill (2.27)
Here $y_i \in \mathbb{R}^p$ is a response vector with predictor matrix $X_i \in \mathbb{R}^{p \times q}$ and regression parameter $\beta \in \mathbb{R}^q$. This is not a regular multivariate linear regression model and is analogous to generalized least squares regression, therefore we refer to it as *generalized multivariate regression*. For our purposes we will only concentrate in the estimation of $\beta$.

**Theorem 2.4.** Suppose $np \geq q$. Then given $\Sigma$ the MLE of $\beta$ is

$$\hat{\beta} = \left( \sum_{i=1}^{n} X_i' \Sigma^{-1} X_i \right)^{-1} \left( \sum_{i=1}^{n} X_i' \Sigma^{-1} y_i \right).$$

**Proof.** See appendix 7.3.2. \qed

Multilinear tensor regression is a model where the responses and predictors are tensors of the same order. It works by fitting a separate multivariate linear regression on each side of the tensor and iterating the regression parameters until convergence using a generalized MLE algorithm. It was proposed by P. Hoff in [6], where he also proposed a Bayesian approach to the estimation of the parameters. Figure 5 shows the trilinear tensor regression model and figure 6 shows a tensor network diagram of the general model, which can be expressed as

$$Y_i = [X_i; M_1, \ldots, M_p] + E_i, \quad E_i \overset{iid}{\sim} N_{m_1, \ldots, m_p}(0; \Sigma_1, \ldots, \Sigma_p), \quad i = 1, \ldots, n. \quad (2.28)$$

Here $Y_i \in \mathbb{R}^{m_1 \times \ldots \times m_p}$ is the tensor response, $X_i \in \mathbb{R}^{h_1 \times \ldots \times h_p}$ its corresponding tensor covariate and $E_i$ the tensor normal noise. The regression parameters or matrix factors are $M_j \in \mathbb{R}^{m_j \times h_j}$ for $i = 1, \ldots, p$. To estimate $M_k$ and $\Sigma_k$ with all the other parameters fixed first apply the $k-$mode matricization to both sides of equation (2.28) using theorems (2.1i) and (2.2b)

$$Y_{i(k)} = M_k X_{i(k)} M_{\perp k}' + E_i, \quad E_i \overset{iid}{\sim} N_{m_k, m_{\perp k}}(0; \Sigma_k, \Sigma_{\perp k}), \quad k = 1, \ldots, p. \quad (2.29)$$

Note that this has a bilinear tensor regression form itself, where the covariates and responses are both matric es. However the purpose of equation (2.29) is only to find steps in the MLE algorithm for $M_k$ and $\Sigma_k$ given $M_{\perp k}$ and $\Sigma_{\perp k}$. Using property (2.2a) we can write equation (2.29) as

$$Y_{i(k)} \Sigma_{\perp k}^{-1/2} = M_k X_{i(k)} M_{\perp k} \Sigma_{\perp k}^{-1/2} + E_i, \quad E_i \overset{iid}{\sim} N_{m_k, m_{\perp k}}(0; \Sigma_k, I_{m_{\perp k}}), \quad k = 1, \ldots, p, \quad (2.30)$$

which is a multivariate multiple regression model with MLEs given in theorem 2.3:

$$M_k = \sum_{i=1}^{n} Y_{i(k)} \Sigma_{\perp k}^{-1/2} M_{\perp k} X_{i(k)}' \left[ \sum_{i=1}^{n} X_{i(k)} M_{\perp k} \Sigma_{\perp k}^{-1} M_{\perp k} X_{i(k)}' \right]^{-1} \quad (2.31)$$
Figure 7: tensor network diagram of the tensor on tensor model

\[
\hat{\Sigma}_k = \frac{1}{nm-k} \sum_{i=1}^{n} (Y_i - M_kX_iM_k')\Sigma_k^{-1}(Y_i - M_kX_iM_k')'.
\] (2.32)

See theorem 5.3 for novel asymptotic results of the general model. The bilinear tensor regression model is commonly referred to as matrix variate regression. It was coined by C. Viroli in [39] but fully estimated by Hoff in [6] and Ding and Cook in [9]. Ding and Cook also proposed envelope models for the matrix variate regression model, which can be extended to the multilinear tensor regression model.

### 2.5 Tensor on Tensor Regression

Tensor on tensor regression works by estimating a tensor with the combined order of the tensor response and covariate. See Figure 7 for a tensor network diagram of the model, which can be expressed as

\[
Y_i = \langle X_i | B \rangle, \quad i = 1, \ldots, n.
\] (2.33)

Here \( Y_i \in \mathbb{R}^{m_1 \times \ldots \times m_p} \) is the tensor response, \( X_i \in \mathbb{R}^{h_1 \times \ldots \times h_l} \) its corresponding tensor covariate and \( B \in \mathbb{R}^{h_1 \times \ldots \times h_l \times m_1 \times \ldots \times m_p} \) is the regression parameter. This form was first proposed in [25], which constrained \( B \) to have CP structure and solved the regression problem using least squares and ridge regularization. Vectorizing both sides of equation (2.33) and using theorem (2.1f) we obtain

\[
\text{vec}(Y_i) = B_{\text{OLS}} \text{vec}(X_i), \quad i = 1, \ldots, n,
\] (2.34)

which is a multivariate regression problem and therefore the least squares solution is the same as the maximum likelihood estimator under normal errors given in theorem 2.3

\[
B_{\text{OLS}} = \left( \sum_{i=1}^{n} (\text{vec}(X_i))(\text{vec}(Y_i))^\prime \right)^{-1} \left( \sum_{i=1}^{n} (\text{vec}(X_i))(\text{vec}(Y_i))^\prime \right).
\] (2.35)

However solving for \( B_{\text{OLS}} \) often leads to overparameterization because of its size, which is \( \prod_{i=1}^{p} m_i \times \prod_{l=1}^{l} h_l \).

The rest of this paper will model the errors in equation (2.33) with the tensor normal distribution and impose different tensor structures on \( B \) to reduce its number of parameters. I will find MLEs of the tensor factors and covariance matrices and find asymptotic results.

### 3 Tensor on Tensor Regression with Tensor Normally Distributed Errors

Tensor on tensor regression can also be formulated using the tensor normal distribution

\[
Y_i = M + \langle X_i | B \rangle + E_i, \quad E_i \sim N_{m_1, \ldots, m_p}(0, \Sigma_1, \ldots, \Sigma_p), \quad i = 1, \ldots, n.
\] (3.1)

This model offers a flexible way to model the covariance structure in our data. For instance, one could impose \( \Sigma_1 \) to have an AR structure, and thus generalize vector autoregressive regression models to higher order tensors.
The intercept term in equation 3.1 can be found by first fitting the model on centered covariates and responses, and finding $\hat{M} = \bar{Y} - \langle \bar{X} \hat{B} \rangle$ afterwards, where $\bar{Y}$ and $\bar{X}$ are the mean response and covariate respectively. In this section we will present two novel tensor on tensor regression models based on equation (3.1) that differ on the structure of $B$. The first one imposes a CP structure and the second one imposes a TC structure, which has been shown to be effective in high dimensional problems. In both cases we will find maximum likelihood estimators for the tensor factors that depend on each other, and therefore we will optimize the likelihood using an MLE algorithm. Additionally the MLE algorithm will be used to find initial values for $\Sigma_1, \ldots, \Sigma_p$ as in [28], which have been shown to be $\sqrt{n}$-consistent estimators in [23]. This way we make sure the MLE algorithm finds estimators that are asymptotically equivalent to the maximum likelihood estimators. As a convergence criteria we can use the change in likelihood at each step; however the estimators for the covariances are already close to the maxima, and therefore we will use the change in Frobenius norm of $B$ instead.

