SOME STATISTICAL APPROACHES TO THE ANALYSIS OF MATRIX-VALUED DATA

Dong Wang

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2015

Approved by:

Young K. Truong

Haipeng Shen

J. S. Marron

Hongtu Zhu

Yin Xia

© 2015
Dong Wang
ALL RIGHTS RESERVED

ABSTRACT

DONG WANG: SOME STATISTICAL APPROACHES TO THE ANALYSIS OF MATRIX-VALUED DATA.
(Under the direction of Young K. Truong and Haipeng Shen.)

In many modern applications, we encounter data sampled in the form of two-dimensional matrices. Simple vectorization of the matrix-valued observations would destroy the intrinsic row and column information embedded in such data. In this research, we study three statistical problems that are specific to matrix-valued data.

The first one concerns dimension reduction for a group of high-dimensional matrix-valued data. We propose a novel dimension reduction approach that has nice approximation property, computes fast for high dimensionality, and also explicitly incorporates the intrinsic two-dimensional structure of the matrices. We discuss the connection of our proposal with existing approaches, and compare them both numerically and theoretically. We also obtain theoretical upper bounds on the approximation error of our method.

The second one is a group independent component analysis approach. Motivated by analysis of groups of high-dimensional imaging data, we develop a framework in the frequency domain through Whittle log-likelihood maximization. Our method starts with efficient population value decomposition, and then models each temporally-dependent source signal via parametric linear processes. The superior performance of our approach is demonstrated through simulation studies and the ADHD200 data.

The third one addresses the problem of regression with matrix-valued covariates. We consider the bilinear regression model, where two coefficient vectors are used to incorporate matrix covariates. We propose two maximum likelihood based estimators. Both estimators are shown to achieve the information lower bound and hence are theoretically optimal under the classical

asymptotic framework. We further propose a bilinear ridge estimator and derive its convergence property. The superior performances of the proposed estimators are demonstrated both theoretically and numerically.

ACKNOWLEDGEMENTS

I am grateful to my advisors, Professor Young Truong and Professor Haipeng Shen. I would like to thank them for their supports and encouragements, and for teaching me so much both in research and in life. Working with them over the past few years is one of the greatest experiences in my life.

I would like to express my gratitude to my committee members for their helpful comments and discussions on my research. I would like to thank Professor Steve Marron for bringing me to the area of object oriented data analysis and for teaching me so much on scientific writing. I would like to thank Professor Hongtu Zhu for allowing me audit his weekly group meeting and teaching me so much on neuroimaging data analysis. I would like to thank Professor Chuanshu Ji for all of his support and help ever since I came to UNC. I would like to thank Professor Yin Xia for her help and discussions on the hypothesis testing project.

I am thankful to Seonjoo Lee for sharing me her MATLAB codes. That was where I started my dissertation work. I am also thankful to Mihye Ahn for sharing us her analysis of the ADHD200 dataset.

I would like to thank all of my classmates in the Department of Statistics and Operations Research. Many thanks for all the help during the past five years.

I am grateful to my parents, my parents-in-law, and my wife, for their love, understanding, supports, and encouragements.

TABLE OF CONTENTS

LIST OF TABLES ix

LIST OF FIGURES x

1 INTRODUCTION 1

2 ADJUSTED POPULATION VALUE DECOMPOSITION 4

2.1 Introduction 4

2.2 Preliminaries 7

2.2.1 The model 7

2.2.2 Review of existing methods 8

2.3 Adjusted population value decomposition 10

2.3.1 The APVD algorithm 11

2.3.2 Memory complexity 11

2.3.3 Computation complexity 13

2.3.4 Connections of APVD with PVD and 2DSVD 14

2.4 Theoretical properties 15

2.4.1 Upper bound for the two-side type approximation 17

2.4.2 Upper bound for the one-side type approximation 17

2.5 Numerical studies 18

2.5.1 Simulation: subspace recovery and normalized reconstruction errors . . 19

2.5.2 Simulation: choices of r_L, r_R, k_i^u and k_i^v 21

2.5.3 Application to the AT&T Database of Faces 23

2.6 Conclusions 24
2.7 Proofs 25

3 GROUP PARAMETRIC INDEPENDENT COLORED SOURCES 31
3.1 Introduction 31
3.2 Background on ICA and PICS 35
3.2.1 Preliminaries 35
3.2.2 Independent Component Analysis 37
3.2.3 Parametric Independent Colored Sources 39
3.3 Methodology 41
3.3.1 Objective functions of GPICS 42
3.3.2 Optimization procedures 43
3.3.3 Dimension reduction of a group of images 46
3.4 Simulations 49
3.4.1 Blind source separation 49
3.4.2 Brain active region detection 51
3.5 Real data analysis 54
3.6 Conclusions 56

4 SCALAR-ON-MATRIX BILINEAR REGRESSION ANALYSIS 60
4.1 Introduction 60
4.2 Scalar-on-matrix bilinear regression analysis 62
4.2.1 Bilinear regression model 62
4.2.2 The maximum-likelihood based flip-flop estimator 65
4.2.3 A truncated flip-flop estimator 68
4.3 Asymptotic properties 69
4.3.1 Consistency and asymptotic covariance 69

4.3.2 Cramér-Rao lower bound and the MLE 73
4.3.3 Asymptotic efficiency 75
4.4 Bilinear ridge regression 76
4.4.1 Bilinear ridge estimator 76
4.4.2 Theoretical properties of the flip-flop bilinear ridge estimator 77
4.5 Simulations 79
4.5.1 Simulation setup 79
4.5.2 The effects of algorithm initialization 81
4.5.3 Simulation results 82
4.6 Discussion 82
4.7 Proofs of theorems 84
4.8 Proofs of lemmas 97

5 FUTURE WORK 108

REFERENCES 109

LIST OF TABLES

2.1 The maximum matrix sizes for the four estimation approaches. 12

2.2 Comparison of computation complexity for the four approaches. 13

2.3 Average results for four (m, n) pairs based on 100 simulation runs. Standard errors are shown in parentheses. 20

3.1 Summary of group structures. 42

3.2 Summary of Whittle log-likelihood and penalty terms for GPICS. $\sum_{g,i,j,k}$, $\sum_{g,i}$ and $\sum_{g}$ are abbreviations of $\sum_{g=1}^{G}\sum_{i=1}^{n_g}\sum_{j=1}^{M}\sum_{k=0}^{T-1}$, $\sum_{g=1}^{G}\sum_{i=1}^{n_g}$ and $\sum_{g=1}^{G}$ respectively. 43

3.3 AR coefficients. 50

3.4 Two sets of signals for the first three ICs. 52

4.1 Simulation results with fixed $snr = 1$ and $q = 20$ 83

4.2 Simulation results with fixed $snr = 2$ and $n = 2000$ 83

4.3 Simulation results with fixed $n = 5000$ and $q = 10$ 84

LIST OF FIGURES

2.1 Boxplots comparing four approaches for subspace recovery and normalized reconstruction errors over 100 simulation runs for $m = 100$ and $n = 50$. (a) Distances between the subspaces spanned by $\hat{L}$ and L defined in (2.7). (b) Distances between the subspaces spanned by $\hat{R}$ and R defined in equation (2.8). (c) Normalized reconstruction errors r defined in (2.5). 20

2.2 The choices of the numbers of singular vectors. (a) $k^u = k^v = r_L = r_R$ and vary from 1 to 20. (b) $k^u = k^v = 20$ and $r_L = r_R$ vary from 1 to 20. (c) $r_L = r_R = 5$ and $k^u = k^v$ vary from 5 to 20. The dotted vertical lines represent the locations of the true ranks of L and R. 21

2.3 One particular subject: the true images (first row) and the reconstructed images by APVD (second row), PVD (third row), 2DSVD (fourth row), and GLRAM (fifth row). 24

2.4 Ten representative subjects with one representative image per subject. 25

3.1 Boxplots of 100 simulation runs of Amari distances of estimated mixing matrices and the truth. The GPICS based estimates are consistently near the value zero while the estimates of fastICA based approaches vary greatly with means above zero. 51

3.2 Boxplots of 100 simulation runs for temporal correlations of the estimation sources and the truth. The GPICS based estimates are consistently near the value 1 for perfect correlation while the fastICA estimates vary greatly with means below 1. 51

3.3 Simulated spatial maps. 52

3.4 Temporal correlations of 100 simulation runs of the estimation temporal courses and the truth. The variability in the fastICA based approaches is clearly greater than the one with GPICS. 53

3.5 Estimated brain active regions for the case H_{00} 54

3.6 False positive and false negative rates as a function of SNR. 54

3.7 False rates as a function of the sample size. 55

3.8 ADHD-200 GPICS results. 57

3.9 Three networks identified by GPICS and fastICA based approaches for typically developing children group. 58

3.10 Three networks identified by GPICS and fastICA based approaches for ADHD-combined group. 58

3.11 Three networks identified by GPICS and fastICA based approaches for ADHD-inattentive group. 59

4.1 The simulation results for 100 random initials under Model I, with $snr = 1$ and $q = 20$. D is the ℓ_2 norm between the true and estimated coefficient vectors. 81

CHAPTER 1: INTRODUCTION

As the technology advances, data with complex structures have become more and more common. One particular type exhibits the form of a two-dimensional (2D) matrix, such as 2D image data, functional magnetic resonance imaging data, daily temperature data, call center data, mortality rate data, among others. These matrix-valued data are usually high dimensional and have both column-column and row-row correlations.

Traditional statistical approaches are mostly developed for the vector-valued data. Applying these approaches to matrix-valued data usually involves the so-called vectorization operation. This operation can bring two potential problems: (1) the resulting vector is usually in very high dimension and thus traditional statistical approaches are not applicable or scalable; (2) this process ignores the intrinsic 2D structure embedded in the matrix data. Hence, statistical approaches preserving the 2D nature are necessary and can potentially improve the analysis efficiency.

In this research, we develop new statistical approaches on the matrix-valued data from three perspectives.

In Chapter 2, we develop a computationally efficient approach to reduce the dimensionality of a group of high-dimensional matrices simultaneously. This approach is based on the model proposed in Ye (2005) which approximates multiple matrices by common left and right basis and subject-specific coefficients. The original estimator *generalized low rank approximations of matrices* (GLRAM) (Ye, 2005) is an iterative procedure which is computationally expensive. Ding and Ye (2005) proposed a non-iterative procedure which is computationally fast under the name *two-dimensional singular value decomposition* (2DSVD). Both GLRAM and 2DSVD are applicable to small or moderate sample size. To deal with massive dataset,

Crainiceanu et al. (2011) proposes the *population value decomposition* (PVD) approach which is a two-stage singular value decomposition procedure. Our procedure is a modified version of the PVD algorithm which differs from PVD in the second stage. We take the relative importance of singular vectors into consideration. Our procedure is computationally and mathematically almost equivalent to the GLRAM and 2DSVD approaches, but requires significantly less memory, and it also improves the performance of PVD by a considerable amount while keeping the attractive feature of PVD of little storage space. The error bound of our procedure has been established theoretically and its superior performance has been demonstrated empirically.

In Chapter 3, we study independent component analysis for a group of matrices. Traditional ICA originally aims at blind source separation for a single matrix by decomposing the observed data matrix into the product of the mixing matrix and the unobserved source signal matrix. When it is extended to the group analysis, many recent papers consider various assumptions on the mixing and signal matrices, such as whether the matrix is homogeneous or heterogeneous across subjects. Nevertheless, the existing group ICA methods do not consider the temporal correlation within each source which is prevalent in many applications and may play an important role in the analysis. The correlation structure is exploited in a single subject ICA paper by Lee et al. (2011) and the method is named *parametric independent colored sources* (PICS). We generalize their work to handle the correlation and the group nature simultaneous by an approach named *group PICS* (GPICS). GPICS models each source temporal signal via a parametric time series model, which enables us to solve for the time series parameters and the mixing matrices iteratively through maximizing the Whittle log-likelihood in the spectral domain. Various combinations of the homogeneity and heterogeneity assumptions can be flexibly accommodated in our model. Lastly, we make use of the novel group dimension reduction tool described in Chapter 2 to first reduce the size of one dimension, usually the extremely high dimensional spatial domain, and feed the resulting output to the likelihood, which innovatively makes the procedure more scalable and even applicable for data larger than the

memory size. The superior performance of GPICS is shown when compared with the popular group ICA approaches through simulation and a real data example.

Chapter 4 addresses the problem of regression with matrix-valued predictors by maintaining the matrix structure via a bilinear form. We consider the scenarios where the sample size is much larger than the two dimensions of the matrix. The bilinear form naturally leads to an iterative flip-flop procedure since it reduces to the linear model while one dimension is fixed. Although the iterative procedure has no guarantee to converge to the global optimum, which is the maximum likelihood estimator, we can still demonstrate that the stationary point achieves the same information lower bound as the maximum likelihood estimator. Due to the computational concerns, a non-iterative procedure is introduced as well, which is even more scalable and desirable for big data and meanwhile possesses an identical asymptotic efficiency as the iterative procedure. The advantage of the proposed methods over straightforward vectorization are demonstrated both theoretically and numerically. Moreover, we also consider the scenarios where the sample size is comparable or even smaller than the dimensions of the matrix. We propose a bilinear ridge estimator which is an extension of the ridge estimator for linear regression. We further establish an upper bound on the excess prediction error of the bilinear ridge estimator.

CHAPTER 2: ADJUSTED POPULATION VALUE DECOMPOSITION

2.1 Introduction

As the technology advances, matrix-valued data are more and more common. For instance, a typical *functional magnetic resonance imaging* (fMRI) data set is usually represented as a group of matrices of the same size, where each matrix is the measurement of the blood oxygen level dependent contrast for one subject, with each column corresponding to a vectorized three-dimensional image at a certain time point, and each row being a sequence of temporal observations for a particular brain voxel.

These matrices are often of high or even ultra-high dimension that needs a large amount of memory. For instance, a collection of fMRI data for 100 subjects may consist of 100 matrices with the spatial dimension corresponding to as many as $200,000$ voxels and the temporal dimension consisting of around 200 time points, which altogether requires about 30 *gigabytes* (GB) memory in double precision. Hence, it is crucial to develop a group-wise dimension reduction technique that is precise and scales well for high-dimensional data, which is the goal of the current chapter.

Most conventional dimension reduction techniques were developed for groups of vector-valued data, such as the popular principal component analysis (PCA) (Jolliffe, 2002). To apply these approaches directly to matrix-valued data, we need to vectorize each matrix. The conventional one dimensional (1D) PCA then projects vector-valued observations onto a set of orthogonal directions that preserve the maximum amount of variation in the data. These directions are characterized by the leading eigenvectors of the sample covariance matrix. However, the vectorization ignores the intrinsic two-dimensional (2D) structure embedded in the matrices, and creates high-dimensional vectors that increase computational/memory burden. This

usually makes the follow-up dimension reduction not efficient.