### 3.1 Tensor on Tensor Regression with CP structure on the Regression Parameter

Consider model (3.1) where the regression parameter has the CP structure

$$B = \langle L_1, \ldots, L_l, M_1, \ldots, M_p \rangle.$$  

(3.2)

Here all the factor matrices have $R$ rows, meaning that the rank of $B$ is constrained to be at most $R$. This way the number of free parameters in $B$ are reduced from $\prod_{i=1}^l h_i \times \prod_{i=1}^p m_i$ to $R \times (\sum_{i=1}^l h_i + \sum_{i=1}^p m_i)$. Note that by assuming that all the covariance matrices are identity matrices, we generalize the model in [25], who solved the same regression model using least squares. See figure 8 for a tensor network diagram of this model. To estimate $M_1, \ldots, M_p$ and $\Sigma_1, \ldots, \Sigma_p$ first let

$$G_i = \langle X_i, L_1 \times 2 \ldots \times L_l \rangle, \quad i = 1, \ldots, n,$$

(3.3)
where $\mathbf{I} \in \mathbb{R}^{R \times \ldots \times R}$ is a diagonal tensor as in equation (2.8) and therefore $\mathbf{G}_i$ is a diagonal tensor too. Now we can write equation (3.1) as

$$
\hat{Y}_i = [\mathbf{G}_i; M_1, \ldots, M_p] + \mathbf{e}_i, \quad \mathbf{e}_i \overset{iid}{\sim} \mathcal{N}_{m_1, \ldots, m_p}(0, \Sigma_1, \ldots, \Sigma_p), \quad i = 1, \ldots, n. \quad (3.4)
$$

Equation (3.4) is a multilinear tensor regression as in equation (2.28) with MLEs given in equations (2.31) and (2.31):

$$
\hat{M}_k = \sum_{i=1}^{n} Y_{i(k)} \left( \bigotimes_{i \neq k} \Sigma_i^{-1} M_i \right) G_{i(k)}' \left( \bigotimes_{i \neq k} M_i' \Sigma_i^{-1} M_i \right) G_{i(k)}' \hat{M}_k = \sum_{i=1}^{n} Y_{i(k)} \left( \bigotimes_{i \neq k} \Sigma_i^{-1} M_i \right) G_{i(k)}' \left( \bigotimes_{i \neq k} M_i' \Sigma_i^{-1} M_i \right) G_{i(k)}' \quad (3.5)
$$

and

$$
\hat{\Sigma}_k = \frac{1}{nm_k} \sum_{i=1}^{n} \left( Y_{i(k)} - M_k G_{i(k)} \bigotimes_{i \neq k} M_i' \right) \left( \bigotimes_{i \neq k} \Sigma_i^{-1} \right) \left( Y_{i(k)} - M_k G_{i(k)} \bigotimes_{i \neq k} M_i' \right)' \quad (3.6)
$$

To find the maximum likelihood estimator of $L_k$ for $k = 1, \ldots, l$ first let

$$
\mathbf{H}_{ik} = \mathbf{X}_i \times_1^{l-1} \ldots \times_k^{l-1} \ldots \times_l^{l} \left[ \mathbf{L}_1, \ldots, \mathbf{L}_{k-1}, \mathbf{I}_R, \mathbf{L}_{k+1}, \ldots, \mathbf{L}_l, \mathbf{M}_1, \ldots, \mathbf{M}_p \right]. \quad (3.7)
$$

The role $\mathbf{H}_{ik}$ plays in the estimation of $L_k$ can be seen in figure 9. Both the rows and columns of $L_k$ are contracted to each $\mathbf{H}_{ik}$ and therefore we can only identify its vectorization. Vectorizing both sides of equation (3.1) with the constraint in (3.2) results in

$$
\text{vec} (\hat{Y}_i) = \mathbf{H}_{ik<2>} \text{vec} (L_k) + \mathbf{e}_i, \quad \mathbf{e}_i \overset{iid}{\sim} \mathcal{N}_{m_1 \times \ldots \times m_p}(0, \bigotimes_{i=p} \Sigma_i), \quad i = 1, \ldots, n. \quad (3.8)
$$

Note that equation (3.8) is a generalized multivariate regression as in equation (2.27) and therefore the MLE of vec ($L_i$) is given in theorem 2.4 as

$$
\text{vec}(L_k) = \left( \sum_{i=1}^{n} \mathbf{H}_{ik<2> \bigotimes_{i=p} \Sigma_i}^{-1} \mathbf{H}_{ik<2>} \right)^{-1} \left( \sum_{i=1}^{n} \mathbf{H}_{ik<2> \bigotimes_{i=p} \Sigma_i}^{-1} \text{vec} (\hat{Y}_i) \right). \quad (3.9)
$$

As in the usual CP decomposition, the columns of the factor matrices $M_1, \ldots, M_p, L_1, \ldots, L_p$ are normalized for identifiability and stability purposes and the column norms of the last factor matrix is stored separately. See theorem 5.4 for asymptotic results of our maximum likelihood estimators.

### 3.2 Tensor on Tensor Regression with TC structure on the Regression Parameter

Now consider model (3.1) where the regression parameter has the TC structure

$$
\mathbf{B} = \text{tr} (\mathbf{L}^{(1)} \times_1 \ldots \times_1 \mathbf{L}^{(l)} \times_1 \ldots \times_1 \mathbf{M}^{(1)} \times_1 \ldots \times_1 \mathbf{M}^{(p)}). \quad (3.10)
$$

Here $\mathbf{L}^{(i)} \in \mathbb{R}^{g_i \times \ldots \times g_l}$ for $i = 1, \ldots, l$ and $\mathbf{M}^{(i)} \in \mathbb{R}^{g_{i+l-1} \times \ldots \times g_{i+l}}$ and $i = 1, \ldots, p$ where $g_0 = g_{p+1}$. By imposing a TC structure on $\mathbf{B}$ we reduce its number of free parameters from $\prod_{i=1}^{l} g_i \times \prod_{i=1}^{p} m_i$ to $\sum_{i=1}^{l} g_i \times \sum_{i=1}^{p} g_{i+l-1} m_i$. This formulation offers the advantage that not all the factor matrices need to have the same number of columns, which is useful when the size of the modes in our response or covariates are radically different, as in color images. This is phenomena is referred to as skewness in modality. Note also that the TC decomposition has been shown to be effective in large dimensions. See figure 10 for a tensor network diagram of the regression model. To estimate $\mathbf{M}^{(k)}$ and $\Sigma_k$ for $k = 1, \ldots, p$ first let...
Figure 10: tensor network diagram of the tensor on tensor regression model with TC structure on the regression parameter.

Figure 11: tensor network diagram equivalent to figure 10.

\[
\begin{align*}
\mathbf{Z}_{ik} &= \mathbf{M}^{(k+1)} \times \cdots \times \mathbf{M}^{(p)} \times \mathbf{R}_i \times \mathbf{M}^{(1)} \times \cdots \times \mathbf{M}^{(k-1)} \\
\end{align*}
\]  

(3.11)

where \( \mathbf{R}_i \in \mathbb{R}^{t \times t} \) is the matrix

\[
\mathbf{R}_i = \left( \mathbf{L}^{(1)} \times \cdots \times \mathbf{L}^{(l)} \right)_{1 \times \cdots \times \hat{i} \times \cdots \times \hat{p}},
\]

(3.12)

The role that \( \mathbf{Z}_{ik} \) plays in the estimation of \( \mathbf{M}^{(k)} \) can be seen in figure 11. Note that \( \mathbf{Z}_{ik} \in \mathbb{R}^{g_{k+1} \times \cdots \times \hat{g}_k \times \cdots \times g_l \times \cdots \times g_1} \). Let \( \mathbf{Z}_{ik}^* \) be its reshaping such that \( \mathbf{Z}_{ik}^* \in \mathbb{R}^{g_{k+1} \times \cdots \times \hat{g}_k \times \cdots \times g_l \times \cdots \times g_1} \), such a reshaping can be done with the function \text{aperm} \ in base R.

Applying the \( k \)-th mode matricization to both sides of equation (3.1) leads to

\[
\mathbf{Y}_{i(k)} = \mathbf{M}^{(2)}_{(2)} \mathbf{Z}_{ik<2>}^* + \mathbf{E}_i, \quad \mathbf{E}_i \sim \mathcal{N}_{mk,m-k}(0; \mathbf{\Sigma}_k, \mathbf{\bar{\Sigma}}_k), \quad i = 1, \ldots, n.
\]

(3.13)

which is on the bilinear tensor regression form as in equation (2.29) and with MLEs given in equations (2.31) and (2.32)

\[
\mathbf{M}^{(2)}_{(2)} = \sum_{i=1}^{n} \mathbf{Y}_{i(k)} \left( \mathbf{\Sigma}_k^{-1} \mathbf{Z}_{ik<2>}^* \right) \left( \sum_{i=1}^{n} \mathbf{Z}_{ik<2>}^* \mathbf{\Sigma}_k^{-1} \mathbf{Z}_{ik<2>}^* \right)^{-1}
\]

(3.14)

and

\[
\mathbf{\hat{\Sigma}}_k = \frac{1}{nm-k} \sum_{i=1}^{n} \left( \mathbf{Y}_{i(k)} - \mathbf{M}^{(2)}_{(2)} \mathbf{Z}_{ik<2>}^* \right) \left( \mathbf{Y}_{i(k)} - \mathbf{M}^{(2)}_{(2)} \mathbf{Z}_{ik<2>}^* \right)^t.
\]

(3.15)

To estimate \( \mathbf{L}^{(k)} \) for \( k = 1, \ldots, l \) first let

\[
\mathbf{J}^{(k)} = \mathbf{X}_{1 \times \cdots \times \hat{g}_k \times \cdots \times \hat{g}_l \times \cdots \times \hat{g}_1} \times \mathbf{J}^{(k)}
\]

(3.16)

where \( \mathbf{J}^{(k)} \) is the MPS tensor without \( \mathbf{L}^{(k)} \)

\[
\mathbf{J}^{(k)} = \mathbf{L}^{(k+1)} \times \cdots \times \mathbf{L}^{(l)} \times \mathbf{M}^{(1)} \times \cdots \times \mathbf{M}^{(p)} \times \mathbf{L}^{(1)} \times \cdots \times \mathbf{L}^{(k-1)}.
\]

(3.17)
The role that $N_{ik}$ plays in the estimation of $L^{(k)}$ can be seen in figure 12. Note that $N_{ik} \in \mathbb{R}^{h_k \times g_k \times m_1 \times \ldots \times m_p \times g_k \times h_k \times g_k \times m_1 \times \ldots \times m_p}$. Let $N_{ik}^*$ be its reshaping such that $N_{ik}^* \in \mathbb{R}^{h_k \times g_k \times m_1 \times \ldots \times m_p \times h_k \times g_k \times m_1 \times \ldots \times m_p}$. Then vectorizing both sides of equation (3.1) leads to
\[
\text{vec}(Y_i) = N_{ik<3}^* \text{vec}(L^{(k)}) + e_i, \quad e_i \sim N_{m_1 \times \ldots \times m_p}(0, \frac{1}{i=p} \bigotimes_{i=p} \Sigma_i), \quad i = 1, \ldots, n. \tag{3.18}
\]
Note that equation (3.18) is a generalized multivariate regression as in equation (2.27) and therefore the MLE of $\text{vec}(L^l)$ is given in theorem 2.4 as
\[
\text{vec}(L^{(k)}) = \left( \sum_{i=1}^{n} N_{ik<3}^* \bigotimes_{i=p} \Sigma_i \right)^{-1} \left( \sum_{i=1}^{n} N_{ik<3}^* \bigotimes_{i=p} \Sigma_i \right)^{-1} \text{vec}(Y_i). \tag{3.19}
\]
See theorem 5.5 for asymptotic results of the regression parameters.

4 Tensor on Tensor Regression Using Least Squares

Note that all the methods found using MLE can be solved using LS by assuming that the covariance matrices are identity. In the case where the regression parameter is constrained to a CP form then this is equivalent to [25]. However some useful tensor structures are not estimable via MLE, and that includes the case where the regression parameter is constrained to a Tucker form. In this section we will solve this regression model using least squares.

4.1 Tensor on Tensor using The Tucker decomposition

Since the CP decomposition is a special case of the Tucker decomposition, in this section we will generalize the model proposed in [25] using the Tucker decomposition. In the context of tensor regression, this generalization is analogous to how [24] generalized [42]. This formulation is useful when data is skewed in modality. Consider the model in equation (2.33) where $B$ now has the Tucker form
\[
B = [Y; L_1, \ldots, L_l, M_1, \ldots, M_p]. \tag{4.1}
\]
See figure 13 for a tensor network diagram of the model. To estimate \( M_1, \ldots, M_p \) first let
\[
G_i = \langle X_i, V \times_1 L_1 \times_2 \cdots \times_l L_l \rangle_i, \quad i = 1, \ldots, n, \tag{4.2}
\]
Then for \( k = 1, \ldots, p \) the LS estimators of \( M_k \) is given in equation (3.6) where \( \Sigma_l = I_{m_l} \) for \( l = 1, \ldots, p \). To find the estimator of \( L_k \) first let
\[
H_{lk} = X_i \times^{k-1, k+1, \ldots} \| V; L_1, \ldots, L_{k-1}, I_{R_l}, L_{k+1}, \ldots, L_l, M_1, \ldots, M_p \|. \tag{4.3}
\]
Then the LS estimator of \( L_k \) is found in equation (3.9) where \( \Sigma_l = I_{m_l} \) for \( l = 1, \ldots, p \). To estimate \( V \) we first vectorize equation (2.33) using equation (2.1f)
\[
\text{vec}(Y_i) = \left( \frac{1}{i = p} \bigotimes M_i \right) V'_{<l>} \left( \frac{1}{i = l} \bigotimes L_i \right) \text{vec}(X_i), \quad i = 1, \ldots, n. \tag{4.4}
\]
Then because of property (2.1b) and theorem (2.1c) the OLS estimator of \( V \) in equation (4.4) is the same as in
\[
\text{vec}[\mathbf{Y}_i; M_i^1, \ldots, M_i^p] = V'_{<l>} \text{vec}[\mathbf{X}_i; L_i^1, \ldots, L_i^p], \quad i = 1, \ldots, n. \tag{4.5}
\]
Therefore the OLS estimator of \( V \) is
\[
\hat{V}_{<l>} = \begin{bmatrix}
(\text{vec}[\mathbf{Y}_i; L_i^1, \ldots, L_i^p])' \\
\vdots \\
(\text{vec}[\mathbf{Y}_i; L_i^1, \ldots, L_i^p])'
\end{bmatrix} \begin{bmatrix}
(\text{vec}[\mathbf{Y}_i; M_i^1, \ldots, M_i^p])' \\
\vdots \\
(\text{vec}[\mathbf{Y}_i; M_i^1, \ldots, M_i^p])'
\end{bmatrix}. \tag{4.6}
\]

The estimators are found iteratively using an ALS algorithm until the Frobenius norm of \( B \) is smaller than a certain threshold. One can also constraint the columns of \( L_1, \ldots, L_l, M_1, \ldots, M_p \) to have unit norm for identifiability and numerical stability.

5 Asymptotics

In this section we derive asymptotic results that apply to our models but also generalize to other models. First we find the fisher information matrix that corresponds to the covariance matrices under tensor normality with mean 0 (theorem 5.1) and show that they are the same as in tensor on tensor regression when the errors are assumed to be tensor normal (theorem 5.2). We proceed showing asymptotic independence between the MLE of the regression parameter and the covariance matrices (theorem 5.2), which is analogous to the multivariate case. This result allows us to focus on the regression parameter independently from the covariances. We next find the asymptotic variance of the parameters in the multilinear tensor regression model [6] (theorem 5.3) and finally we find the asymptotic variance of the parameters in the tensor on tensor regression model under normality when the repression parameter is assumed to have a CP form (theorem 5.4) and a TC form (theorem 5.5).

The following theorem finds the asymptotic variance of the MLEs of the covariance matrices under tensor normality with mean 0 [28]. This results applies also to the matrix normal case in the original MLE algorithm [10].

**Theorem 5.1.** Suppose \( \mathbf{X}_1, \ldots, \mathbf{X}_m \sim \text{iid} \mathcal{N}_{m_1, \ldots, m_p} (\mathbf{0}, \Sigma_1, \ldots, \Sigma_p) \).

Let \( \theta_\Sigma = \{ (\text{vech} \Sigma_1)', \ldots, (\text{vech} \Sigma_p)' \} \), then the Fisher information matrix \( \|\Sigma\| \) is a block matrix with \( k \)-th block diagonal matrix \( (k = 1, \ldots, p) \)

\[
\mathbb{E} \left( -\frac{s^2 l(\Sigma_k)}{\partial (\text{vech} \Sigma_k) \partial (\text{vech} \Sigma_k)'} \right) = \frac{nm_k}{2} D_{m_k} (\Sigma^{-1}_k \otimes \Sigma^{-1}_k) D_{m_k}
\]

and \( (k, l) \)-th off-diagonal block matrix where \( k, l = 1, \ldots, p, k \neq l \), \( m_{kl} = \prod_{i=1, i \neq k, l}^p m_i \),

\[
\mathbb{E} \left( -\frac{s^2 l(\Sigma_k, \Sigma_l)}{\partial (\text{vech} \Sigma_k) \partial (\text{vech} \Sigma_l)'} \right) = \frac{nm_{kl}}{2} D_{m_k} (\text{vec}(\Sigma^{-1}_k) \text{vec}(\Sigma^{-1}_l)') D_{m_l}.
\]
Proof. See appendix 7.4.1.

The next theorem shows that the asymptotic variance of the unstructured regression parameter is independent from the asymptotic variance of the covariance matrices. This result is important because we already found the asymptotic variance of the covariance matrices in theorem 5.1 and therefore it allows us to focus only in the asymptotic variance of the regression parameter, regardless of its structure.