Several dimension reduction methods have been developed that incorporate the 2D structure of matrices. Motivated by 1DPCA, 2DPCA of Yang et al. (2004) projects each matrix onto the principal eigen-space of the row-row covariance matrix without vectorization. 2DPCA can also be understood through the perspective of a one-side-type low rank approximation to matrices. However, 2DPCA only takes into consideration the row-row covariance matrix. To fully capture both the row-row and column-column correlations, Ye (2005) proposed the *generalized low rank approximations of matrices* (GLRAM) approach which is a two-side-type low rank approximation. The idea of GLRAM originates from the minimization of the sum of squared residuals. The optimization criterion has no closed form solution and naturally leads to an iterative algorithm that can be slow. To achieve better computational efficiency, Ding and Ye (2005) proposed a non-iterative algorithm named *two-dimensional singular value decomposition* (2DSVD) which only implements eigen-decomposition on the row-row and column-column covariance matrices. Zhang and Zhou (2005) independently proposed *two-directional two-dimensional principal component analysis* ((2D)^{2}PCA) that is intrinsically equivalent to 2DSVD. The reduction of the computation cost inevitably makes the reconstruction error of 2DSVD and (2D)^{2}PCA larger than GLRAM, the optimal iterative procedure.

Besides computational speed, the limitation of the computer memory is another major hurdle that one has to tackle when analyzing massive data. Take the aforementioned fMRI data for example. The large amount of memory needed is beyond general computer capacity and the various algorithms discussed above are hence not implementable.

To cope with memory-demanding data and further speed up the computation, recently Crainiceanu et al. (2011) proposed the *population value decomposition* (PVD) approach that essentially boils down to a two-step *singular value decomposition* (SVD) algorithm. In the first step, SVDs are applied separately to individual matrices that are of relatively small size, and the leading left and right singular vectors are retained. This can be performed either in parallel

or sequentially and often requires much less memory. In the second step, the leading left and right singular vectors obtained in the first step are concatenated column-wise, respectively; and SVD is applied again to each concatenated matrix. These aggregated matrices are substantially smaller than the raw data matrices if one only keeps the few leading singular vectors. The resulting left singular vectors in the second step are used to obtain the final approximation for the original matrices. Obviously, ignoring the higher-order singular vectors in the first step results in less accuracy for PVD. But PVD effectively reduces the computational burden, and is applicable for high-dimensional matrices.

One drawback of PVD is that the computational efficiency does come at the price of reduced approximation accuracy. In this chapter, we further improve PVD and develop an *adjusted PVD* (APVD) algorithm that has the same computational cost and requires the same amount of memory as PVD, but produces more precise results. In fact, APVD often performs as accurate as GLRAM and 2DSVD for matrices of small to moderate sizes when they can be computed.

The key idea of the APVD modification arises from the observation that PVD assigns equal weights in the group-level SVD to those leading singular vectors obtained in the first SVD step. We all know that the singular vectors have a natural order of relative importance as reflected by the corresponding singular values. Hence, we adjust PVD by incorporating the relative importance of the singular vectors, which indeed results in a more accurate estimation of the group components. The first step of APVD is the same as PVD. While in the second step, our APVD procedure concatenates the scaled singular vectors, i.e. the product of the singular vectors and their corresponding singular values from each individual matrix, instead of concatenating just the singular vectors. Furthermore, we establish theoretical justification for APVD in terms of upper bound on the normalized reconstruction errors.

The rest of this chapter is organized as follows. In Section 2.2, we state the model and give a brief review of the GLRAM, 2DSVD, and PVD procedures. In Section 2.3, we then

describe the APVD algorithm, and compare the computational complexities of the various approaches along with their connections. The theoretical properties of APVD are studied in Section 4.3. Numerical comparisons through simulation studies and a classical face image data set are presented in Section 2.5, to show that APVD performs comparable to GLRAM and 2DSVD and better than PVD. We conclude in Section 2.6, and relegate all proofs to 2.7.

2.2 Preliminaries

2.2.1 The model

Consider there are I matrices of dimension $m \times n$, denoted as X_i, $i = 1, \ldots, I$. To achieve group dimension reduction for the matrices, a reasonable model can be written as

$$X_i = LW_iR^T + E_i, \tag{2.1}$$

where $L \in \mathbb{R}^{m \times r_L}$ and $R \in \mathbb{R}^{n \times r_R}$ are orthonormal matrices representing the left and right group components respectively, $W_i \in \mathbb{R}^{r_L \times r_R}$ is the individual coefficient matrix, and the error matrix E_i contains the individual approximation errors. Throughout this chapter, we assume that $\sum_{i=1}^{I} X_i = 0$. If we choose $r_L \ll n$ and $r_R \ll m$, the size of the individual component W_i ($r_L r_R$) is much smaller than the size of the original data (mn), which achieves the goal of dimension reduction.

The decomposition of X_i as in (2.1) is closely related to the SVD of a single matrix. Suppose $I = 1$, that is, there is only one matrix. Then the optimal L and R that minimize the sum of squares of the approximation errors in E_i are the $r = \min(r_L, r_R)$ leading left and right singular vectors of X_1, and W_1 can always be required to be a diagonal matrix with the r leading singular values. When $I > 1$, Model (2.1) relaxes the requirement that all of the subject-specific terms W_i should be diagonal matrices and only keep the orthonormal constraints of the group components. The reason is that the subspace spanned by the columns of L (or R)

can be thought of as the best rank r_L (or r_R) subspace that spans the column (or row) subspace of all the X_i's; the W_i's are the coefficients when projecting X_i onto L and R, which are not necessarily diagonal matrices.

2.2.2 Review of existing methods

The GLRAM, 2DSVD, PVD (and APVD) procedures offer different ways of estimating Model (2.1), as we shall review below. Least squares offers a natural criterion for model estimation. It can be shown that the least square estimator of W_i is given by $\widehat{W_i} = \hat{L}^T X_i \hat{R}$, once we obtain the group component estimates, $\hat{L}$ and $\hat{R}$. Therefore, for the rest of this chapter, we focus on the estimation of L and R. Moreover, for simplicity, we describe how each approach can be used to estimate the left component L; R can be estimated in the same way using the transpose of X_i.

The GLRAM of Ye (2005) borrows the minimum reconstruction error property of SVD and seeks L, R and W_i to minimize the reconstruction error in the least square sense:

$$\begin{aligned} \min_{L,R,W_i} \quad & \sum_{i=1}^{I} \|X_i - L W_i R^T\|_F^2, \\ \text{s.t.} \quad & R^T R = I_{r_R},\ L^T L = I_{r_L}, \\ & L \in \mathbb{R}^{m \times r_L},\ R \in \mathbb{R}^{n \times r_R} \text{ and } W_i \in \mathbb{R}^{r_L \times r_R}, \end{aligned} \tag{2.2}$$

where $\|\cdot\|_F$ is the matrix Frobenius norm.

Ye (2005) pointed out that the optimization problem (2.2) has no closed form solutions for L and R. Hence, GLRAM solves the problem in an iterative fashion. In each iteration, it alternates the updating of L (or R) as the leading r_L (or r_R) eigenvectors of $\sum_{i=1}^{I} X_i R R^T X_i^T$ (or $\sum_{i=1}^{I} X_i^T L L^T X_i$), by fixing R (or L) as the corresponding estimate obtained in the previous iteration. The algorithm terminates until it reaches a certain convergence criterion.

The 2DSVD of Ding and Ye (2005) offers an alternative estimation approach that is non-

iterative. It estimates L by either SVD of the concatenated data matrix or eigen-decomposition of the column-column covariance matrix. One can concatenate the matrices as $A = [X_1, X_2, \ldots, X_I]$, and then perform SVD on A to obtain the leading r_L left singular vectors as the estimate for L. Alternatively, the relationship between eigen-decomposition and SVD suggests that the left singular vectors of A are the same as the eigenvectors of the matrix AA^T, which equal to the eigenvectors of the column-column covariance matrix $\sum_{i=1}^{I} X_i X_i^T$. Note that the column-column covariance matrix plays a role similar to the sample covariance matrix in the single-matrix case. We comment that 2DSVD reduces the computational cost of the GLRAM procedure but does not directly minimize the objective function in (2.2). Hence it offers an approximation solution to the optimization problem (2.2).

Unfortunately, neither of GLRAM and 2DSVD is applicable for high-dimensional matrices. For example, in many neuroimaging studies, the dimension of each matrix and the number of subjects are usually very large. For the fMRI data example in Section 4.1, we have $m = 200,000$, $n = 200$ and $I = 100$. It follows that the sizes of the concatenated matrix A and the column-column covariance matrix $\sum_{i=1}^{I} X_i X_i^T$ are $200,000 \times 20,000$ and $200,000 \times 200,000$, which respectively require about 30 and 300 GB to store in double precision. Apparently, the enormous sizes of the matrices make their storage nearly impossible in general computers, and their SVD and eigen-decomposition computationally prohibitive.

The PVD approach of Crainiceanu et al. (2011) aims at overcoming these physical computer memory and computation limitations when dealing with massive data. Recall that 2DSVD computes the left singular vectors of the concatenated matrix A. Instead of combining all the raw data matrices, the idea of PVD is to only concatenate the dominating singular vectors of the individual matrices. This idea naturally makes PVD a two-step SVD approach. Formally, PVD first performs SVD on each individual matrix, denoted as $X_i = U_i D_i V_i^T$ where U_i is the $m \times r_i$ orthonormal left singular vector matrix, D_i is the $r_i \times r_i$ diagonal singular value matrix with the diagonal entries being the positive singular values in descending order, V_i is the $n \times r_i$

orthonormal right singular vector matrix, and r_i denotes the rank of X_i. Given the SVDs, only the first k_i^u and k_i^v columns of U_i and V_i are retained and denoted as $\widetilde{U}_i$ and $\widetilde{V}_i$ respectively. The matrices $\widetilde{U}_i$, $i = 1, \ldots, I$, are then concatenated as $P^* \equiv [\widetilde{U}_1, \widetilde{U}_2, \ldots, \widetilde{U}_I]$. In the second step, PVD estimates L with the first r_L left singular vectors of P^*. Similarly, PVD obtains the estimate for R with the first r_R left singular vectors of $Q^* \equiv [\widetilde{V}_1, \widetilde{V}_2, \ldots, \widetilde{V}_I]$.

PVD replaces one SVD on a single large matrix with many SVDs on relatively small matrices, and can efficiently reduce the computational cost. Coming back to the previous fMRI data example, the size of each individual matrix is $200,000 \times 200$ which requires about 0.3 GB memory and the corresponding SVD can be easily computed. If we retain the first 10 left singular vectors from each matrix, the aggregated matrix of the singular vectors is of size $200,000 \times 1,000$ which requires about 1.5 GB, and it is still feasible to obtain its SVD.

2.3 Adjusted population value decomposition

We propose to further improve PVD with one adjustment - proper incorporation of the singular values in D_i. Our motivation of the adjustment for PVD is from the fact that PVD gives equal weights to all singular vectors. As mentioned in Section 4.1, one important property of SVD is that the singular vectors of a matrix has a relative importance order which is reflected by their singular values. Taking this order information into consideration by scaling each singular vector with its corresponding singular value, it can potentially improve the estimation accuracy on the estimates of group components.

We present our adjusted population value decomposition (APVD) approach in this section. We first present the APVD algorithm in Section 2.3.1, and then discuss its memory and computation complexities in Sections 2.3.2 and 2.3.3, and its connections with PVD and 2DSVD in Section 2.3.4.

2.3.1 The APVD algorithm

Our APVD procedure modifies the PVD approach by taking into account the relative importance of the singular vectors in the group-level SVD step of PVD, while keeping the first SVD step intact.

The complete APVD procedure is presented below in Algorithm 2.1.

Algorithm 2.1 The APVD algorithm

1. Perform SVD on each X_i and obtain $X_i = U_i D_i V_i^T$.
   Let $\widetilde{U}_i$ and $\widetilde{V}_i$ denote the first k_i^u and k_i^v columns of U_i and V_i respectively.
   Let $\widetilde{D}_i^u$ and $\widetilde{D}_i^v$ be the corresponding diagonal matrices with the diagonal elements being the first k_i^u and k_i^v singular values of X_i respectively.
   Define $P \equiv [\widetilde{U}_1\widetilde{D}_1^u, \widetilde{U}_2\widetilde{D}_2^u, \ldots, \widetilde{U}_I\widetilde{D}_I^u]$ and $Q \equiv [\widetilde{V}_1\widetilde{D}_1^v, \widetilde{V}_2\widetilde{D}_2^v, \ldots, \widetilde{V}_I\widetilde{D}_I^v]$.
2. Perform SVD on P and Q.
   Obtain the APVD estimates of L and R as the first r_L and r_R left singular vectors of P and Q respectively.

Selection of k_i^u and k_i^v For both PVD and APVD, one needs to choose k_i^u and k_i^v in the individual-level SVD step. Our numerical experience suggests that as long as $k_i^u \geq r_L, k_i^v \geq r_R$, the specific choices of k_i^u and k_i^v do not matter that much. The condition that the number of the components preserved in the first step should be no smaller than the rank of the final estimator is intuitive, in that we must not throw away any *useful* information in each individual matrix. Since the smaller these two quantities k_i^u and k_i^v are, the less time-consuming the algorithm is, we recommend to set $k_i^u = r_L$ and $k_i^v = r_R$ in practice.

2.3.2 Memory complexity

We want to compare the amount of memory needed by each of the four methods: GLRAM, 2DSVD, PVD, and APVD. The four methods do not necessarily need to load all data matrices into memory at the same time except the SVD algorithm for 2DSVD, i.e. one can load one data matrix into memory at a time, perform the calculation and then remove it. Hence, we

compare the memory requirement using the largest matrix size needed to do SVD or eigen-decomposition for each approach.

For ease of illustration, we assume that, for PVD and APVD, the numbers of components kept in the second SVD step are the same across different subjects and equal the desired ranks, $k_i^u = r_L$, $k_i^v = r_R$, $i = 1, ..., I$. Furthermore, the ranks are assumed to be significantly smaller than the size of the original matrix, i.e. $r_L \ll n$, $r_R \ll m$, which is usually the case for dimension reduction to be useful. Without loss of generality, we assume that $m \geq n$.

GLRAM needs to do an eigen-decomposition of size $m \times m$ to obtain the estimate of L at each iteration. 2DSVD has two versions: the one that computes the SVDs of the concatenated matrix needs to do SVD on a matrix of size $m \times nI$ or $n \times mI$; the other one that computes the eigen-decomposition of the covariance matrices needs to calculate eigen-decomposition on a matrix of maximum size $m \times m$. As for PVD and APVD, the algorithms need to do SVDs on matrices of size $m \times n$ in the first step and of maximum size $m \times r_L I$ during the second step. By introducing the notations $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$, the maximum matrix size to do SVD for PVD and APVD is $m \times (n \vee (r_L I))$. The above comparison results are summarized in Table 2.1.