**Theorem 5.2.** For \( i = 1, \ldots, n \) suppose that \( \mathbf{Y}_i \overset{iid}{\sim} \mathcal{N}_{m_1, \ldots, m_p} (\langle \mathbf{X}_i | \mathbf{B} \rangle, \Sigma_1, \ldots, \Sigma_p) \).
If \( \mathbf{B} = [\text{vec}(\mathbf{B}_i)^\prime | \mathbf{B}_i] \), where \( \mathbf{B}_i = [\text{vec}(\Sigma_1)^\prime, \ldots, \text{vec}(\Sigma_p)^\prime] \), then the asymptotic variance of \( \mathbf{\theta} \) is

\[
\text{avar}(\mathbf{\theta}) = \begin{bmatrix}
\left( \sum_{i=1}^{n} (\text{vec} \mathbf{X}_i)(\text{vec} \mathbf{X}_i)^\prime \right)^{-1} \otimes \left( \frac{1}{i=\mathbb{I}} \Sigma_i \right) & 0 \\
0 & \mathbb{I}^{-1}_\Sigma
\end{bmatrix}
\]

where \( \mathbb{I}_\Sigma \) is given in theorem 5.1.

**Proof.** See appendix 7.4.2.

Now we will use the previous two theorems to find the asymptotic variance of the multilinear tensor regression model [6] under tensor normal errors. This proof also applies to any type of tensor on tensor regression that uses theorem 2.3 to obtain the MLEs of the tensor factors and will be used in the proof of theorems 5.4 and 5.5. This result generalizes the asymptotic findings for matrix variate regression in [9].

**Theorem 5.3.** For \( i = 1, \ldots, n \) suppose that \( \mathbf{Y}_i \overset{iid}{\sim} \mathcal{N}_{m_1, \ldots, m_p} (\langle \mathbf{X}_i | \mathbf{M}_1, \ldots, \mathbf{M}_p \rangle, \Sigma_1, \ldots, \Sigma_p) \).
Let \( \mathbf{\theta}_M = [\text{vec}(\mathbf{M}_1)^\prime, \ldots, \text{vec}(\mathbf{M}_p)^\prime] \) where \( \mathbf{\theta}_\Sigma = [\text{vec}(\Sigma_1)^\prime, \ldots, \text{vec}(\Sigma_p)^\prime] \). Then the asymptotic variance of \( \mathbf{\theta}_M \) is

\[
\text{avar}(\mathbf{\theta}_M) = \\
\begin{bmatrix}
\mathbb{S}^{21} \otimes \Sigma_1 & P_1 \left( I_n \otimes \left( \frac{1}{i=\mathbb{I}} \Sigma_i \right) \right) P_2' & \ldots & P_1 \left( I_n \otimes \left( \frac{1}{i=\mathbb{I}} \Sigma_i \right) \right) P_m' & 0 \\
P_2 \left( I_n \otimes \left( \frac{1}{i=\mathbb{I}} \Sigma_i \right) \right) P_1' & \mathbb{S}^{22} \otimes \Sigma_2 & \ldots & P_2 \left( I_n \otimes \left( \frac{1}{i=\mathbb{I}} \Sigma_i \right) \right) P_m' & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
P_m \left( I_n \otimes \left( \frac{1}{i=\mathbb{I}} \Sigma_i \right) \right) P_1' & P_m \left( I_n \otimes \left( \frac{1}{i=\mathbb{I}} \Sigma_i \right) \right) P_2' & \ldots & \mathbb{S}^{2m} \otimes \Sigma_p & 0 \\
0 & 0 & \ldots & 0 & \mathbb{I}^{-1}_\Sigma
\end{bmatrix}
\]

where \( \mathbb{S}_{2k} = \sum_{i=1}^{n} \mathbf{X}_i(k) \left( \bigotimes_{i \neq k} \Sigma_i^{-1} \mathbf{M}_i \right) \mathbf{X}_i(k)^\prime \), \( P_k = (Q_k \otimes I_{m_k})(I_n \otimes K(k)) \),

\[
Q_k = \begin{bmatrix}
\left( \bigotimes_{i \neq k} \Sigma_i^{-1} \mathbf{M}_i \right) \mathbf{X}_i(k)^\prime \mathbb{S}^{-1}_{2k} \\
\vdots \\
\left( \bigotimes_{i \neq k} \Sigma_i^{-1} \mathbf{M}_i \right) \mathbf{X}_i(k)^\prime \mathbb{S}^{-1}_{2k}
\end{bmatrix}, \quad \text{and } K(k) \text{ is given in theorem 2.1k.}
\]

**Proof.** See appendix 7.4.3.

Note that the asymptotic variance of each vec \( \mathbf{M}_k \) has a Kronecker separable structure and therefore they asymptotically follow a matrix normal distribution (see definition 2.1). Now we will find the asymptotic variance of the MLEs of the matrix factors of the regression.
parameter when the errors are assumed to be tensor normal, as in section 3.1. We use theorem 5.3 for the asymptotic variance of $M_1, \ldots, M_p$ because they were found using theorem 2.3. Further, the asymptotic variance of $L_1, \ldots, L_p$ can be used whenever the MLEs of the matrix factors are obtained using theorem 2.4 and will be used in the proof of theorem 5.5.

**Theorem 5.4.** For $i = 1, \ldots, n$ suppose that $\mathbf{Y}_i \overset{iid}{\sim} N_{m_1, \ldots, m_p}(\mathbf{X}_i|\mathbf{B}), \Sigma_1, \ldots, \Sigma_p)$. Furthermore let $\mathbf{B} = [L_1, \ldots, L_t, M_1, \ldots, M_p]$ and $\mathbf{\theta}_{CP} = [\text{vec}(L_1)', \ldots, \text{vec}(L_t)', \text{vec}(M_1)', \ldots, \text{vec}(M_p)']'$, then the asymptotic variance of $\mathbf{\theta}_{CP}$ is

$$
\text{avar}(\mathbf{\theta}_{CP}) = \begin{bmatrix}
L & J \\
J' & M
\end{bmatrix}
$$

where

$$
L = \begin{bmatrix}
S_{11}^{-1} & R_1 \left( I_n \otimes \left( \frac{1}{p} \sum_{i=p} \Sigma_i \right) \right) R_2' & \ldots & R_1 \left( I_n \otimes \left( \frac{1}{p} \sum_{i=p} \Sigma_i \right) \right) R_l' \\
R_2 \left( I_n \otimes \left( \frac{1}{p} \sum_{i=p} \Sigma_i \right) \right) R_1' & S_{12}^{-1} & \ldots & R_2 \left( I_n \otimes \left( \frac{1}{p} \sum_{i=p} \Sigma_i \right) \right) R_l' \\
\vdots & \vdots & \ddots & \vdots \\
R_l \left( I_n \otimes \left( \frac{1}{p} \sum_{i=p} \Sigma_i \right) \right) R_1' & R_l \left( I_n \otimes \left( \frac{1}{p} \sum_{i=p} \Sigma_i \right) \right) R_2' & \ldots & S_{ll}^{-1}
\end{bmatrix},
$$

$$
J = \begin{bmatrix}
R_1 \left( I_n \otimes \left( \frac{1}{p} \sum_{i=p} \Sigma_i \right) \right) P_1' & \ldots & R_1 \left( I_n \otimes \left( \frac{1}{p} \sum_{i=p} \Sigma_i \right) \right) P_m' \\
\vdots & \ddots & \vdots \\
R_l \left( I_n \otimes \left( \frac{1}{p} \sum_{i=p} \Sigma_i \right) \right) P_1' & \ldots & R_l \left( I_n \otimes \left( \frac{1}{p} \sum_{i=p} \Sigma_i \right) \right) P_m'
\end{bmatrix},
$$

$$
S_{1k} = \sum_{i=1}^n H_{ik} \left( \frac{1}{p} \sum_{i=p} \Sigma_i \right)^{-1} H_{ik}', \quad R_k = \left[ S_{1k}^{-1} H_{ik} \left( \frac{1}{p} \sum_{i=p} \Sigma_i \right)^{-1} \ldots S_{1k}^{-1} H_{nk} \left( \frac{1}{p} \sum_{i=p} \Sigma_i \right)^{-1} \right],
$$

$\mathbf{H}_{ik}$ are given in equation (3.7) and $M = \text{avar}(\mathbf{\theta}_M)$ is given in theorem 5.3 with $S_{2k}$ and $Q_k$ given in theorem 5.3 by replacing $\mathbf{G}_i$ (given in equation 3.3) with $\mathbf{X}_i$.

**Proof.** See appendix 7.4.4.

Note that the asymptotic variance of each vec $L_k$ does not have a Kronecker separable structure and thus they don’t follow asymptotically a matrix normal distribution. The following theorem uses all of our previous results.

**Theorem 5.5.** For $i = 1, \ldots, n$ suppose that $\mathbf{Y}_i \overset{iid}{\sim} N_{m_1, \ldots, m_p}(\mathbf{X}_i|\mathbf{B}), \Sigma_1, \ldots, \Sigma_p)$. Furthermore let $\mathbf{B} = \text{tr}(\mathbf{L}(1)^{p+1} \times 1 \mathbf{M}(1)^{p+1} \times 1)$ and $\mathbf{\theta}_{TC} = [\text{vec}(\mathbf{L}(1))', \ldots, \text{vec}(\mathbf{L}(l))', \text{vec}(\mathbf{M}(1)^{(1)})', \ldots, \text{vec}(\mathbf{M}(1)^{(p)})']'$, then the asymptotic variance of $\mathbf{\theta}_{TC}$ is

$$
\text{avar}(\mathbf{\theta}_{TC}) = \begin{bmatrix}
L & J \\
J' & M
\end{bmatrix},
$$

where $L$ and $J$ are given in theorem 5.4 with $S_{1k}$ and $R_k$ obtained by replacing $H_{ik} < 2 >$ with $\mathbf{N}_{ik < 3 >}$ (given in equation 3.16) and $M = \text{avar}(\mathbf{\theta}_M)$ is given in theorem 5.3 with

$$
S_{2k} = \sum_{i=1}^n \mathbf{Z}_{ik < 2 >} \Sigma^{-1} \mathbf{Z}_{ik < 2 >}' \quad \text{and} \quad Q_k = \begin{bmatrix}
\left( \frac{1}{p} \sum_{i=p} \Sigma_i^{-1} \right) \mathbf{Z}_{ik < 2 >} \Sigma^{-1} \mathbf{Z}_{ik < 2 >}' \\
\vdots \\
\left( \frac{1}{p} \sum_{i=p} \Sigma_i^{-1} \right) \mathbf{Z}_{ik < 2 >} \Sigma^{-1} \mathbf{Z}_{ik < 2 >}'
\end{bmatrix}.
$$
Proof. See appendix 7.4.5.

6 Simulation

In this section we will compare four methods for regressing matrices on matrices. The first method is tensor on tensor regression using least squares [25], which we refer to as LSCP. The second method is tensor on tensor regression using maximum likelihood estimation, as in section 3.1, which we refer to as MLECP. The third method is matrix variate regression [9] (we use their implementation) or bilinear tensor regression [6], which we refer to as MATREG. The last method is matrix variate regression with envelope models [9], which we refer to as MATREGENV; we used stepwise BIC to select the envelope size, as implemented in the paper.

Our covariates $X_i$ and responses $Y_i$ are both $10 \times 10$ matrices. The $i$th covariate is composed elementwise from simulations of the normal distributions with mean $10i$ and variance 10. The regression parameter $\mathbf{B} \in \mathbb{R}^{10 \times 10 \times 10}$ is a tensor composed elementwise from simulations of the gamma distribution with parameters $\alpha = 200$, $\beta = 1$. The covariance matrices are both quadratic forms of matrices composed of normal distributions with variance 1 and means 4 and 1 corresponding to the first and second covariance matrix.

Note that in this simulation both MATREG and MLECP have the same number of paramaters because we restricted the CP rank of the regression parameters to $R = 5$. In this case MATREG has two $10 \times 10$ regression parameters and MLECP has four $5 \times 10$ regression parameters, and both have the same number of parameters in the covariance structure. LSCP has a smaller number of parameters because it lacks a covariance structure and MATREGENV also has a smaller number of parameters because of the sparcity constraint.

We simulated $n = 1500$ pairs of data $(X_i, Y_i)$ and implemented the methods using seven different sample sizes: $n = 100, 200, 300, 500, 800, 1000$ and 1500. For comparison we use the Mean Sum of Standardized Squared Errors (MSSSE) criteria defined as

$$MSSSE = \frac{1}{n} \sum_{i=1}^{n} ||\hat{\Sigma}_1^{-1/2}(Y_i - \hat{Y}_i^{est})\hat{\Sigma}_2^{-1/2}||_F^2, \quad \hat{Y}_i^{est} = \hat{M} + \langle X_i, \hat{\mathbf{B}} \rangle$$

and assume that the LS methods have both covariances set to identity. This MSSSE criteria is analogous to the univariate case $\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i^{est})^2/\hat{\sigma}$ where the estimation of $\hat{\sigma}$ is useful for assessing the predictive error.

Figure 14 shows a comparison between MSSSE among all methods. We can observe that the least squares method performs the worst because it did not account for the variability in the data. All the other methods perform well in the presence of heteroskedastic errors.

Figure 14: Comparison of MSSSE among all methods
Figure 15 shows a comparison between matrix variate regressions and MLECP. We can observe that MLECP outperforms the matrix variate regression even though they have the same number of parameters. This is perhaps the data was generated from a structure similar to MLECP.

![MSSSE plot](image)

Figure 15: Comparison of MSSSE among all methods except LSCP

Finally Figure 16 shows MLECP alone. We can observe that the MSSSE decays rapidly as the sample size increases.

![MSSSE plot](image)

Figure 16: MSSSE vs sample size for the MLCP method
7 Appendix

7.1 Proof of theorem 2.1: tensor algebra properties

Proof. For the following proofs note that the matrix product between $A \in \mathbb{R}^{m_1 \times m}$ with elements $A(i, j) = a_{ij}$ and $B \in \mathbb{R}^{m \times m_2}$ with elements $B(i, j) = b_{ij}$ can be expressed as

$$AB = \sum_{i_1=1}^{m_1} \sum_{j_1=1}^{m} \left[ a_{i_1j_1} (e_{i_1}^m) (e_{j_1}^m)' \right] \sum_{j_2=1}^{m_2} \sum_{i_2=1}^{m} \left[ b_{j_2i_2} (e_{j_2}^m) (e_{i_2}^m)' \right]$$

$$= \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \left[ \sum_{j=1}^{m} a_{i_1j} b_{j_2i} (e_{i_1}^m) (e_{j_2}^m)' \right]$$

(7.1)

a. First note that using the definition of matrix vectorization and theorem 2.1d

$$\text{vec}(X) = \text{vec}(X_{<p-1>}) = \begin{bmatrix} X_{<p-1>}(; 1) \\ \vdots \\ X_{<p-1>}(; m_p) \end{bmatrix} = \begin{bmatrix} \text{vec}(X(; \ldots , ; 1)) \\ \vdots \\ \text{vec}(X(; \ldots , ; m_p)) \end{bmatrix}.$$ 

The proof follows by doing the above procedure $k - 1$ more times.

b. Suppose $a_i \in \mathbb{R}^{m_i}, i = 1, \ldots , p$. Then using equations (2.1) and (2.5)

$$\text{vec}\left( \prod_{i=1}^{p} a_i \right) = \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} \left\{ \prod_{j=1}^{p} a_j(i_j) \right\} \left( \otimes_{q=p}^{m} e_{q}^m \right) = \prod_{i=p}^{1} a_i.$$

c. Using equations (2.5), (2.7) and theorem 2.1b

$$\text{vec}[X; A_1, \ldots , A_p] = \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} x_{i_1 \ldots i_p} \text{vec} \left( \prod_{q=1}^{p} A_q(; i_q) \right)$$

$$= \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} x_{i_1 \ldots i_p} \left( \otimes_{q=p}^{m} A_q(; i_q) \right) = \prod_{j=1}^{p} A_j \text{vec}(X).$$

d. Note that from equations (2.3) and (2.4) it follows that $X_{<1>} = X_{(1)}$. The rest follows from theorem 2.1b and property 2.1c

$$\text{vec}(X_{<l>}) = \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} x_{i_1 \ldots i_p} \text{vec} \left( \left( \otimes_{q=l}^{m} e_{q}^m \right) \left( \otimes_{q=p}^{l-1} e_{q}^m \right) \right)$$

$$= \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} x_{i_1 \ldots i_p} \left( \otimes_{q=p}^{m} e_{q}^m \right) = \text{vec}(X).$$

e. This follows from equations (2.4) and (2.5)

$$X_{<p-1>} = \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} x_{i_1 \ldots i_p} \left( \left( \otimes_{q=p}^{m} e_{q}^m \right) \left( \otimes_{q=1}^{p-1} e_{q}^m \right) \right)'$$

$$= \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} x_{i_1 \ldots i_p} \left( e_{i_p}^m \right) \left( \otimes_{q=p}^{l-1} e_{q}^m \right)' = X_{(p)}.$$

f. This proof follows from equations (2.3), (2.5) and (7.1)

$$X_{<l>} \text{vec} B = \sum_{i_1=1}^{m_1} \cdots \sum_{i_p=1}^{m_p} x_{i_1 \ldots i_p} \left( \left( \otimes_{q=l}^{m} e_{q}^m \right) \left( \otimes_{q=p}^{l-1} e_{q}^m \right) \right) \left[ \sum_{i_1=1}^{m_1} \cdots \sum_{i_l=1}^{m_l} b_{i_1 \ldots i_l} \left( \otimes_{q=l}^{m} e_{q}^m \right) \right]$$