Maximum matrix size	APVD	PVD	2DSVD	GLRAM
for SVD	$m \times (n \vee (r_L I))$	$m \times (n \vee (r_L I))$	$m \times nI$ or $n \times mI$	
for eigen-decomposition			$m \times m$	$m \times m$

Table 2.1: The maximum matrix sizes for the four estimation approaches.

Back to the fMRI data example with $m = 200,000$, $n = 200$, $I = 100$, and $r_L = r_R = 10$ (for example). The matrices of size $m \times nI$ and $m \times m$ require 30 GB and 300 GB computer memory respectively. On the other hand, the matrix of size $m \times (n \vee (r_L I))$ only needs 1.5 GB memory. Hence, in this case, the 2DSVD and GLRAM approaches need much more memory than APVD and PVD.

2.3.3 Computation complexity

We compare the computational time complexities of these four algorithms as follows, which are summarized in Table 2.2.

- Suppose GLRAM converges after K iterations. During each iteration, one needs to compute $\sum_{i=1}^{I} X_i R R^T X_i^T$ and $\sum_{i=1}^{I} X_i^T L L^T X_i$, which takes $\mathcal{O}\left((m+n)(mr_R + nr_L)I\right)$ flops; the follow-up eigen-decomposition requires $\mathcal{O}\left(m^3 + n^3\right)$ flops. Hence, GLRAM can be computed in $\mathcal{O}\left(K(m+n)(mr_R + nr_L)I + Km^3 + Kn^3\right)$ flops.

- When 2DSVD computes the SVDs of $[X_1, X_2, \ldots, X_I]$ and $[X_1^T, X_2^T, \ldots, X_I^T]$ which are of size $m \times nI$ and $n \times mI$, the calculation takes $\mathcal{O}\left(mnI(m \wedge (nI) + n \wedge (mI))\right)$ flops. When 2DSVD computes the eigen-decompositions of $\sum_{i=1}^{I} X_i X_i^T$ and $\sum_{i=1}^{I} X_i^T X_i$, the matrix multiplication consumes $\mathcal{O}\left(m^2 nI + mn^2 I\right)$ and the eigen-decompositions takes $\mathcal{O}\left(m^3 + n^3\right)$ flops.

- The first step of PVD and APVD consists of I individual SVDs of matrices of size $m \times n$ and can be implemented in $\mathcal{O}\left(mn^2 I\right)$ flops. The second step involves two SVDs of matrices of size $m \times r_L I$ and $n \times r_R I$ and needs $\mathcal{O}\left(mr_L I(m \wedge (r_L I)) + nr_R I(n \wedge (r_R I))\right)$ flops.

	Computation Complexity
APVD	$\mathcal{O}\left(mn^2 I + mr_L I(m \wedge (r_L I)) + nr_R I(n \wedge (r_R I))\right)$
PVD	$\mathcal{O}\left(mn^2 I + mr_L I(m \wedge (r_L I)) + nr_R I(n \wedge (r_R I))\right)$
2DSVD	$\mathcal{O}\left(m^2 nI + mn^2 I + m^3 + n^3\right)$ or $\mathcal{O}\left(mnI(m \wedge (nI) + n \wedge (mI))\right)$
GLRAM	$\mathcal{O}\left(K(m+n)(mr_R + nr_L)I + Km^3 + Kn^3\right)$

Table 2.2: Comparison of computation complexity for the four approaches.

2.3.4 Connections of APVD with PVD and 2DSVD

GLRAM optimizes the least squares criterion but is computationally the most expensive approach. 2DSVD is less costly to compute and has been shown to be near-optimal (Ding and Ye, 2005). Both PVD and APVD attempt to overcome the memory limitation of GLRAM and 2DSVD through individual dimension reduction prior to SVD of the concatenated matrix.

In this section, we further investigate the connection between APVD and 2DSVD, as well as between APVD and PVD. We first show that 2DSVD and APVD produce similar estimates. Given the near-optimality of 2DSVD, it follows that APVD also possesses nice estimation property. We then prove that, under certain conditions, PVD and APVD recover the same subspace, although in most scenarios APVD estimates better.

APVD and 2DSVD Consider the SVD of X_i as $X_i = U_i D_i V_i^T$. Note that APVD concatenates the leading components of the scaled singular vectors $\{U_i D_i\}$, while 2DSVD concatenates the original data matrices $\{X_i\}$. The following Proposition 1 suggests that if APVD concatenates the full set of the scaled singular vectors from each subject, then APVD and 2DSVD give the same estimates for L and R in Model (2.1).

Proposition 1 *The concatenated data matrix* $[X_1, X_2, \ldots, X_I]$ *and the concatenated scaled singular vector matrix* $[U_1 D_1, U_2 D_2, \ldots, U_I D_I]$ *have the same set of left singular vectors and the singular values. Similarly,* $[X_1^T, X_2^T, \ldots, X_I^T]$ *and* $[V_1 D_1, V_2 D_2, \ldots, V_I D_I]$ *have the same set of left singular vectors and singular values.*

Given the above equivalence, when APVD only concatenates the first k_i^u scaled singular vectors, instead of the full set, the estimates from APVD are not exactly the same as those from 2DSVD, but they are close as we show below. 2DSVD computes the left singular vectors of $[X_1, X_2, \ldots, X_I]$, which are the eigenvectors of $\sum_{i=1}^I X_i X_i^T$. On the other hand, APVD calculates the left singular vectors of $[\widetilde{U}_1 \widetilde{D}_1^u, \widetilde{U}_2 \widetilde{D}_2^u, \ldots, \widetilde{U}_I \widetilde{D}_I^u]$, which are the eigenvectors of $\sum_{i=1}^I \widetilde{U}_i (\widetilde{D}_i^u)^2 \widetilde{U}_i^T$. Since $X_i X_i^T = U_i D_i^2 U_i^T \approx \widetilde{U}_i (\widetilde{D}_i^u)^2 \widetilde{U}_i^T$, we obtain $\widehat{L^{APVD}} \approx \widehat{L^{2DSVD}}$.

APVD and PVD We remind that $P^* \equiv [\widetilde{U}_1, \widetilde{U}_2, \ldots, \widetilde{U}_I]$, $Q^* \equiv [\widetilde{V}_1, \widetilde{V}_2, \ldots, \widetilde{V}_I]$, $P \equiv [\widetilde{U}_1\widetilde{D}_1^u, \widetilde{U}_2\widetilde{D}_2^u, \ldots, \widetilde{U}_I\widetilde{D}_I^u]$, and $Q \equiv [\widetilde{V}_1\widetilde{D}_1^v, \widetilde{V}_2\widetilde{D}_2^v, \ldots, \widetilde{V}_I\widetilde{D}_I^v]$. The second step of PVD calculates SVD of P^* and Q^*, while APVD does SVD on P and Q.

The PVD procedure concatenates the singular vectors while APVD concatenates the scaled singular vectors. If every singular vector in the aggregated matrix of PVD has a corresponding scaled vector in APVD, the column spaces of the two aggregated matrices would be the same. On the other hand, a fundamental property of SVD is that the left singular vectors corresponding to the non-zero singular values represent an orthonormal basis of the matrix column space. Therefore, the full set of the left singular vectors with non-zero singular values of the two aggregated matrices are basis representing the same subspace. Hence, there exists an orthogonal transformation relationship between the two full sets of singular vectors with non-zero singular values. We summarize the relationship between APVD and PVD in the following proposition.

Proposition 2 *The ranks of P and P^* are the same and we denote it by r_P. Moreover, the first r_P left singular vectors of P^* are orthogonal rotations of the first r_P left singular vectors of P. Similar results hold for Q and Q^*.*

Proposition 2 shows the orthogonal transformation relationship between APVD and PVD only if $r_L = r_P$. In practice, r_L is usually smaller than r_P since we take k_i^u greater than or equal to r_L. Hence, in most cases, there does not exist an orthogonal transformation relationship between the estimates by PVD and APVD. Simulation results in Section 2.5 show that APVD outperforms PVD measured by subspace recovery and normalized reconstruction errors.

2.4 Theoretical properties

In this section, we study some theoretical properties of APVD regarding normalized reconstruction error which is a measure of accuracy of the proposed procedure. We will show that the normalized reconstruction error is bounded from above by the normalized information lost

of the two-step SVD for both one-side and two-side type approximations. Before we proceed to the main results, we first introduce some notations.

In both SVD steps of the APVD approach, we only retain the first few singular vectors and singular values which can explain most of the variance. The amount of information kept or lost in each step can be characterized by the singular values or eigenvalues. For the first SVD step, let

$$\theta^u = \min_{i\in\{1,\ldots,I\}} \frac{\sum_{j=1}^{k_i^u} d_{ij}^2}{\sum_{j=1}^{r_i} d_{ij}^2}, \quad \text{and} \quad \theta^v = \min_{i\in\{1,\ldots,I\}} \frac{\sum_{j=1}^{k_i^v} d_{ij}^2}{\sum_{j=1}^{r_i} d_{ij}^2}, \tag{2.3}$$

where d_{ij} is the jth singular value of X_i in descending order. Then θ^u and θ^v denote the minimum fraction of variances kept from individual matrices.

Recall that P and Q are defined as

$$P \equiv [\widetilde{U}_1\widetilde{D}_1^u, \widetilde{U}_2\widetilde{D}_2^u, \ldots, \widetilde{U}_I\widetilde{D}_I^u], \quad \text{and} \quad Q \equiv [\widetilde{V}_1\widetilde{D}_1^v, \widetilde{V}_2\widetilde{D}_2^v, \ldots, \widetilde{V}_I\widetilde{D}_I^v]$$

which are the aggregation matrices of the scaled singular vectors. Let d_{Pj} and d_{Qj} be the jth singular values of P and Q in descending order, respectively. Similar to θ^u and θ^v, we define

$$\theta^P = \frac{\sum_{j=1}^{r_L} d_{Pj}^2}{\sum_{j=1}^{r_P} d_{Pj}^2}, \quad \text{and} \quad \theta^Q = \frac{\sum_{j=1}^{r_R} d_{Qj}^2}{\sum_{j=1}^{r_Q} d_{Qj}^2}, \tag{2.4}$$

which represent the information we retain in the second SVD step.

The normalized reconstruction error r is a commonly used metric to measure and compare the performances of various approaches. It is defined as

$$r = \frac{\sum_{i=1}^{I} \|X_i - \widehat{X}_i\|_F^2}{\sum_{i=1}^{I} \|X_i\|_F^2}, \tag{2.5}$$

where $\widehat{X}_i$ denotes the reconstruction of X_i. We present the upper bound of r for the APVD approach in terms of θ^P, θ^Q, θ^u and θ^v in the following theorems.

2.4.1 Upper bound for the two-side type approximation

The following theorem specifies how the normalized reconstruction error of the proposed APVD approach is bounded from above by the normalized information lost in the two SVD steps.

Theorem 1 *Consider Model (2.1) and assume that the estimates of L and R are given by the APVD procedure. Then the upper bound of the normalized reconstruction error r is*

$$r \leq (1 - \theta^u \theta^P) + (1 - \theta^v \theta^Q).$$

From the definitions, we know that θ^u denotes the minimum fraction of variances retained in the first SVD step, and θ^P represents the fraction of information kept in the second SVD step for estimating L. Hence, the fraction of variances kept in the two SVD steps is at least $\theta^u \theta^P$. It follows that the normalized information lost is at most $1 - \theta^u \theta^P$ for estimating L. Similarly, $1 - \theta^v \theta^Q$ is the largest normalized information lost for estimating R. Theorem 1 shows that the normalized reconstruction error is bounded from above by the sum of the maximum fraction of variances lost while estimating the left and right components.

2.4.2 Upper bound for the one-side type approximation

We comment here that our APVD procedure can also be used to estimate group components of the 2DPCA model (Yang et al., 2004) which is a one-side type low rank approximation of matrices. To be more specific, 2DPCA approximates each $m \times n$ matrix X_i, $i = 1, \ldots, I$, by a product of one group-specific right component R^{2DPCA} of size $n \times r_R$ and one subject-specific term W_i^{2DPCA} of size $m \times r_R$, i.e.,

$$X_i = W_i^{\mathrm{2DPCA}} (R^{\mathrm{2DPCA}})^T + E_i, \tag{2.6}$$

where E_i of size $m \times n$ is the error matrix.

The APVD estimation procedures of L and R in the two-side type model (2.1) are seperate processes. Hence, the estimate of R obtained by APVD can also be used as an estimate of R^{2DPCA} in Model (2.6). The subject-specific term W_i^{2DPCA} can be obtained via $\widehat{W_i^{\mathrm{2DPCA}}} = X_i\widehat{R^{\mathrm{2DPCA}}}$. It follows that X_i can be reconstructed by $\widehat{X_i} = \widehat{W_i^{\mathrm{2DPCA}}}(\widehat{R^{\mathrm{2DPCA}}})^T$. The upper bound of the normalized reconstruction error under the 2DPCA model (2.6) is given by the following theorem.

Theorem 2 *Consider model (2.6) and assume the estimate of R^{2DPCA} is given by the APVD procedure. Then the upper bound of the normalized reconstruction error r is*

$$r \leq 1 - \theta^v \theta^Q.$$

Similar to the interpretation of Theorem 1, Theorem 2 shows that the normalized reconstruction error is bounded from above by the maximum fraction of variances lost when estimating the right group component.

2.5 Numerical studies

We evaluate the performance of our proposed APVD approach through simulations in Sections 2.5.1 and 2.5.2. The simulation results show that APVD performs comparable to GLRAM and 2DSVD in terms of estimation accuracy and all three methods are more accurate than PVD. We then apply the methods to a well-known face image database in Section 2.5.3.

We compare the performance of the methods in terms of subspace recovery, normalized reconstruction errors, and computational time. Since the columns of L and $\hat{L}$ form two sets of orthonormal basis, we use the following metric to measure the discrepancy between the corresponding subspaces:

$$D(\hat{L}, L) = \|\hat{L}\hat{L}^T - LL^T\|_2, \tag{2.7}$$

where $\|\cdot\|_2$ denotes the matrix spectral norm. This distance metric equals the sine of the largest canonical angle between two subspaces, and has been used by Golub and Van Loan (1996) among others in the dimension reduction literature. We note that $0 \leq D(\hat{L}, L) \leq 1$ with a smaller value corresponding to a better estimate. Similarly, we can define the distance for the right group component as

$$D(\hat{R}, R) = \|\hat{R}\hat{R}^T - RR^T\|_2. \tag{2.8}$$

2.5.1 Simulation: subspace recovery and normalized reconstruction errors

Data are simulated according to Model (2.1):

$$X_i = LW_iR^T + E_i,$$

for $i = 1, 2, \ldots, I$ with $I = 10$, $r_L = 10$ and $r_R = 6$. The jth column of L (and R) is $(0, 0, \ldots, 0, 1, 0, \ldots, 0)^T$ with the jth entry being 1 and the others being 0. Each entry of the individual component W_i is i.i.d. $N(0, 1)$, and each entry of the error matrix E_i is i.i.d. $N(0, \sigma^2)$ where σ is the noise level determined as follows. We use $r_L r_R/(mn\sigma^2)$ to approximate the signal to noise ratio (SNR). Then σ is given by $\sqrt{r_L r_R/(mn \cdot \mathrm{SNR})}$. In all simulations, we consider SNR as 2 to calculate the corresponding value of σ.