$$= \sum_{i_1=1}^{m_1} \cdots \sum_{i_l=1}^{m_l} x_{i_1 \ldots i_l} b_{i_1 \ldots i_l} \left( \otimes_{q=l}^{m} e_{q}^m \right) = \text{vec}(X_{(l)} B).$$

g. The invariance of the inner product under vectorization follows from equations (2.5), (2.13) and (7.1)
\[(\text{vec } \mathbf{X})'(\text{vec } \mathbf{Y}) = \left[ \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} x_{i_1...i_p} \left( \frac{1}{q=p} e_{i_q}^{m_q} \right) \right] \left[ \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} y_{i_1...i_p} \left( \frac{1}{q=p} e_{i_q}^{m_q} \right) \right],\]

\[= \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} x_{i_1...i_p} y_{i_1...i_p} = \langle \mathbf{X}, \mathbf{Y} \rangle\]

whereas the invariance of the inner product under \(k\)-mode matricization follows from equations (2.3), (7.1) and the commutative property of the trace

\[\text{tr} (\mathbf{X}(k) \mathbf{Y}'(k)) = \text{tr} \left\{ \left[ \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} x_{i_1...i_p} e_{i_k}^{m_k} \left( \frac{1}{q=p} e_{i_q}^{m_q} \right) \right] \left[ \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} y_{i_1...i_p} \left( \frac{1}{q=p} e_{i_q}^{m_q} \right) \right] \right\},\]

\[= \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} x_{i_1...i_p} y_{i_1...i_p} \text{tr} \left( e_{i_k}^{m_k} e_{i_k}^{m_k} \right) \]

\[= \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} x_{i_1...i_p} y_{i_1...i_p} \text{tr} \left( e_{i_k}^{m_k} e_{i_k}^{m_k} \right) \]

\[= \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} x_{i_1...i_p} y_{i_1...i_p} = \langle \mathbf{X}, \mathbf{Y} \rangle.\]

h. This follows from theorems 2.1(a and c).

i. This follows from equation (2.7) and (7.1)

\[\begin{align*}
\left[ \mathbf{X}; A_1, \ldots, A_p \right](k) &= \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} x_{i_1...i_p} \left( \frac{1}{q=p} A_q(:, i_q) \right)(k) \\
&= \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} x_{i_1...i_p} A_k(:, i_k) \left( \frac{1}{q=p} A_q(:, i_q) \right)(k)' \\
&= \sum_{i_k=1}^{m_k} A_k(:, i_k) \left( e_{i_k}^{m_k} \right)' \left[ \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} x_{i_1...i_p} \left( \frac{1}{q=p} A_q(:, i_q) \right)' \right] \\
&= A_k \left[ \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} x_{i_1...i_p} \left( \frac{1}{q=p} A_q(:, i_q) \right)' \right] \\
&= A_k \left[ \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} x_{i_1...i_p} \left( \frac{1}{q=p} e_{i_k}^{m_k} \right) \left( \frac{1}{q=p} A_q(:, i_q) \right)' \right] \\
&= A_k \left[ \sum_{i=1}^{m_1} \sum_{i_p=1}^{m_p} x_{i_1...i_p} \left( \frac{1}{q=p} e_{i_k}^{m_k} \right) \left( \frac{1}{q=p} A_q(:, i_q) \right)' \right] \\
&= A_k \mathbf{X}(k) A'_k
\end{align*}\]

j. Following the same steps as in the proof of theorem 2.1i where our central tensor is given in equation (2.8)

\[\left[ A_1, \ldots, A_p \right](k) = \ldots = A_k \left[ \sum_{i=1}^{R} e_{i}^{R} \left( \frac{1}{q=p} A_q(:, i) \right)' \right] = A_k \left[ \sum_{i=1}^{R} \left( \frac{1}{q=p} A_q(:, i) \right)' \right] = A_k \left( \sum_{i=1}^{R} \left( \frac{1}{q=p} A_q(:, i) \right)' \right) = A_k \left( A_q(:, R) \right) = A_k \left( \frac{1}{q=p} A_q(:, i) \right) \]

k. This proof shows how to obtain the linear transformation of the reshaping of a vectorized tensor, which allows us to obtain the distribution of any reshaping of a multilinear distribution that is elliptically contoured. Using theorems 2.1a, 2.1d and 2.1e
\[ \text{vec}(X) = \begin{bmatrix} \text{vec} X_{(i,\ldots,i,1,\ldots,1)} \\ \vdots \\ \text{vec} X_{(i,\ldots,i,m_{k+1},\ldots,m_p)} \end{bmatrix} = \begin{bmatrix} \text{vec} (X_{(i,\ldots,i,1,\ldots,1)})^{<k-1>} \\ \vdots \\ \text{vec} (X_{(i,\ldots,i,m_{k+1},\ldots,m_p)})^{<k-1>} \end{bmatrix} \]

\[ = (I_{\prod_{i=1}^{p-k+1} m_i} \otimes K_{m_k,\prod_{i=1}^{k-1} m_i}) \]

\[ = (I_{\prod_{i=1}^{p-k+1} m_i} \otimes K_{m_k,\prod_{i=1}^{k-1} m_i}) \text{vec}(X_{(k)}). \]

Since the commutation matrix is orthogonal, we use properties 2.1e and 2.1f to obtain

\[ \text{vec}(X_{(k)}) = (I_{\prod_{i=1}^{p-k+1} m_i} \otimes K_{m_k,\prod_{i=1}^{k-1} m_i}) \text{vec}(X). \]

7.2 Proof of theorem 2.2: properties of the tensor normal distribution

Proof.

a. Using theorems 2.1c and 2.2a and definition 2.2

\[ \text{vec}(X_1 \times A_1 \times_2 \ldots \times_p A_p) \sim N_{\prod_{i=p}^1 m_i} \left( \left( \prod_{i=p}^1 A_i \right) \text{vec}(M), \left( \prod_{i=p}^1 A_i \right) \left( \prod_{i=p}^1 I_i \right) \left( \prod_{i=p}^1 A_i^\top \right) \right) \]

the rest follows from theorems 2.1c and 2.1d and definition 2.2.

b. This proof is key as it shows how the modes of a normally distributed tensor can be permuted. Intuitively, the k-mode matricization brings the k-mode to the position of the first mode and shifts the modes that were in between. From properties (2.2a) and (2.1k)

\[ \text{vec}(X_{(k)}) \sim N_{m_{k+1}} \left( M_{(k)}, \prod_{i=1}^{m_{k+1}} \left( \prod_{i=k+1}^{k-1} K_{m_k,\prod_{i=1}^{k-1} m_i} \left( \prod_{i=k-1}^{m_k} I_i \right) \left( \prod_{i=k-1}^{m_k} A_i \right) \right) \right) \]

\[ = N_{m_{k+1}} \left( M_{(k)}, \prod_{i=k+1}^{k-1} K_{m_k,\prod_{i=1}^{k-1} m_i} \right) \]

c. The pdf of vec(X) is given in property 2.2b, where the mean is vec(M) and the covariance matrix is \( \prod_{i=p}^1 \Sigma_i \). The product of determinants follows from property 2.1a and the rest follows using property 2.1b and theorem 2.1h.

d. \( \mathbb{E}(X \otimes X) = (\Sigma_1^{1/2} \otimes \Sigma_1^{1/2}) \mathbb{E}(Z \otimes Z) (\Sigma_2^{1/2} \otimes \Sigma_2^{1/2}) \), \( Z \sim N_{p,\Sigma_1}\), \( \prod_{i=p}^1 I_i \)

\[ = (\Sigma_1^{1/2} \otimes \Sigma_1^{1/2}) \left( \prod_{i=k+1}^{k-1} K_{m_k,\prod_{i=1}^{k-1} m_i} \right) \]

\[ = \left\{ \sum_{i=1}^p \sigma_{k,i}^{1/2} J_{i,j} \Sigma_2^{1/2} J_{i,j} \right\}_{k,l} \]

\[ = \left\{ \sigma_{k,l}^{2/2} \right\}_{k,l} \]

The product of determinants follows from property 2.1b and theorem 2.1h.
7.3 Maximum Likelihood Estimators for multivariate linear regressions

7.3.1 Proof of theorem 2.3

Proof.
The complete log likelihood can be obtained from theorem 2.2c

\[ l(A, \Sigma) = c - \frac{nr}{2} \log |\Sigma| - \frac{1}{2} \text{tr}(\Sigma^{-1}Z), \quad Z = \sum_{i=1}^{n} (Y_i - AX_i)(Y_i - AX_i)'. \]

Now we take the first differential and use the commutative property of the trace to group by differentials

\[ \partial l = \frac{1}{2} \text{tr} \left[ \Sigma^{-1} \partial \Sigma (-nr + \Sigma^{-1}Z) \right] - \text{tr} \left[ \Sigma^{-1} \partial A \left( \left( \sum_{i=1}^{n} X_i X_i' \right)A' - \left( \sum_{i=1}^{n} X_i Y_i' \right) \right) \right] \]

Our estimators are obtained by setting the above to zero. However note that unless \( nr - q \geq p \), \( \sum_{i=1}^{n} X_i X_i' \) is singular and the differential will never be 0. For concavity note that the second differential

\[ \partial^2 l = -\frac{nr}{2} \text{tr} \left[ \Sigma^{-1} \partial \Sigma \Sigma^{-1} \partial \Sigma \right] - \text{tr} \left[ \Sigma^{-1} \partial A \left( \sum_{i=1}^{n} X_i X_i' \right)(\partial A)' \right] \]

is strictly negative.