For 2DSVD and GLRAM, we take the first $r_L = 10$ and $r_R = 6$ leading eigenvectors of the corresponding matrices as the estimates of L and R. For APVD and PVD, we take $k_i^u = 10$ and $k_i^v = 6$ in the first-step SVD and $r_L = 10$ and $r_R = 6$ in the second-step SVD.

Figure 2.1 shows the results for $m = 100$ and $n = 50$ over 100 simulation replications. Panels (a) and (b) depict the boxplots of the distances between the estimated subspaces and the underlying truth for L and R, respectively. Panel (c) compares the boxplots of the normalized reconstruction errors r (2.5). According to all three measures, APVD achieves comparable performance as the near-optimal 2DSVD and the optimal GLRAM, all of which outperform

PVD.

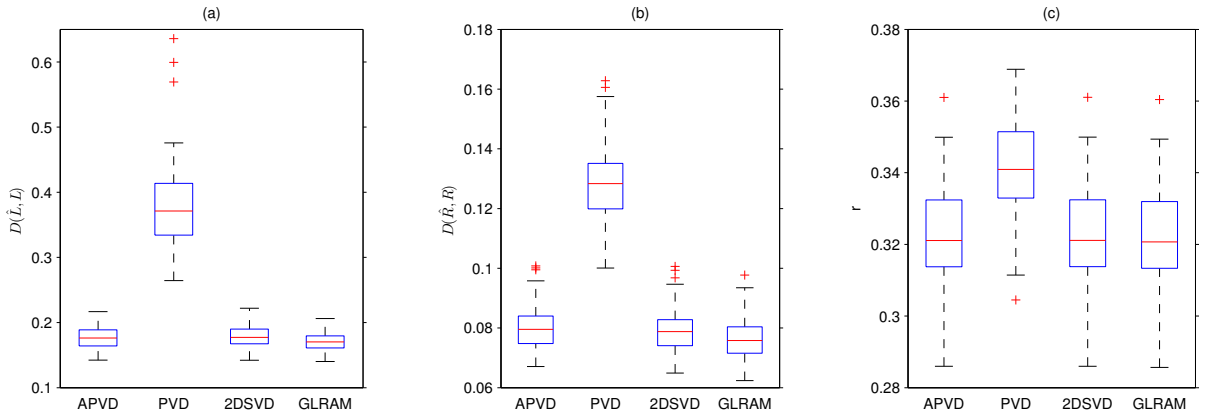


Figure 2.1: Boxplots comparing four approaches for subspace recovery and normalized reconstruction errors over 100 simulation runs for $m = 100$ and $n = 50$. (a) Distances between the subspaces spanned by $\hat{L}$ and L defined in (2.7). (b) Distances between the subspaces spanned by $\hat{R}$ and R defined in equation (2.8). (c) Normalized reconstruction errors r defined in (2.5).

Furthermore, Table 2.3 compares the average results for four (m, n) paris over 100 runs: $(100, 20)$, $(100, 50)$, $(500, 100)$ and $(500, 250)$. We can see that GLRAM achieves the best performance in all three measures, APVD performs comparable to 2DSVD and outperforms PVD. In terms of computing time, 2DSVD is the fastest one, while APVD and PVD have nearly the same speed, both of which are faster than the iterative GLRAM.

m	n	Approach	$D(\hat{L}, L)$	$D(\hat{R}, R)$	r	Time
100	20	APVD	0.276 (0.030)	0.086 (0.012)	0.306 (0.012)	0.009 (0.001)
		PVD	0.502 (0.094)	0.147 (0.023)	0.335 (0.014)	0.009 (0.003)
		2DSVD	0.278 (0.030)	0.083 (0.011)	0.306 (0.012)	0.003 (0.000)
		GLRAM	0.267 (0.028)	0.078 (0.010)	0.305 (0.012)	0.038 (0.001)
100	50	APVD	0.177 (0.017)	0.080 (0.007)	0.322 (0.014)	0.020 (0.001)
		PVD	0.380 (0.063)	0.129 (0.014)	0.342 (0.014)	0.019 (0.000)
		2DSVD	0.179 (0.018)	0.079 (0.007)	0.322 (0.014)	0.007 (0.015)
		GLRAM	0.171 (0.015)	0.076 (0.007)	0.322 (0.014)	0.054 (0.002)
500	100	APVD	0.120 (0.010)	0.034 (0.003)	0.328 (0.013)	0.216 (0.009)
		PVD	0.213 (0.025)	0.067 (0.010)	0.334 (0.013)	0.215 (0.009)
		2DSVD	0.120 (0.011)	0.034 (0.003)	0.328 (0.013)	0.199 (0.007)
		GLRAM	0.119 (0.010)	0.034 (0.003)	0.328 (0.013)	1.750 (0.077)
500	250	APVD	0.076 (0.007)	0.033 (0.002)	0.333 (0.013)	0.737 (0.026)
		PVD	0.162 (0.020)	0.063 (0.009)	0.337 (0.013)	0.738 (0.028)
		2DSVD	0.076 (0.007)	0.033 (0.002)	0.333 (0.013)	0.343 (0.013)
		GLRAM	0.075 (0.007)	0.033 (0.002)	0.333 (0.013)	2.613 (0.106)

Table 2.3: Average results for four (m, n) pairs based on 100 simulation runs. Standard errors are shown in parentheses.

The above simulation results offer great support for APVD (over PVD). It performs compa-

rable with 2DSVD and GLRAM in these small to moderate simulation studies. When GLRAM and 2DSVD can not be computed for high-dimensional matrices, we expect that APVD still outperforms PVD.

2.5.2 Simulation: choices of r_L, r_R, k_i^u and k_i^v

In this simulation, we study how different choices of r_L, r_R, k_i^u and k_i^v can affect the estimation of L and R. Data are simulated according to Model (2.1) with $I = 10$, $m = 100$ and $n = 50$. The true rank of L and R are chosen to be 15. The matrices L, R, W_i and E_i are simulated in the same way as in Section 2.5.1. We replicate the simulation 100 times.

For APVD and PVD, we choose the same number of singular vectors in the first SVD step for each individual matrix, i.e. $k_i^u = k^u$ and $k_i^v = k^v$ for all i. Note that r_L and r_R represent the numbers of columns of $\hat{L}$ and $\hat{R}$ respectively. We study how the number of components affects the normalized reconstruction error by letting (1) the four parameters equal and vary together; (2) k^u and k^v are fixed at 20 while r_L and r_R change; and (3) r_L and r_R are fixed at 5 while k^u and k^v vary. The results are shown in Figure 2.2.

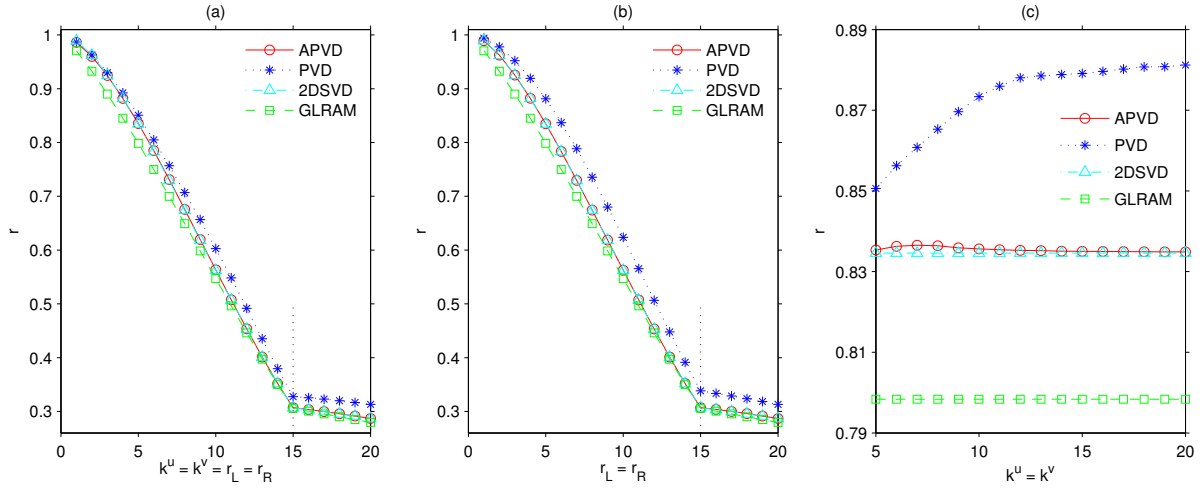


Figure 2.2: The choices of the numbers of singular vectors. (a) $k^u = k^v = r_L = r_R$ and vary from 1 to 20. (b) $k^u = k^v = 20$ and $r_L = r_R$ vary from 1 to 20. (c) $r_L = r_R = 5$ and $k^u = k^v$ vary from 5 to 20. The dotted vertical lines represent the locations of the true ranks of L and R.

Figure 2.2(a) studies the question of how many components we need to choose. We take

$k^u = k^v = r_L = r_R$ and vary them together from 1 to 20. Recall that the true ranks of L and R are 15. As the numbers of columns of $\hat{L}$ and $\hat{R}$ (r_L and r_R) increase, the more components are used to approximate the data matrices and hence the normalized reconstruction errors decrease for all four methods. We note that the errors decrease quickly towards the true rank 15 and slowly after 15. Moreover, APVD, 2DSVD and GLRAM have comparable normalized reconstruction errors and are smaller than the PVD approach for all cases.

Figure 2.2(b) investigates how the number of components chosen in the second SVD step will affect the final estimates. We set $k^u = k^v = 20$ and let $r_L = r_R$ vary from 1 to 20. As the number of components increases, the errors decreases. As in Figure 2.2(a), there is a change of decreasing rate at the true rank 15. We can see that APVD has comparable performance as 2DSVD and GLRAM and outperforms PVD.

Figure 2.2(c) studies the relation between the number of components chosen in the first SVD step and the final estimates. We set $r_L = r_R = 5$ and let $k^u = k^v$ change from 5 to 20. Here the numbers of the first SVD step (k^u and k^v) are chosen to be greater than or equal to the numbers of the second SVD step (r_L and r_R). This is reasonable since we should preserve information in each matrix. We note the following observations: (1) the error of the PVD procedure goes up ; (2) the normalized error of APVD changes little as k^u and k^v increase; and (3) APVD is very close to 2DSVD. These observations show the advantages of incorporating the singular values in the analysis. Below we provide intuitive explanations for the observations.

We illustrate (1) and (2) using the left group component L. As k^u increases, new singular vectors or scaled singular vectors from individual subjects are gradually added into the aggregation matrices. For the PVD method, since the newly added singular vectors are treated equally to the previous ones, the leading eigenspace of the aggregation matrix will be affected by these new unit vectors. However, if the new vectors are scaled and the scales are relatively small compared to the previous ones, the principal eigenspace is still dominated by the previous

scaled singular vectors. Hence, the eigenspace will remain stable as k^u increases. These explain why the normalized errors of the PVD procedure go up but those of the APVD algorithm only vary slightly. As for (3), recall from Section 2.3.4 and Proposition 1, the estimates from 2DSVD are the same as the APVD ones when we take the full sets of individual scaled singular vectors. From (2), we know that adding more individual scaled singular vectors will only slightly vary the leading eigenspace. Hence, the normalized reconstruction errors of APVD are very close to those of 2DSVD.

2.5.3 Application to the AT$\&$T Database of Faces

We apply the four methods (GLRAM, 2DSVD, PVD, APVD) to the well-known AT&T Database of Faces (Samaria and Harter, 1994), and demonstrate their real applicabilities. The database contains 400 gray-scale images of faces of size 92×112 from 40 individuals. There are 10 face images per individual.

For each individual subject, let $X_i \in \mathbb{R}^{92\times 112}$ denote the ith image, $i = 1, 2, \ldots, 10$. We first subtract the mean image $\overline{X} = \sum_{i=1}^{10} X_i/10$ from each image, and consider the following model for each individual:

$$X_i - \overline{X} = LW_iR^T + E_i.$$

We then apply the four methods to estimate the model, and obtain the reconstruction for each image as

$$\widehat{X_i} = \widehat{L}\widehat{L}^T(X_i - \overline{X})\widehat{R}\widehat{R}^T + \overline{X}.$$

Here, we take $k^u = k^v = r_L = r_R = 20$. We perform the model fitting and reconstruction separately for each of the 40 subjects.

For a particular subject, Figure 2.3 shows the original face images, and the reconstruction results from each method. The first row contains the original images, while the second to the fifth rows are the reconstructed images given by APVD, PVD, 2DSVD and GLRAM, respectively. As one can see, the reconstructed images given by APVD are visually very similar to

the ones from 2DSVD and GLRAM, all of which are better than the PVD reconstructions and capture more finer details.

We then randomly choose 10 representative subjects and select 1 representative image for each subject. The images and the reconstructions are shown in Figure 2.4. We can make the same observations as in Figure 2.3 that APVD gives finer reconstructions than PVD, and works comparably with 2DSVD and GLRAM.

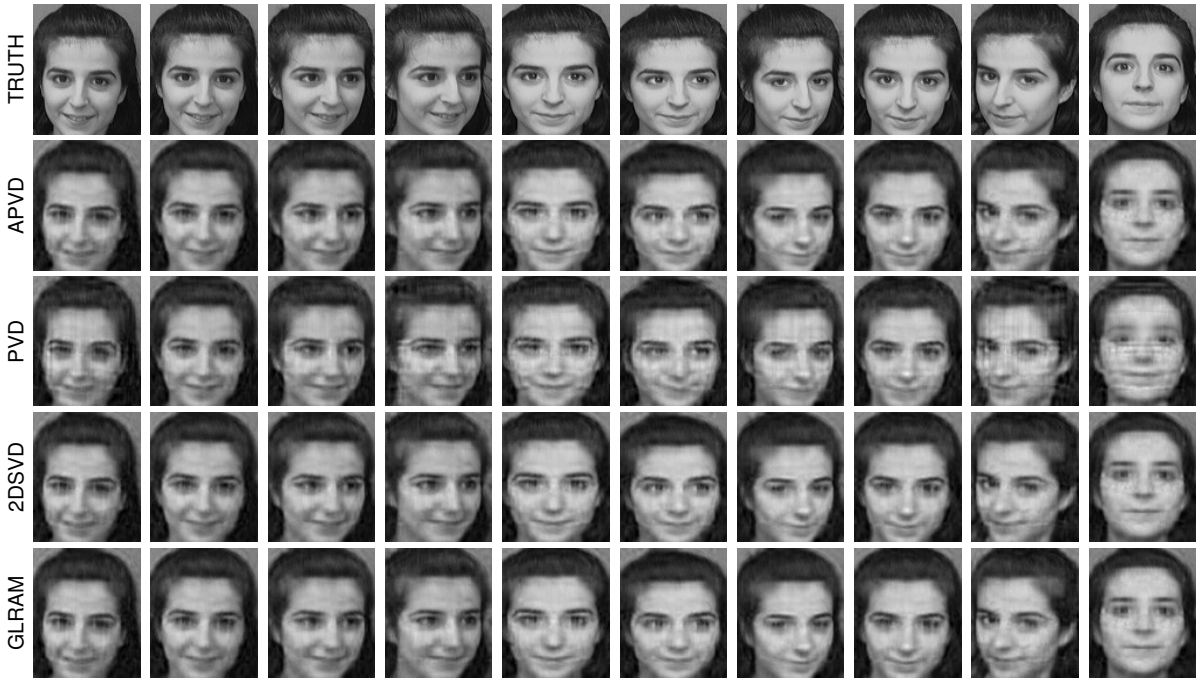


Figure 2.3: One particular subject: the true images (first row) and the reconstructed images by APVD (second row), PVD (third row), 2DSVD (fourth row), and GLRAM (fifth row).

2.6 Conclusions

We consider the problem of dimension reduction for groups of matrix-valued data, motivated by analysis of high-dimensional neuroimaging data. We develop a computationally efficient method - *adjusted population value decomposition* (APVD) that requires significantly less memory than most of the existing methods. Our method performs comparably with the state of the art algorithms such as GLRAM and 2DSVD when they can be computed for ma-



Figure 2.4: Ten representative subjects with one representative image per subject.

trices of small-to-moderate sizes, and improves the performance of PVD by a considerable amount but maintains the nice property of PVD that it requires little storage space. Furthermore, we establish the error bound of APVD theoretically and demonstrate its superior performance numerically.

2.7 Proofs

Before we proceed to the proofs, we introduce some notations. Let u_{ij} and v_{ij} denote the jth column of U_i and V_i respectively for $j = 1, \ldots, r_i$. We define P and Q as the aggregated matrices $[\widetilde{U}_1\widetilde{D}_1^u, \widetilde{U}_2\widetilde{D}_2^u, \ldots, \widetilde{U}_I\widetilde{D}_I^u]$ and $[\widetilde{V}_1\widetilde{D}_1^v, \widetilde{V}_2\widetilde{D}_2^v, \ldots, \widetilde{V}_I\widetilde{D}_I^v]$. Write $M = PP^T$ and $N = QQ^T$. Let the eigen-decomposition of M be

$$M = U_M \Lambda_M U_M^T,$$

where U_M is an m by r_M matrix with orthonormal columns, Λ_M is an r_M by r_M diagonal matrix with diagonal entries $\lambda_{M1} \geq \lambda_{M2} \geq \ldots \geq \lambda_{Mr_M} > 0$, and r_M denotes the rank of M. Similarly, we can define the eigen-decomposition of N by

$$N = V_N \Lambda_N V_N^T,$$

with an orthonormal matrix V_N of size $n \times r_N$ and a diagonal matrix Λ_N with diagonal entries $\lambda_{N1} \geq \lambda_{N2} \geq \ldots \geq \lambda_{Nr_N} > 0$.

From the relation between eigen-decomposition and SVD, we know that the eigenvectors of M and the left singular vectors of P are the same, and $\lambda_{Mj} = d_{Pj}^2$. Similarly, the eigenvectors of N and the left singular vectors of Q are the same, and $\lambda_{Nj} = d_{Qj}^2$. Hence, the estimates of the APVD procedure can be obtained from the eigenvectors of M and N.

To be more specific, let $\widetilde{U}_M$ and $\widetilde{V}_N$ consist of the first r_L and r_R columns of U_M and V_N respectively. Then $\widetilde{U}_M$ and $\widetilde{V}_N$ are the final estimates of L and R by the APVD approach. For the rest of this chapter, we will use the eigenvectors and eigenvalues of M and N instead of the left singular vectors of P and Q. Moreover, we define the following quantities

$$\theta^M = \frac{\sum_{t=1}^{r_L} \lambda_{Mt}}{\sum_{t=1}^{r_M} \lambda_{Mt}}, \quad \text{and} \quad \theta^N = \frac{\sum_{t=1}^{r_R} \lambda_{Nt}}{\sum_{t=1}^{r_N} \lambda_{Nt}}, \tag{2.9}$$

where θ^M and θ^N represent the normalized information retained in the second SVD step. Then we have $\theta^M = \theta^P$ and $\theta^N = \theta^Q$.

Proof of Proposition 1. Let F denote the aggregated data matrix $[X_1, X_2, \ldots, X_I]$ and G denote the concatenated matrix $[U_1 D_1^u, U_2 D_2^u, \ldots, U_I D_I^u]$. Firstly we note that the left singular vectors of F and G are the same as the eigenvectors of FF^T and GG^T. The singular values of F and G are the square roots of the corresponding eigenvalues of FF^T and GG^T. Hence we

only need to show that $FF^T = GG^T$. Observe that

$$FF^T = \sum_{i=1}^{I} X_i X_i^T = \sum_{i=1}^{I} \sum_{j=1}^{r_i} d_{ij}^2 u_{ij} u_{ij}^T,$$

and

$$GG^T = \sum_{i=1}^{I} U_i D_i^u D_i^u U_i^T = \sum_{i=1}^{I} \sum_{j=1}^{r_i} d_{ij}^2 u_{ij} u_{ij}^T.$$

Then we have

$$FF^T = GG^T.$$

Proof of Theorem 2. We can express M and N in terms of the singular vectors and the singular values of the individual matrices as follows:

$$M = PP^T = \sum_{i=1}^{I} \sum_{j=1}^{k_i^u} d_{ij}^2 u_{ij} u_{ij}^T, \tag{2.10}$$

$$N = QQ^T = \sum_{i=1}^{I} \sum_{j=1}^{k_i^v} d_{ij}^2 v_{ij} v_{ij}^T. \tag{2.11}$$

The estimate $\hat{L}$ is given by $\widetilde{U}_M$. We first give a lower bound on the sum of squares of the reconstructions $\alpha = \sum_{i=1}^{I} \|\widetilde{U}_M \widetilde{U}_M^T X_i\|_F^2$. It is clear that

$$\alpha = \sum_{i=1}^{I} \mathrm{tr}(\widetilde{U}_M \widetilde{U}_M^T X_i X_i^T \widetilde{U}_M \widetilde{U}_M^T) = \sum_{i=1}^{I} \mathrm{tr}(\widetilde{U}_M^T X_i X_i^T \widetilde{U}_M),$$

where $\mathrm{tr}(\cdot)$ denote the trace of a square matrix. The second equality holds because the columns of $\widetilde{U}_M$ are orthonormal. If we exchange the order of the trace and sum operators and write $X_i X_i^T$ in terms of the singular vectors and the singular values of X_i, we have

$$\alpha = \mathrm{tr}\left(\widetilde{U}_M^T (\sum_{i=1}^{I} X_i X_i^T) \widetilde{U}_M\right) = \mathrm{tr}\left(\widetilde{U}_M^T (\sum_{i=1}^{I} \sum_{j=1}^{r_i} d_{ij}^2 u_{ij} u_{ij}^T) \widetilde{U}_M\right).$$

Partitioning the first k_i^u singular vectors and singular values into one group and the rest into the other group for each matrix X_i yields

$$\alpha = \operatorname{tr}\left(\widetilde{U}_M^T(\sum_{i=1}^{I}\sum_{j=1}^{k_i^u} d_{ij}^2 u_{ij}u_{ij}^T)\widetilde{U}_M\right) + \operatorname{tr}\left(\widetilde{U}_M^T(\sum_{i=1}^{I}\sum_{j=k_i^u+1}^{r_i} d_{ij}^2 u_{ij}u_{ij}^T)\widetilde{U}_M\right).$$

The matrix $\sum_{i=1}^{I}\sum_{j=k_i^u+1}^{r_i} d_{ij}^2 u_{ij}u_{ij}^T$ is positive semidefinite. Then, it follows that

$$\operatorname{tr}\left(\widetilde{U}_M^T(\sum_{i=1}^{I}\sum_{j=k_i^u+1}^{r_i} d_{ij}^2 u_{ij}u_{ij}^T)\widetilde{U}_M\right) \geq 0,$$

and hence

$$\alpha \geq \operatorname{tr}\left(\widetilde{U}_M^T(\sum_{i=1}^{I}\sum_{j=1}^{k_i^u} d_{ij}^2 u_{ij}u_{ij}^T)\widetilde{U}_M\right).$$

Note that $\sum_{i=1}^{I}\sum_{j=1}^{k_i^u} d_{ij}^2 u_{ij}u_{ij}^T$ is M by equation (2.10). Together with equation (2.9), we have

$$\alpha \geq \operatorname{tr}(\widetilde{U}_M^T M \widetilde{U}_M) = \sum_{t=1}^{r_L}\lambda_{Mt} = \theta^M \sum_{t=1}^{r_M}\lambda_{Mt}.$$

We note that $\sum_{t=1}^{r_M}\lambda_{Mt} = \operatorname{tr}(M)$ and by (2.10), we have

$$\operatorname{tr}(M) = \operatorname{tr}(\sum_{i=1}^{I}\sum_{j=1}^{k_i^u} d_{ij}^2 u_{ij}u_{ij}^T) = \sum_{i=1}^{I}\sum_{j=1}^{k_i^u} d_{ij}^2.$$

Together with the definition of θ^u in equation (2.3), it follows that

$$\alpha \geq \theta^M\theta^u \sum_{i=1}^{I}\sum_{j=1}^{r_i} d_{ij}^2.$$

The Frobenius norm of each data matrix is connected with its singular values, which leads to $\|X_i\|_F^2 = \sum_{j=1}^{r_i} d_{ij}^2$. Thus,

$$\alpha \geq \theta^M\theta^u \|X_i\|_F^2.$$

Now we can establish the upper bound for the normalized reconstruction error r through α as follows.

$$\frac{\sum_{i=1}^{I}\|X_i-\widetilde{U}_M\widetilde{U}_M^TX_i\|_F^2}{\sum_{i=1}^{I}\|X_i\|_F^2}=1-\frac{\sum_{i=1}^{I}\|\widetilde{U}_M\widetilde{U}_M^TX_i\|_F^2}{\sum_{i=1}^{I}\|X_i\|_F^2}\leq 1-\theta^u\theta^M=1-\theta^u\theta^P.$$

Similarly, we can prove that

$$\frac{\sum_{i=1}^{I}\|X_i-X_i\widetilde{V}_N\widetilde{V}_N^T\|_F^2}{\sum_{i=1}^{I}\|X_i\|_F^2}=1-\frac{\sum_{i=1}^{I}\|X_i\widetilde{V}_N\widetilde{V}_N^T\|_F^2}{\sum_{i=1}^{I}\|X_i\|_F^2}\leq 1-\theta^v\theta^N=1-\theta^v\theta^Q.$$

Proof of Theorem 1. Let

$$\alpha=\sum_{i=1}^{I}\|\widetilde{U}_M\widetilde{U}_M^TX_i\|_F^2,\quad \text{and}\quad \beta=\sum_{i=1}^{I}\|X_i\widetilde{V}_N\widetilde{V}_N^T\|_F^2.$$

Then from the proof of Theorem 2, we know that

$$\alpha=\sum_{i=1}^{I}\|\widetilde{U}_M\widetilde{U}_M^TX_i\|_F^2=\sum_{i=1}^{I}\|\widetilde{U}_M^TX_i\|_F^2\geq\theta^M\theta^u\sum_{i=1}^{I}\|X_i\|_F^2, \tag{2.12}$$

and

$$\beta=\sum_{i=1}^{I}\|X_i\widetilde{V}_N\widetilde{V}_N^T\|_F^2=\sum_{i=1}^{I}\|X_i\widetilde{V}_N\|_F^2\geq\theta^N\theta^v\sum_{i=1}^{I}\|X_i\|_F^2. \tag{2.13}$$

Let $\widetilde{U}_M^c\in\mathbb{R}^{m\times(m-r_L)}$ and $\widetilde{V}_N^c\in\mathbb{R}^{n\times(n-r_R)}$ be two orthonormal matrices such that

$$\widetilde{U}_M\widetilde{U}_M^T+\widetilde{U}_M^c\widetilde{U}_M^{cT}=I_{m\times m},\quad \text{and}\quad \widetilde{V}_N\widetilde{V}_N^T+\widetilde{V}_N^c\widetilde{V}_N^{cT}=I_{n\times n}.$$

Then the reconstruction error is given by

$$\sum_{i=1}^{I}\|X_i-\widetilde{U}_M\widetilde{U}_M^TX_i\widetilde{V}_N\widetilde{V}_N^T\|_F^2=\sum_{i=1}^{I}\mathrm{tr}(\widetilde{U}_M\widetilde{U}_M^TX_i\widetilde{V}_N^c\widetilde{V}_N^{cT}X_i^T+\widetilde{U}_M^c\widetilde{U}_M^{cT}X_iX_i^T).$$

Since the matrix $\widetilde{U}_M^{cT} X_i \widetilde{V}_N^c \widetilde{V}_N^{cT} X_i^T \widetilde{U}_M^c$ is positive semidefinite, we have

$$\mathrm{tr}(\widetilde{U}_M^{cT} X_i \widetilde{V}_N^c \widetilde{V}_N^{cT} X_i^T \widetilde{U}_M^c) \geq 0.$$

Thus,

$$\begin{aligned}
& \sum_{i=1}^{I} \|X_i - \widetilde{U}_M \widetilde{U}_M^T X_i \widetilde{V}_N \widetilde{V}_N^T\|_F^2 \\
\leq & \sum_{i=1}^{I} \mathrm{tr}\left(\widetilde{U}_M \widetilde{U}_M^T X_i \widetilde{V}_N^c \widetilde{V}_N^{cT} X_i^T + \widetilde{U}_M^c \widetilde{U}_M^{cT} X_i X_i^T + \widetilde{U}_M^c \widetilde{U}_M^{cT} X_i \widetilde{V}_N^c \widetilde{V}_N^{cT} X_i^T\right) \\
= & \sum_{i=1}^{I} \left(\|X_i \widetilde{V}_N^c\|_F^2 + \|\widetilde{U}_M^{cT} X_i\|_F^2\right).
\end{aligned}$$

By the following equalities

$$\|\widetilde{U}_M^{cT} X_i\|_F^2 = \|X_i\|_F^2 - \|\widetilde{U}_M^T X_i\|_F^2, \quad \text{and} \quad \|X_i \widetilde{V}_N^c\|_F^2 = \|X_i\|_F^2 - \|X_i \widetilde{V}_N\|_F^2,$$

and together with (2.12) and (2.13), we can establish the upper bound for the normalized reconstruction error as follows.

$$\begin{aligned}
& \frac{\sum_{i=1}^{I} \|X_i - \tilde{U}_M \tilde{U}_M^T X_i \widetilde{V}_N \widetilde{V}_N^T\|_F^2}{\sum_{i=1}^{I} \|X_i\|_F^2} \\
\leq & \frac{\sum_{i=1}^{I} \left(\|X_i\|_F^2 - \|X_i \widetilde{V}_N\|_F^2\right) + \sum_{i=1}^{I} \left(\|X_i\|_F^2 - \|\widetilde{U}_M^T X_i\|_F^2\right)}{\sum_{i=1}^{I} \|X_i\|_F^2} \\
\leq & (1 - \theta^N \theta^v) + (1 - \theta^M \theta^u) \\
= & (1 - \theta^v \theta^Q) + (1 - \theta^u \theta^P).
\end{aligned}$$

CHAPTER 3: GROUP PARAMETRIC INDEPENDENT COLORED SOURCES

3.1 Introduction

Independent Component Analysis is a statistical method for extracting factors or components from a random vector $\mathbf{x}$. One way to approach this problem is to express the random vector as a linear mixture of some underlying, latent or hidden source random vector $\mathbf{s}$ so that

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{3.1}$$

where $\mathbf{A}$ is a non-random matrix. To make this ill-defined problem more tractable, it is necessary to impose some conditions on the hidden source $\mathbf{s}$. A common approach is to assume the source consists of *independent components* or independent random variables that are nongaussian. See (Hyvärinen et al., 2001; Stone, 2004) for a survey of this idea and a wide variety of applications.