7.3.2 Proof of theorem 2.4

Proof.
The complete log likelihood can be obtained from the pdf of the multivariate normal distribution

\[ l(\beta) = c - \frac{1}{2} \sum_{i=1}^{n} (y_i - X_i \beta)' \Sigma^{-1} (y_i - X_i \beta). \]

Taking the first differential and grouping by the differential

\[ \partial l = (\partial \beta)' \left( \left( \sum_{i=1}^{n} X_i' \Sigma^{-1} y_i \right) - \left( \sum_{i=1}^{n} X_i' \Sigma^{-1} X_i \right) \beta \right). \]

Our estimator is obtained by setting the above to zero. However note that unless \( np \geq q \), \( \sum_{i=1}^{n} X_i' \Sigma^{-1} X_i \) is singular and the differential will never be 0. For concavity note that the second differential

\[ \partial^2 l = -(\partial \beta)' \left( \sum_{i=1}^{n} X_i' \Sigma^{-1} X_i \right)(\partial \beta) \]

is strictly negative.

7.4 Proof of Asymptotic Results

The following matrix calculus property is useful for multilinear statistics

**Theorem 7.1.** Suppose \( \Sigma \) is a symmetric non-singular matrix. Then

\[ \frac{\partial \text{vec} \Sigma^{-1}}{\partial \text{vec} \Sigma} = -(\Sigma^{-1} \otimes \Sigma^{-1}) \]

**Proof.** Consider the differential of the inverse

\[ \partial \Sigma^{-1} = -\Sigma^{-1} \partial \Sigma \Sigma^{-1} \]

The proof follows from vectorizing both sides \( \partial (\text{vec} \Sigma^{-1}) = -(\Sigma^{-1} \otimes \Sigma^{-1}) \partial (\text{vec} \Sigma) \)
7.4.1 Proof of theorem 5.1

Proof. We will start by finding the diagonal of the block Fisher information matrix. Using theorem 2.2b we know that for \(k = 1, \ldots, p\), \(X(\bar{k}) \sim N_{\mu_k, \Sigma_{\bar{k}}} (0, \Sigma_k, \Sigma_{-k})\). Based on this we find the second differential of the pdf (see theorem 2.2c) w.r.t. \(\Sigma_k\) only.

\[
\begin{align*}
    l(\Sigma_k) &= \text{const} - \frac{m_k}{2} \log |\Sigma_k| - \frac{1}{2} \text{tr}(\Sigma_k^{-1}X(\bar{k})\Sigma_{-k}^{-1}X'(\bar{k})) \\
    \partial l(\Sigma_k) &= -\frac{m_k}{2} \text{tr}(\Sigma_k^{-1}\partial \Sigma_k) + \frac{1}{2} \text{tr}(\Sigma_k^{-1}\partial \Sigma_k\Sigma_k^{-1}X(\bar{k})\Sigma_{-k}^{-1}X'(\bar{k})) \\
    \partial^2 l(\Sigma_k) &= \frac{m_k}{2} \text{tr}(\Sigma_k^{-1}\partial \Sigma_k\Sigma_k^{-1}\partial \Sigma_k) - \text{tr}(\Sigma_k^{-1}\partial \Sigma_k\Sigma_k^{-1}\partial \Sigma_k\Sigma_k^{-1}X(\bar{k})\Sigma_{-k}^{-1}X'(\bar{k}))
\end{align*}
\]

From the first moment of the wishart distribution we know that \(E(X(\bar{k})\Sigma_{-k}^{-1}X'(\bar{k})) = m_{-k}\Sigma_k\). Thus using the duplication matrix and theorem 2.1h we obtain

\[
E(-\partial^2 l(\Sigma_k)) = -\frac{m_k}{2} \text{tr}(\Sigma_k^{-1}\partial \Sigma_k\Sigma_k^{-1}\partial \Sigma_k) = \partial(\text{vec } \Sigma_k)' \left(-\frac{m_k}{2} D'_{m_{-k}}(\Sigma_k \otimes \Sigma_{\bar{k}}) D_{m_{-k}} \right) \partial(\text{vec } \Sigma_k).
\]

Now we will find the element in the position (2,1) of the Fisher information matrix. This finds the rest of the Fisher information matrix because the order of the covariances in the tensor normal distribution can be arbitrarily permuted along with its corresponding modes (as in property (2.2b)). Let \(\Sigma_{-12} = \otimes_{i=p}^3 \Sigma_i\) and \(m_{-12} = \prod_{i=3}^p m_i\). Then based on the distribution of \(X(1)\) we can write the terms in the likelihood that depend on \(\Sigma_1\) and \(\Sigma_2\) only as

\[
\begin{align*}
l(\Sigma_1, \Sigma_2) &= \frac{1}{2} \text{tr} \left( (\Sigma_{-12}^{-1} \otimes \Sigma_{-21}^{-1}) X(1) X(1)' \right) \Sigma_1^{-1} X(1) \\
&= \frac{1}{2} \text{vec}(\Sigma_{-12}^{-1} \otimes \Sigma_{-21}^{-1})(X(1) \otimes X(1))'(\Sigma_1^{-1}) \\
&= \frac{1}{2} \text{vec}(\Sigma_{-21}^{-1}) R'_{\Sigma_{-12}} (X(1) \otimes X(1))'(\Sigma_1^{-1}),
\end{align*}
\]

where the last step comes from property 2.1g. We obtain the second differential by applying theorem 7.1 twice. Taking its negative expectation results in

\[
-\text{E} \left( \frac{\partial^2 l(\Sigma_1, \Sigma_2)}{\partial(\text{vec } \Sigma_2)(\text{vec } \Sigma_1)'} \right) = \frac{1}{2} \text{vec}(\Sigma_{-12}^{-1} \otimes \Sigma_{-21}^{-1}) R'_{\Sigma_{-12}} \left\{ E(X(1) \otimes X(1))' (\Sigma_1^{-1} \otimes \Sigma_1^{-1}) \right\} \\
= \frac{1}{2} \text{vec}(\Sigma_{-12}^{-1} \otimes \Sigma_{-21}^{-1}) R'_{\Sigma_{-12}} \left\{ \text{vec}(\Sigma_1) (\text{vec}(\Sigma_{-12} \otimes \Sigma_2))' (\Sigma_1^{-1} \otimes \Sigma_1^{-1}) \right\} \quad \text{ (theorem 2.2d)} \\
= \frac{1}{2} \text{vec}(\Sigma_{-12}^{-1} \otimes \Sigma_{-21}^{-1}) R'_{\Sigma_{-12}} \text{vec}(\Sigma_{-12} \otimes \Sigma_2) (\text{vec } \Sigma_1)' \quad \text{ (property 2.1g)} \\
= \frac{m_{-12}}{2} (\text{vec } \Sigma_2)' (\text{vec } \Sigma_1)' \quad \text{ (see below)} \\
= \frac{m_{-12}}{2} (\text{vec } \Sigma_2)' (\text{vec } \Sigma_1)',
\]

The rest follows multiplying \(D_{m_2}'\) on the left and \(D_{m_2}\) on the right. Note that the following simplifies greatly

\[
\begin{align*}
R'_{\Sigma_{-12}} R_{\Sigma_{-12}} &= \left((\text{vec } \Sigma_{-12}') \otimes I_{m_2}) (I_{m_{12}} \otimes K_{m_{12},m_2}) (\text{vec } \Sigma_{-12}) \otimes I_{m_2} \right) \left(I_{m_{12}} \otimes K_{m_{12},m_2}) (\text{vec } \Sigma_{-12} \otimes I_{m_2}) \otimes I_{m_2} \right) \\
&= \left((\text{vec } \Sigma_{-12}') \otimes I_{m_2}) (I_{m_{12}} \otimes K_{m_{12},m_2}) (I_{m_{12}} \otimes K_{m_{12},m_2}) (\text{vec } \Sigma_{-12} \otimes I_{m_2}) \otimes I_{m_2} \right) \\
&= \left((\text{vec } \Sigma_{-12}') \otimes I_{m_2}) (I_{m_{12}} \otimes K_{m_{12},m_2}) (I_{m_{12}} \otimes K_{m_{12},m_2}) (\text{vec } \Sigma_{-12} \otimes I_{m_2}) \otimes I_{m_2} \right) \\
&= \text{tr}(\Sigma_{-12} \otimes I_{m_2}) \otimes I_{m_2} \\
&= m_{-12} I_{m_2}. \quad \blacksquare
\]
7.4.2 Proof of theorem 5.2