The main appeal of ICA is its flexibility in representing multivariate data. Typically, the random vector x is viewed as an array of sensors that can be recorded repeatedly resulting a huge amount of data both in space (sensors) and time (repetition). The sensor array can be one-dimensional as in the cock-tail party problem originally formulated in terms of the *blind source separation* (BSS) in signal processing (Hyvärinen et al., 2001); two-dimensional as in electroencephalogram (EEG), magnetoencephalographic (MEG) brain activation recordings (Vigário et al., 2000); or three-dimensional *functional magnetic resonance imaging* (fMRI) for examining brain dynamics (McKeown et al., 1998). The number of sensors ranges from two for the BSS problem to hundreds of thousands in fMRI. The temporal part may be short, a few hundred time points in fMRI; or long, millions in EEG. So the data to be considered are very

high in dimension. Finding ways to visualize them is one of the strengths of ICA.

Many ICA algorithms have been developed to estimate sources. For instance, *Infomax* (Bell and Sejnowski, 1995), *fastICA* (Hyvärinen et al., 2001), *Kernel ICA* (Bach and Jordan, 2002), and ProDenICA (Hastie and Tibshirani, 2002). Most of these methods employed only the marginal density information of the sources. However, for fMRI data and other functional type neuroimaing data, usually there exist temporal autocorrelation structures for each source (Worsley et al., 2002). Incorporating this temporal information into ICA would conceivably reveal more features of the sources.

Recently, Lee et al. (2011) proposed a novel ICA algorithm by considering the spectral properties of the sources and the algorithm is named *colorICA* or *parametric independent colored sources* (PICS). The algorithm assumes that each temporal source has its own parametric autocorrelation specified by the *autoregressive* (AR), *moving average* (MA), or *autoregressive moving average* (ARMA) structures. The approach is carried out in the spectral domain via the Whittle likelihood (Whittle, 1952) which is expressed as a function of the sensor time series along with the source parameters (both the correlation or ARMA coefficients and noise variance), and a matrix reflecting linear mixing operations (mixing matrix). The estimates of time series parameters and mixing matrix are obtained by minimizing the negative Whittle log-likelihood.

The new approach is very suitable for multivariate data analysis whose temporal pattern is the key to the analysis, very much like an important predictor or carrier in regression problems. For example, in many brain activation experiments, the subjects will carry out certain tasks such as finger tappings that have some temporal patterns. The main interest is to correlate these patterns with the spatial brain maps. Specifically, an fMRI dataset consists a series of three-dimensional (3D) images observed over time. Each 3D image represents measurements of *blood oxygen level dependent* (BOLD) contrast at a specific time point. Before statistical analysis, these 3D images are usually transformed into vectors via vectorization operation and

hence the four-dimensional (4D) fMRI dataset can be represented by a two-dimensional (2D) space-time matrix. Each column of the matrix consists a vectorized 3D image and each row is a temporal course at a particular spatial location (voxel). Lee et al. (2011) considered decomposing the fMRI data into a set of spatial maps and the associated temporal courses. The core of the analysis is to extract the main temporal component carrying the experimental task information along with its activation brain map. It was also demonstrated that the temporal correlations did play an important role in extracting the source features.

The description given so far is based on the notion of temporal features. That is, the source temporal components are considered to be independent time series. This form of ICA is referred to as *temporal ICA* (tICA). A parallel development can be formulated in terms of spatial features, where the spatial maps are independent random vectors, viewed as the columns of the mixing matrix. This is known as *spatial ICA* (sICA). Most algorithms are developed based on (3.1) since sICA is the transpose version of tICA. Thus, PICS is a tICA approach because the algorithm assumes independent temporal sources.

In recent years, a great deal of effort has been put into extending ICA to multi-subjects analysis. Contrary to the above single subject analysis in which the data is acquired for one run, one session, or one subject; many human brain mapping studies involve multiple runs, sessions, or subjects. The main question now is how to compare brains of two subjects, or brains of several groups of subjects. Motivated by these situations, Calhoun et al. (2001) proposed the *group ICA* (GICA) approach which applies ICA to multiple subjects or groups. GICA has enjoyed an increasing popularity in the analysis of multiple neuroimaging observations and many GICA procedures have been developed over the last ten years. See Calhoun et al. (2009), Calhoun and Adali (2012) and Hyvärinen (2013) for more related reviews on this topic.

Multiple fMRI observations can be organized in a hierarchical way such that one observation from upper level consists of various samples from one level below. For instance, the dataset would have many groups and each group contains multiple subjects (i.e., a two-level

organization). The most two commonly studied hierarchical organizations are one-level (i.e., multiple subjects) and two-level (i.e., various groups with each group consisting of multiple subjects). The organization has levels more than two can be analyzed in a similar way. Hence, in this chapter, we focus on these two types. GICA on one-level and two-level organizations are named *multisubject GICA* and *multigroup GICA* respectively. Multisubject GICA usually assumes heterogeneity across subjects while multigroup GICA assumes homogeneity among subjects within the same group and heterogeneity across groups. In the following, we only review multisubject GICA. Multigroup GICA can be extended by additional homogeneity subjects constraint.

One of the goals of multi-subject GICA is to identify features shared by subjects. Recall that single subject ICA decomposes fMRI data into spatial maps and temporal courses. Based on prior knowledge on the data, we are interested in estimating common spatial maps, temporal courses, or both. These different types of priors are named *group structures*. Each GICA approach assumes one or more types of group structures as a prior and the estimation procedure is mainly directed by the assumption. Calhoun et al. (2001) imposes common spatial maps and subject specific temporal courses assumptions and conducts analysis via sICA. Beckmann and Smith (2005) assumes both common spatial maps and temporal courses with subject specific loadings and the approach is an extension of *probabilistic ICA* (PICA) (Beckmann and Smith, 2004). The Matlab toolbox *GIFT* and software package *MELODIC* which implement aforementioned two approaches have been widely used in GICA studies. Guo and Pagnoni (2008) and Guo (2011) proposed *Expectation-Maximization* (EM) algorithm based GICA approach which can incorporate various group structures. Esposito et al. (2005) assumes that each individual subject IC can be represented as a function of group common components and proposed to apply single subject ICA first on each individual matrix and then categorizes single subject ICs via *self-organizing clustering*. Guo and Tang (2013) assumes similar group structure such that subject ICs are group common components with subject specific noise and proposed an

EM algorithm to estimate them.

All of the aforementioned GICA approaches do not consider the temporal autocorrelation structures. Some of them can only incorporate one type of group structures and most of them are based on sICA and not scalable to large datasets. In this chapter, we propose a novel tICA type GICA approach which is an extension of PICS to group inferences. We name our approach *group parametric independent colored sources* (GPICS). GPICS represents each temporal source by a parametric time series model and hence automatically takes the autocorrelation structures into account. Moreover, if the prior knowledge is that subjects share the same temporal sources, previous GICA approaches would assume individuals have exactly same set of temporal courses. However, GPICS would assume the sources of each subject have same time series models and allows temporal courses to be different across subjects. This makes GPICS take the subject level noise into consideration. Furthermore, the parameters are estimated through maximizing log-likelihood in the spectral domain by an iterative procedure. The log-likelihood is a function of data samples, time series parameters and mixing matrices. Hence, various group structures can be accommodated into GPICS. In addition, the procedure only needs one step dimension reduction via PCA on the spatial direction for single subjects. Since the temporal direction of single subject fMRI data is usually small, after spatial direction dimension reduction step, GPICS is scalable to a large number of subjects.

3.2 Background on ICA and PICS

We first introduce some basic definitions and notations for time series analysis in Section 3.2.1 and review ICA model in Section 3.2.2. We provide details of PICS procedure in Section 3.2.3.

3.2.1 Preliminaries

This section describes some basic notions and tools for analyzing time series considered in this chapter.

Let $Y(t)$, $t = 0, \pm 1, \pm 2, \ldots$ denote a real-valued stationary time series in the strict sense. That is, the finite dimensional distribution of $Y(t_1 + u), Y(t_2 + u), \ldots, Y(t_k + u)$ is equal to the finite dimensional distribution of $Y(t_1), Y(t_2), \ldots, Y(t_k)$ for $k = 1, 2, \ldots$ and $u \in \mathbb{R}$ (Brillinger, 2001; Brockwell and Davis, 2009). The autocovariance function is defined by $c(u) = \mathrm{cov}(Y(t), Y(t + u))$, $u \in \mathbb{R}$. Suppose the condition $\sum_u |c(u)| < \infty$ holds. Then the spectral density function or simply spectrum of Y is defined by

$$f(r) = \frac{1}{2\pi} \sum_{u=-\infty}^{\infty} c(u) \exp\{-iru\}, \quad -2\pi \leq r \leq 2\pi.$$

Much of statistical inference will be focused on estimating the spectral density function $f(r)$ for $r \in [-2\pi, 2\pi]$. Here r is referred to as the frequency. Note that this will lead to an estimate of the autocovariance function $c(\cdot)$ via the inverse Fourier transformation (Brillinger, 2001).

A popular approach to estimate the spectral density function is based on the discrete Fourier transform (DFT), which can be described as follows. Consider a realization $Y(0), Y(1), \ldots, Y(T-1)$ of length T from the time series Y. The DFT of that realization is defined by

$$\varphi(r_k, Y) = \sum_{t=0}^{T-1} Y(t) \exp(-ir_k t), \quad r_k = 2\pi k/T, \quad k = 0, \ldots, T-1.$$

The second-order periodogram is given by

$$\tilde{f}(r_k, Y) = \frac{1}{2\pi T} |\varphi(r_k, Y)|^2, \quad k = 0, \ldots, T-1.$$

Alternatively, the spectrum f can be estimated by fitting parametric time series models.

This will now be described. Let B denote the backshift operator, i.e., $BY(t) = Y(t-1)$. The AR process of order p (AR(p)), MA process of order q (MA(q)) and ARMA process of orders p and q (ARMA(p,q)) can be written as $\mathbf{\Phi}(B)Y(t) = z(t)$, $Y(t) = \mathbf{\Theta}(B)z(t)$ and $\mathbf{\Phi}(B)Y(t) = \mathbf{\Theta}(B)z(t)$ respectively where $z(t) \sim WN(0, \sigma^2)$ and $t = 0, \pm 1, \pm 2, \ldots$. Here $\mathbf{\Phi}(z) = 1 - \phi_1 z - \cdots - \phi_p z^p$ and $\mathbf{\Theta}(z) = 1 + \theta_1 z + \cdots + \theta_q z^q$ are AR and MA polynomials of degree p and q respectively.

For these processes, the spectral density functions will be a rational function whose coefficients are identified by the parameters of the processes. This type of spectral density functions will be denoted by $f(r, \boldsymbol{\phi})$, where $\boldsymbol{\phi}$ is a vector of parameters specified by the ARMA models above. Consequently, the statistical estimation will concentrate on the estimation of $\boldsymbol{\phi}$ using some types of likelihood approach (Brockwell and Davis, 2009).

The Whittle likelihood (Whittle, 1952) is one of such approaches. The basic idea of this approach is to compare the periodogram and the parametric model based estimates. This is given by

$$L(f; Y) = -\frac{1}{2} \sum_{k=0}^{T-1} \left\{ \frac{\tilde{f}(r_k, Y)}{f(r_k, \boldsymbol{\phi})} + \ln f(r_k, \boldsymbol{\phi}) \right\}. \tag{3.2}$$

Under some appropriate conditions, it has been shown that the Whittle estimates possess some optimal properties (Dzhaparidze and Kotz, 1986).

This procedure will be applicable if the source in ICA or BSS is given by the Y series. However, a modification is necessary if the source is observed through the mixtures. That is, the observed series Y is a mixture of the source(s). This is how the notion of independent components (IC) was developed. Before describing ICA, it is necessary to note that the above description for the univariate time series Y can be extended to vector-valued series. More specifically, let $\mathbf{x}(t) \in \mathbb{R}^M$ for $t = 0, 1, \ldots, T-1$ denote T observations of a real-valued vector stationary process with mean $\mathbf{0}$ and autocovariance function $\mathbf{c_{xx}}(u) = \mathrm{cov}(\mathbf{x}(t), \mathbf{x}(t+u))$ satisfying the condition $\sum_{u=-\infty}^{\infty} |\mathbf{c_{xx}}(u)| < \infty$. Then the second-order periodogram is given

as

$$\tilde{\mathbf{f}}(r_k, \mathbf{X}) = \frac{1}{2\pi T}\boldsymbol{\varphi}(r_k, \mathbf{X})\boldsymbol{\varphi}^*(r_k, \mathbf{X}),$$

where $\boldsymbol{\varphi}(r_k, \mathbf{X}) = \sum_{t=0}^{T-1} \mathbf{x}(t) \exp\{-ir_k t\}$ is the DFT and * denotes conjugate transpose operator.

3.2.2 Independent Component Analysis

ICA assumes observed multivariate signals as a linear mixture of hidden independent sources such that at most one source can have Gaussian distribution. Let $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_M(t)]^T$ be an M dimensional random vector representing the observation signal at time point t for $t = 0, 1, \ldots, T-1$ and let the vector $\mathbf{s}(t) = [s_1(t), s_2(t), \ldots, s_M(t)]^T$ denote the corresponding random multivariate source signal. The *noise-free ICA* model can be mathematically expressed as:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \quad t = 0, 1, \ldots, T-1. \tag{3.3}$$

Here $\mathbf{A}$ is a nonsingular deterministic matrix of size $M \times M$ representing the linear mixing operator and it is called *mixing matrix*. The inverse of A is denoted as $\mathbf{W} = \mathbf{A}^{-1}$ under the name *unmixing matrix*.

Given T observations $\mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(T-1)$, ICA aims at recovering the mixing matrix $\mathbf{A}$ and sources $\mathbf{s}(0), \mathbf{s}(1), \ldots, \mathbf{s}(T-1)$. Let $\mathbf{X} = [\mathbf{x}(0), \ldots, \mathbf{x}(T-1)]$ and $\mathbf{S} = [\mathbf{s}(0), \ldots, \mathbf{s}(T-1)]$ be the row-wise concatenation of $\mathbf{x}(t)$ and $\mathbf{s}(t)$ respectively. Then, for $t = 0, 1, \ldots, T-1$, Equation (3.3) can be combined and written in a concise form as:

$$\mathbf{X}_{M\times T} = \mathbf{A}_{M\times M}\mathbf{S}_{M\times T}. \tag{3.4}$$

The source signals $\mathbf{S}$ can be recovered by $\hat{\mathbf{S}} = \hat{\mathbf{W}}\mathbf{X}$ where $\hat{\mathbf{W}}$ is an estimate of the unmixing matrix $\mathbf{W}$.

As an important application, ICA has been widely used in fMRI data analysis. Let $\mathbf{Y}$ of

size $V \times T$ denote a single observation where V and T are the numbers of voxels and time points respectively. Each row of $\mathbf{Y}$ is a temporal course at a specific voxel and each column represents a vectorized 3D image at a time point. Single subject ICA aims at decomposing $\mathbf{Y}$ as outer products of M spatial maps (represented by the columns of $\mathbf{H}_{V\times M}$ and the associated temporal courses (denoted by the corresponding row of $\mathbf{S}_{M\times T}$). Then ICA decomposition on the observation $\mathbf{Y}$ can be written as

$$\mathbf{Y}_{V\times T} = \mathbf{H}_{V\times M}\mathbf{S}_{M\times T}.$$

For fMRI data, V is usually very large. Hence, there is usually a pre-processing step to reduce the dimension of $\mathbf{Y}$ to $M \times T$ and ICA decomposition is conducted via Equation (3.4). We will discuss the data pre-processing further in Section 3.3.3

3.2.3 Parametric Independent Colored Sources

Suppose the sources $s_j(0), s_j(1), \ldots, s_j(T-1)$ were available. Then each source can be fitted by a parametric model using the Whittle likelihood (3.2) given by

$$L(f_{jj}; s_j) = -\frac{1}{2}\sum_{k=0}^{T-1}\left\{\frac{\tilde{f}(r_k, s_j)}{f_{jj}(r_k, \boldsymbol{\phi}_j)} + \ln f_{jj}(r_k, \boldsymbol{\phi}_j)\right\},$$

where $f_{jj}(\cdot)$ is the spectral density function of the j-th source. Since ICA assumes that sources are mutually independent, the joint Whittle log-likelihood for all sources is the sum of individual ones and can be written as

$$L(\mathbf{f}; \mathbf{s}) = -\frac{1}{2}\sum_{j=1}^{M}\sum_{k=0}^{T-1}\left\{\frac{\tilde{f}(r_k, s_j)}{f_{jj}(r_k, \boldsymbol{\phi}_j)} + \ln f_{jj}(r_k, \boldsymbol{\phi}_j)\right\},$$

where $\mathbf{f} = [f_{11}, f_{22}, \ldots, f_{MM}]^T$ and $\mathbf{s} = [s_1, s_2, \ldots, s_M]^T$.

In practice, our observations are the mixture signals $\mathbf{x}(t)$ for $t = 0, 1, \ldots, T-1$. Recover

the source signals through the unmixing matrix $\mathbf{W}$ via $\mathbf{s}(t) = \mathbf{W}\mathbf{x}(t)$, we can express the second-order periodogram of source s_j in terms of the observation matrix $\mathbf{X}$ and the unmixing matrix $\mathbf{W}$ by

$$\tilde{f}(r_k, s_j) = \mathbf{e}_j^T \mathbf{W} \tilde{\mathbf{f}}(r_k, \mathbf{X}) \mathbf{W}^T \mathbf{e}_j,$$

where $\mathbf{e}_j \in \mathbb{R}^M$ with the jth element being 1 and others being 0. Then the source log-likelihood can be rewritten as

$$L(\mathbf{W}, \mathbf{f}; \mathbf{X}) = -\frac{1}{2} \sum_{j=1}^{M} \sum_{k=0}^{T-1} \left\{ \frac{\mathbf{e}_j^T \mathbf{W} \tilde{\mathbf{f}}(r_k, \mathbf{X}) \mathbf{W}^T \mathbf{e}_j}{f_{jj}(r_k, \boldsymbol{\phi}_j)} + \ln f_{jj}(r_k, \boldsymbol{\phi}_j) \right\} + T \ln|\det(\mathbf{W})|. \quad (3.5)$$

In fMRI study, it has been shown that modeling the temporal sources via AR type time series is efficient (Worsley et al., 2002; Lee et al., 2011). Hence, in this chapter, we would only consider the AR type sources. MA and ARMA models can be considered similarly. The spectral density of AR(p_j) model is given by

$$f_{jj}(r, \boldsymbol{\phi}_j) = \frac{\sigma_j^2}{2\pi |\boldsymbol{\Phi}(e^{-ir})|^2}.$$

Moreover, let $\boldsymbol{\phi}$ and $\boldsymbol{\sigma}^2$ symbolically denote all source AR coefficients and noise levels, i.e., $\boldsymbol{\phi} = [\phi_{11}, \ldots, \phi_{1p_1}, \ldots, \phi_{Mp_M}]^T$ and $\boldsymbol{\sigma}^2 = [\sigma_1^2, \ldots, \sigma_M^2]^T$. Then we can rewrite the Whittle log-likelihood $L(\mathbf{W}, \mathbf{f}; \mathbf{X})$ as $L(\mathbf{W}, \boldsymbol{\phi}, \boldsymbol{\sigma}^2; \mathbf{X})$.

Without loss of generality, we may assume that the unmixing matrix is orthogonal. In fact, this can be achieved by prewhitening the data before applying ICA. See Section 3.3.3, Lee et al. (2011) and the references therein. To derive the estimates of time series model parameters and unmixing matrix with the orthogonality constraint, Lee et al. (2011) proposed a Lagrange multiplier method, named cICA-YW, by minimizing a constraint negative log-likelihood,

$$F^{PICS}(\mathbf{W}, \boldsymbol{\phi}, \boldsymbol{\sigma}^2, \boldsymbol{\lambda}; \mathbf{X}) = -L(\mathbf{W}, \boldsymbol{\phi}, \boldsymbol{\sigma}^2; \mathbf{X}) + \boldsymbol{\lambda}^T \mathbf{C}, \quad (3.6)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{M(M+1)/2}$ is the Lagrange multiplier, and $\mathbf{C} \in \mathbb{R}^{M(M+1)/2}$ with $\mathbf{C}_{(i-1)M+j} = (\mathbf{W}\mathbf{W}^T - \mathbf{I}_{M\times M})_{ij}$ for $i = 1, \ldots, M$ and $j = 1, \ldots, i$.

The cICA-YW algorithm alternatingly updates $[\mathrm{vec}^T(\mathbf{W}), \boldsymbol{\lambda}^T]^T$ and time series parameters $\boldsymbol{\phi}$ and $\boldsymbol{\sigma}^2$ by fixing the others where $\mathrm{vec}(\mathbf{W})$ transforms matrix $\mathbf{W}$ into a vector of length M^2 by stacking columns of W on top of each other. It updates the unmixing matrix and Lagrange multiplier via Newton-Raphson method as

$$[\mathrm{vec}^T(\check{\mathbf{W}}), \check{\boldsymbol{\lambda}}^T]^T = [\mathrm{vec}^T(\mathring{\mathbf{W}}), \mathring{\boldsymbol{\lambda}}^T]^T - (\ddot{F}(\mathring{\mathbf{W}}, \mathring{\boldsymbol{\phi}}, \mathring{\boldsymbol{\sigma}}^2, \mathring{\boldsymbol{\lambda}}))^{-1}\dot{F}(\mathring{\mathbf{W}}, \mathring{\boldsymbol{\phi}}, \mathring{\boldsymbol{\sigma}}^2, \mathring{\boldsymbol{\lambda}}),$$

where $\dot{F}$ and $\ddot{F}$ are first and second derivatives of Equation (3.6) with respect to $[\mathrm{vec}^T(\mathbf{W}), \boldsymbol{\lambda}^T]$, and $\check{}$ and $\mathring{}$ denote variables in current and previous iterations respectively. After obtaining unmixing matrix updates, the sources can be estimated by $\check{\mathbf{S}} = \check{\mathbf{W}}\mathbf{X}$. The jth row of $\check{\mathbf{S}}$ are T observations of the jth source. Then the AR coefficients and noise levels of this source are estimated via Yule-Walker method and the order p_j can be selected by traditional approaches such as Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). Finally, the algorithm is terminated if the Amari error (Amari et al., 1996) between $\check{\mathbf{W}}$ and $\mathring{\mathbf{W}}$ is below a pre-specified threshold. The Amari error is defined as

$$d(\check{\mathbf{W}}, \mathring{\mathbf{W}}) = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{\sum_{j=1}^{M}|\mathbf{G}_{ij}|}{\max_j|\mathbf{G}_{ij}|} - 1\right) + \frac{1}{M}\sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{M}|\mathbf{G}_{ij}|}{\max_i|\mathbf{G}_{ij}|} - 1\right),$$

where $\mathbf{G} = \check{\mathbf{W}}\mathring{\mathbf{W}}^{-1}$.

3.3 Methodology

In this section, we provide details of our GPICS approach which is an extension of PICS to group inference. We start with some notations and group structures considered by GPICS. Then we present the objective functions involving unmixing matrix and time series parameters in Section 3.3.1 and optimization procedures in Section 3.3.2. The inputs of GPICS are

dimension reduced and prewhitened from raw data. Hence, we describe approaches for dimension reduction of a group of images in Section 3.3.3 which is needed for both pre- and post-processing the data.

We illustrate our approach using the multigroup hierarchical organization type since multi-subject GPICS is a special case of multigroup GPICS by taking only one subject in each group. Suppose we observe G groups of data samples and each group has n_g subjects. Let $\mathbf{X}^{gi}$ of size $M \times T$ denote the ith observation matrix from group g for $g = 1, 2, \ldots, G$ and $i = 1, 2, \ldots, n_g$ with single subject ICA decomposition $\mathbf{X}^{gi} = \mathbf{A}^{gi}\mathbf{S}^{gi}$ where $\mathbf{A}^{gi}$ and $\mathbf{S}^{gi}$ are mixing matrices and temporal courses respectively. Further let $\boldsymbol{\phi}_j^{gi} = [\phi_{j1}^{gi}, \phi_{j2}^{gi}, \ldots, \phi_{jp_j}^{gi}]^T$ and σ_j^{2gi} denote the AR coefficients and noise level of the jth source of subject i in group g.

We construct group structures for GPICS as follows. Within each group, we assume that the group is homogeneous both in space and time, i.e., $\mathbf{A}^{gi} = \mathbf{A}^g$, $\boldsymbol{\phi}_j^{gi} = \boldsymbol{\phi}_j^g$, and $\sigma_j^{2gi} = \sigma_j^{2g}$. Between groups, we can impose four different kinds of structures: (1) H_{00}: Homogeneous in both space and time, i.e., $\mathbf{A}^g = \mathbf{A}$, $\boldsymbol{\phi}_j^g = \boldsymbol{\phi}_j$ and $\sigma_j^{2g} = \sigma_j^2$; (2) H_{01}: Homogeneous in space but not in time, i.e., $\mathbf{A}^g = \mathbf{A}$; (3) H_{10}: Homogeneous in time but not in space, i.e., $\boldsymbol{\phi}_j^g = \boldsymbol{\phi}_j$ and $\sigma_j^{2g} = \sigma_j^2$; and (4) H_{11}: Inhomogeneous in both space and time. We summarize the four group structures in Table 3.1.

Group structure	Mixing matrix	Unmixing matrix	AR coefficients	AR noise level
H_{00}	$\mathbf{A}^{gi} = \mathbf{A}^g = \mathbf{A}$	$\mathbf{W}^{gi} = \mathbf{W}^g = \mathbf{W}$	$\boldsymbol{\phi}_j^{gi} = \boldsymbol{\phi}_j^g = \boldsymbol{\phi}_j$	$\sigma_j^{2gi} = \sigma_j^{2g} = \sigma_j^2$
H_{01}	$\mathbf{A}^{gi} = \mathbf{A}^g = \mathbf{A}$	$\mathbf{W}^{gi} = \mathbf{W}^g = \mathbf{W}$	$\boldsymbol{\phi}_j^{gi} = \boldsymbol{\phi}_j^g$	$\sigma_j^{2gi} = \sigma_j^{2g}$
H_{10}	$\mathbf{A}^{gi} = \mathbf{A}^g$	$\mathbf{W}^{gi} = \mathbf{W}^g$	$\boldsymbol{\phi}_j^{gi} = \boldsymbol{\phi}_j^g = \boldsymbol{\phi}_j$	$\sigma_j^{2gi} = \sigma_j^{2g} = \sigma_j^2$
H_{11}	$\mathbf{A}^{gi} = \mathbf{A}^g$	$\mathbf{W}^{gi} = \mathbf{W}^g$	$\boldsymbol{\phi}_j^{gi} = \boldsymbol{\phi}_j^g$	$\sigma_j^{2gi} = \sigma_j^{2g}$

Table 3.1: Summary of group structures.