If we vectorize both sides of equation (3.1) we obtain

\[
\text{vec}(\mathbf{Y}_i) - \mathbf{B}_{\langle l \rangle}' \sim N_{m_1 \times \ldots \times m_p} \left( 0, \bigotimes_{i=p}^1 \Sigma_i \right), \quad i = 1, \ldots, n. \tag{7.2}
\]

Now let \( X = [\text{vec}(\mathbf{X}_1) \ldots \text{vec}(\mathbf{X}_n)] \), \( Y = [\text{vec}(\mathbf{Y}_1) \ldots \text{vec}(\mathbf{Y}_n)] \) and \( \Sigma = \bigotimes_{i=p}^1 \Sigma_i \). We can write the joint distribution of all of the data as

\[
Y - \mathbf{B}_{\langle l \rangle}'X \sim N_{\prod_{i=1}^m n_i} \left( 0, \Sigma, I_n \right)
\]

The complete log likelihood can be obtained from theorem (2.2c). We find the first two differentials with respect to both \( \Sigma \) and \( \mathbf{B}_{\langle l \rangle} \)

\[
l = -\frac{n}{2} \log |\Sigma| - \frac{1}{2} \text{tr} \left[ \Sigma^{-1} (Y - \mathbf{B}_{\langle l \rangle}'X)(Y - \mathbf{B}_{\langle l \rangle}'X)' \right]
\]

\[
\partial l = -\frac{n}{2} \partial(\Sigma^{-1} \partial \Sigma) + \frac{1}{2} \partial(\Sigma^{-1} \partial \Sigma \Sigma^{-1} (Y - \mathbf{B}_{\langle l \rangle}'X)(Y - \mathbf{B}_{\langle l \rangle}'X)') + \text{tr}(\Sigma^{-1} \partial \mathbf{B}_{\langle l \rangle}'X(Y - \mathbf{B}_{\langle l \rangle}'X)')
\]

\[
\partial^2 l = \frac{n}{2} \text{tr}(\Sigma^{-1} \partial \Sigma 
\]

\[
\partial \Sigma \Sigma^{-1} \partial \Sigma) - \text{tr}(\Sigma^{-1} \partial \Sigma \Sigma^{-1} \partial \Sigma \Sigma^{-1} (Y - \mathbf{B}_{\langle l \rangle}'X)(Y - \mathbf{B}_{\langle l \rangle}'X)') + 2 \text{tr}(\Sigma^{-1} \partial \mathbf{B}_{\langle l \rangle}'X(Y - \mathbf{B}_{\langle l \rangle}'X)') - \text{tr}(\Sigma^{-1} \partial \mathbf{B}_{\langle l \rangle}'X X' \partial \mathbf{B}_{\langle l \rangle}')
\]

Note that based on equation 7.2 the first two elements in the second differential correspond to the second differential of the complete log likelihood of \( n \) iid samples from the tensor normal distribution with mean \( \mathbf{0} \) and covariance matrices \( \Sigma_1, \ldots, \Sigma_p \). We already found this Fisher information matrix with respect to \( \theta_{\Sigma} \) in theorem 5.1. Further, upon taking the negative expectation the third element in the second differential (which contains both \( \partial \mathbf{B}_{\langle l \rangle} \) and \( \partial \Sigma \)) cancels and thus the non-diagonal block matrices in the Fisher information are 0. The only term left in the second differential is the last one, which contains \( \partial \mathbf{B}_{\langle l \rangle} \) twice. We can express it as

\[
-\text{tr}(\Sigma^{-1} \partial \mathbf{B}_{\langle l \rangle}'X X' \partial \mathbf{B}_{\langle l \rangle}') = (\text{vec} \partial \mathbf{B}_{\langle l \rangle}')' \left( \Sigma^{-1} \Sigma \right) (\text{vec} \partial \mathbf{B}_{\langle l \rangle}')
\]

Therefore the Fisher information matrix of \( \text{vec} \mathbf{B}_{\langle l \rangle}' \) is \( XX' \otimes \Sigma^{-1} \) and the lower right block matrix in the asymptotic variance is its inverse.

7.4.3 Proof of theorem 5.3

\textbf{Proof.} First notice that if we vectorize both sides of equation (2.28) we obtain

\[
\text{vec}(\mathbf{Y}_i) - \bigotimes_{i=p}^1 M_i \text{vec}(\mathbf{X}_i) \sim N_{m_1 \times \ldots \times m_p} \left( 0, \bigotimes_{i=p}^1 \Sigma_i \right), \quad i = 1, \ldots, n,
\]

which is exactly the form in equation 7.2 and therefore we know from theorem 5.2 that the elements on the right hand side and the bottom of the asymptotic variance are 0. We also know from theorem 5.2 that the bottom right element in the asymptotic variance is the inverse of the Fisher information in theorem 5.1. To obtain the rest of the asymptotic variance first note that for \( k = 1, \ldots, m \) we obtain from equation (2.31) that

\[
\hat{M}_k = [Y_{1(k)} \ldots Y_{1(k)}]Q_k.
\]
Vectorizing both sides and using theorem 2.1(c and k) results in

\[ \text{vec}(\bar{M}_k) = (Q'_k \otimes I_{m_k}) \text{vec}[Y_{1(k)} \ldots Y_{n(k)}] = (Q'_k \otimes I_{m_k}) \begin{bmatrix} \text{vec} Y_{1(k)} \\ \vdots \\ \text{vec} Y_{n(k)} \end{bmatrix} = P_k y, \]

where from property 2.2c and definition 2.2

\[ y = \begin{bmatrix} \text{vec} Y_1 \\ \vdots \\ \text{vec} Y_n \end{bmatrix} \sim N_{n \times \prod_{i=1}^m m_i} \left( \begin{bmatrix} \text{vec}(X_1 | B) \\ \vdots \\ \text{vec}(X_n, B) \end{bmatrix}, I_n \otimes \left( \bigotimes_{i=p}^1 \Sigma_i \right) \right). \]

Therefore using property 2.2a

\[ \hat{\theta} = \begin{bmatrix} \text{vec}(\bar{M}_1) \\ \vdots \\ \text{vec}(\bar{M}_m) \end{bmatrix} = \begin{bmatrix} P_1 \\ \vdots \\ P_m \end{bmatrix} y \sim N_{R(\prod_{i=1}^m m_i)} \left( \theta, avar(\theta) \right) \]

The rest of the proof comes from observing that for \( k = 1, \ldots, m, \)

\[ P_k \otimes \frac{1}{l} \sum_{i=p}^l P'_k = S_{2k}^{-1} \otimes \Sigma_k. \]

\[ \Box \]

### 7.4.4 Proof of theorem 5.4

Similar to the proof of theorem 5.3 we obtain that \( \text{vec}(\bar{M}_k) = P_k y \) for \( k = 1, \ldots, p \) and from equation 3.9 we can write \( \text{vec}(\bar{L}_k) = R_k y \) for \( k = 1, \ldots, l. \)

Therefore

\[ \hat{\theta} = \begin{bmatrix} \text{vec}(\bar{L}_1) \\ \vdots \\ \text{vec}(\bar{L}_l) \end{bmatrix} = \begin{bmatrix} R_1 \\ \vdots \\ P_1 \end{bmatrix} y \sim N_{R(\prod_{i=1}^l h_i)(\prod_{i=1}^m m_i)} \left( \theta, avar(\theta) \right) \]

The rest of the proof comes from observing that for \( k = 1, \ldots, l, \)

\[ R_k \otimes \frac{1}{l} \sum_{i=p}^l R'_k = S_{1k}^{-1} \]

and for \( k = 1, \ldots, m, \)

\[ P_k \otimes \frac{1}{m} \sum_{i=p}^m P'_k = S_{2k}^{-1} \otimes \Sigma_k. \]

### 7.4.5 Proof of theorem 5.5

This proof is identical to the proof of theorem 5.4 with parameter \( \hat{\theta}_{TC} \) instead of \( \hat{\theta}_{CP}. \)
References