3.3.1 Objective functions of GPICS

In this section, we present objective functions used to estimate unmixing matrices and time series parameters of GPICS. Recall that the objective function of PICS has two parts: the one that contains negative Whittle log-likelihood and the other one that involves penalties the

unmixing matrices. In analogy with the single subject case, the objective function of GPICS also consists of two parts and it varies across group structures. In the following, we use the equation

$$F^{GS}(\mathbf{W}, \boldsymbol{\phi}, \boldsymbol{\sigma}^2, \boldsymbol{\lambda}; \mathbf{X}) = -L^{GS}(\mathbf{W}, \boldsymbol{\phi}, \boldsymbol{\sigma}^2; \mathbf{X}) + J^{GS}(\mathbf{W}), \tag{3.7}$$

to symbolically represent the objective functions for GPICS. Here, GS stands for one particular group structure (i.e., GS = H_{00}, H_{01} or H_{10}), $\mathbf{W}$, $\boldsymbol{\phi}$, $\boldsymbol{\sigma}^2$, $\boldsymbol{\lambda}$ and $\mathbf{X}$ represent unmixing matrices, time series parameters and data samples for all subjects respectively and the dimension of these quantities may vary with group structures, and $L^{GS}(\cdot)$ and $J^{GS}(\cdot)$ denote Whittle log-likelihood and penalty terms for group structure GS respectively.

Since the observations are independent, the joint log-likelihood of all samples is the sum of all individual ones, i.e.,

$$L^{GS}(\mathbf{W}, \boldsymbol{\phi}, \boldsymbol{\sigma}^2; \mathbf{X}) = \sum_{g=1}^{G} \sum_{i=1}^{n_g} L(\mathbf{W}^{gi}, \boldsymbol{\phi}^{gi}, \boldsymbol{\sigma}^{2gi}; \mathbf{X}^{gi}), \tag{3.8}$$

where $L(\mathbf{W}^{gi}, \boldsymbol{\phi}^{gi}, \boldsymbol{\sigma}^{2gi}; \mathbf{X}^{gi})$ is given in Equation 3.5. Plugging Equation (3.5) and group structures in Table 3.1 into Equation (3.8), we can write the Whittle log-likelihood of GPICS in terms of subject level unmixing matrices, time series parameters and data samples. The expressions are summarized in Table 3.2.

For the cases H_{00} and H_{01}, we have only one universal unmixing matrix $\mathbf{W}$ across all subjects. Hence, the penalty term is just $\boldsymbol{\lambda}^T\mathbf{C}$ where $\boldsymbol{\lambda}$ is the Lagrange multiplier and $\mathbf{C}$ is explained in Equation (3.6). For the case H_{10}, the unmixing matrices are different across groups. We need to penalize every unmixing matrix in the objective function as $\sum_{g=1}^{G} \boldsymbol{\lambda}^{gT}\mathbf{C}^g$ where $\boldsymbol{\lambda}^{gT}\mathbf{C}^g$ is the penalty term for the gth unmixing matrix $\mathbf{W}^g$. The penalty terms for three group structures are summarized in Table 3.2.

GS	$L^{GS}(\mathbf{W}, \boldsymbol{\phi}, \boldsymbol{\sigma}^2; \mathbf{X})$	$J^{GS}(\mathbf{W})$
H_{00}	$-\frac{1}{2}\sum_{g,i,j,k}\left\{\frac{\mathbf{e}_j^T\mathbf{W}\tilde{\mathbf{f}}(r_k,\mathbf{X}^{gi})\mathbf{W}^T\mathbf{e}_j}{f_{jj}(r_k)}+\ln f_{jj}(r_k)\right\}+\sum_{g,i}T\ln\lvert\det(\mathbf{W})\rvert$	$\boldsymbol{\lambda}^T\mathbf{C}$
H_{01}	$-\frac{1}{2}\sum_{g,i,j,k}\left\{\frac{\mathbf{e}_j^T\mathbf{W}\tilde{\mathbf{f}}(r_k,\mathbf{X}^{gi})\mathbf{W}^T\mathbf{e}_j}{f^g_{jj}(r_k)}+\ln f^g_{jj}(r_k)\right\}+\sum_{g,i}T\ln\lvert\det(\mathbf{W})\rvert$	$\boldsymbol{\lambda}^T\mathbf{C}$
H_{10}	$-\frac{1}{2}\sum_{g,i,j,k}\left\{\frac{\mathbf{e}_j^T\mathbf{W}^g\tilde{\mathbf{f}}(r_k,\mathbf{X}^{gi})\mathbf{W}^{gT}e_j}{f_{jj}(r_k)}+\ln f_{jj}(r_k)\right\}+\sum_{g,i}T\ln\lvert\det(\mathbf{W}^g)\rvert$	$\sum_g\boldsymbol{\lambda}^{gT}\mathbf{C}^g$

Table 3.2: Summary of Whittle log-likelihood and penalty terms for GPICS. $\sum_{g,i,j,k}$, $\sum_{g,i}$ and $\sum_g$ are abbreviations of $\sum_{g=1}^{G}\sum_{i=1}^{n_g}\sum_{j=1}^{M}\sum_{k=0}^{T-1}$, $\sum_{g=1}^{G}\sum_{i=1}^{n_g}$ and $\sum_{g=1}^{G}$ respectively.

3.3.2 Optimization procedures

Recall that PICS alternatingly updates the vector containing unmixing matrix and Lagrange multiplier and the vector of times series parameters by fixing the others. We propose to optimize the GPICS objective function, Equation (3.7), in a similar way. We iteratively updates the set of unmixing matrices and associated Lagrange multipliers via Newton-Raphson method and the set of time series parameters via Yule-Walker method. For different group structures, the updating rules vary slightly. We start with how to update the unmixing matrix and Lagrange multiplier of various group structures and then present the estimation procedure of obtaining time series parameters.

Updating unmixing matrices and Lagrange multipliers. For the H_{00} and H_{01} cases, the objective functions contain only one unmixing matrix $\mathbf{W}$ and Lagrange multiplier $\boldsymbol{\lambda}$. And thus these parameters can be updated via Newton-Raphson method in the same way as PICS by

$$[\mathrm{vec}^T(\breve{\mathbf{W}}),\breve{\boldsymbol{\lambda}}^T]^T=[\mathrm{vec}^T(\mathring{\mathbf{W}}),\mathring{\boldsymbol{\lambda}}^T]^T-(\ddot{F}^{GS}(\mathring{\mathbf{W}},\mathring{\boldsymbol{\phi}},\mathring{\boldsymbol{\sigma}}^2,\mathring{\boldsymbol{\lambda}}))^{-1}\dot{F}^{GS}(\mathring{\mathbf{W}},\mathring{\boldsymbol{\phi}},\mathring{\boldsymbol{\sigma}}^2,\mathring{\boldsymbol{\lambda}}),\qquad(3.9)$$

where GS stands for H_{00} and H_{01}.

For the H_{10} case, the objective function contains G unmixing matrices $\{\mathbf{W}^g\}$ for $g=1,2,\ldots,G$. We use $F^{10,g}(\mathbf{W}^g,\boldsymbol{\phi},\boldsymbol{\sigma}^2,\boldsymbol{\lambda}^g;\mathbf{X})$ to denote the objective function for the gth group and $L^{10,gi}(\mathbf{W}^g,\boldsymbol{\phi},\boldsymbol{\sigma}^2,\boldsymbol{\lambda}^g;bX^{gi})$ to denote the Whittle log-likelihood of subject i in group g.

Then we have

$$F^{10,g}(\mathbf{W}^g, \boldsymbol{\phi}, \boldsymbol{\sigma}^2, \boldsymbol{\lambda}^g; \mathbf{X}) = -\sum_{i=1}^{n_g} L^{10,gi}(\mathbf{W}^g, \boldsymbol{\phi}, \boldsymbol{\sigma}^2, \boldsymbol{\lambda}^g; \mathbf{X}^{gi}) + \boldsymbol{\lambda}^{gT}\mathbf{C}^g,$$

and

$$F^{10}(\mathbf{W}, \boldsymbol{\phi}, \boldsymbol{\sigma}^2, \boldsymbol{\lambda}; \mathbf{X}) = \sum_{g=1}^{G} F^{10,g}(\mathbf{W}^g, \boldsymbol{\phi}, \boldsymbol{\sigma}^2, \boldsymbol{\lambda}^g; \mathbf{X}).$$

Since $\mathbf{W}^g$ and $\boldsymbol{\lambda}^g$ only appear in $F^{10,g}(\mathbf{W}^g, \boldsymbol{\phi}, \boldsymbol{\sigma}^2, \boldsymbol{\lambda}^g; \mathbf{X})$, we can update them one by one via

$$[\mathrm{vec}^T(\check{\mathbf{W}}^g), \check{\boldsymbol{\lambda}}^{gT}]^T = [\mathrm{vec}^T(\mathring{\mathbf{W}}^g), \mathring{\boldsymbol{\lambda}}^{gT}]^T - (\ddot{F}^{10,g}(\mathring{\mathbf{W}}^g, \mathring{\boldsymbol{\phi}}, \mathring{\boldsymbol{\sigma}}^2, \mathring{\boldsymbol{\lambda}}^g))^{-1}\dot{F}^{10,g}(\mathring{\mathbf{W}}^g, \mathring{\boldsymbol{\phi}}, \mathring{\boldsymbol{\sigma}}^2, \mathring{\boldsymbol{\lambda}}^g). \tag{3.10}$$

Updating time series parameters. After updating the unmixing matrices, source temporal courses can be recovered by $\mathring{\mathbf{S}}^{gi} = \mathring{\mathbf{W}}\mathbf{X}^{gi}$ for the H_{00} and H_{01} cases and $\mathring{\mathbf{S}}^{gi} = \mathring{\mathbf{W}}^g\mathbf{X}^{gi}$ for the H_{10} case. Then time series model parameters can be estimated from these source temporal courses.

In the H_{00} case, for the jth source, there exists only one set of parameters across subjects. Let $\mathring{\mathbf{s}}_j^{giT}$ denote the jth row of $\mathring{\mathbf{S}}^{gi}$ and $\check{\boldsymbol{\phi}}_j$ and $\check{\sigma}_j^2$ are updating AR coefficients and variance of this source. We concatenate $\{\mathring{\mathbf{s}}_j^{giT}\}$ horizontally and denote it as $\mathring{\mathbf{r}}_j = [\mathring{\mathbf{s}}_j^{11T}, \mathring{\mathbf{s}}_j^{12T}, \ldots, \mathring{\mathbf{s}}_j^{1n_1T}, \ldots, \mathring{\mathbf{s}}_j^{Gn_GT}]$. $\mathring{\mathbf{r}}_j$ can be considered as a temporal course of length TN where $N = \sum_{g=1}^{G} n_g$ is the total number of subjects. Then $\check{\boldsymbol{\phi}}_j$ and $\check{\boldsymbol{\sigma}}_j^2$ can be estimated from $\mathring{\mathbf{r}}_j$ by Yule-Walker method.

In the H_{01} case, time series parameters are different across groups. For the jth source parameters of group g, only data in group g contribute to the estimation. Within this group, it is a H_{00} structure and thus $\check{\boldsymbol{\phi}}_j^g$ and $\check{\sigma}_j^{2g}$ can be estimated from $\mathring{\mathbf{r}}_j^g = [\mathring{\mathbf{s}}_j^{g1T}, \mathring{\mathbf{s}}_j^{g2T}, \ldots, \mathring{\mathbf{s}}_j^{gn_gT}]$ via Yule-Walker method.

In the H_{10} case, the time series parameters are the same across subjects. Hence, we can adopt a similar idea as in H_{00} case that collects all temporal courses associated with the jth

source from all subjects and concatenate them to estimate $\check{\boldsymbol{\phi}}_j$ and $\check{\sigma}_j^2$. Different from the H_{00} case, we need to match temporal components across subjects and groups because ICA does not have an order among ICs. From our experiences, as long as the initial $\mathbf{W}^g$ is close to the truth, the order of ICs in iteration steps would not change. Hence if we could give good initials of $\mathbf{W}^g$, we do not need to match ICs within the iteration steps but match them among initials. Applying H_{00} algorithm on each group, we obtain initial estimates for $\mathbf{W}^g$ which is denoted as $\mathbf{W}^g_{\text{init}}$. We estimate the group sources as $\mathbf{S}^g_{\text{init}} = \mathbf{W}^g_{\text{init}}[\mathbf{X}^{g1}, \mathbf{X}^{g2}, \ldots, \mathbf{X}^{gn_g}]$ for $g = 1, 2, \ldots, G$. Then we match the row orders of $\mathbf{S}^g_{\text{init}}$ for $g = 2, 3, \ldots, G$ with the row order of $\mathbf{S}^1_{\text{init}}$ by correlation coefficients. For instance, we calculate the correlation coefficients of the first row of $\mathbf{S}^1_{\text{init}}$ with each row of $\mathbf{S}^g_{\text{init}}$ and find the row has the maximum absolute coefficient. Then we move this row to the first row of $\mathbf{S}^g_{\text{init}}$. Continue this process for the rest rows of $\mathbf{S}^1_{\text{init}}$ until all rows of $\mathbf{S}^g_{\text{init}}$ are matched to $\mathbf{S}^1_{\text{init}}$. We further adjust column orders of $\mathbf{W}^g_{\text{init}}$ corresponding to the row orders of $\mathbf{S}^g_{\text{init}}$. Once the orders of initial $\mathbf{W}^g_{\text{init}}$ are matched, in each iteration step, we can estimate $\check{\boldsymbol{\phi}}_j$ and $\check{\boldsymbol{\sigma}}_j^2$ from $\mathring{\mathbf{r}}_j = [\mathring{\mathbf{s}}_j^{11T}, \mathring{\mathbf{s}}_j^{12T}, \ldots, \mathring{\mathbf{s}}_j^{1n_1T}, \ldots, \mathring{\mathbf{s}}_j^{Gn_GT}]$ via Yule-Walker approach. We note that only in the H_{10} case we need to consider the order matching problem. This is because the orders of temporal sources is determined by the order of unmixing matrix. The same mixing matrix assumption in the H_{00} and H_{01} cases naturally makes the order of unmixing matrix fixed across subjects, and thus the order of temporal components.

Initialization We need first give unmixing matrices initials as part of the inputs. For the H_{00} and H_{01} cases, $\mathbf{W}_{\text{init}}$ can be given as identity matrix or estimated by any single subject ICA algorithm applied on the concatenation matrix $[\mathbf{X}^{11}, \mathbf{X}^{12}, \ldots, \mathbf{X}^{1n_1}, \ldots, \mathbf{X}^{Gn_G}]$. For the H_{10} structure, as described above, the algorithm first gets estimates of $\mathbf{W}^g_{\text{init}}$ and then reorder the columns of $\mathbf{W}^g_{\text{init}}$ by matching temporal components via correlation coefficients.

Stopping criterion The algorithm is terminated if $d(\check{\mathbf{W}}, \mathring{\mathbf{W}})$ for H_{00} and H_{01} cases or $\max_{g\in\{1,2,\ldots,G\}} d(\check{\mathbf{W}}^g, \mathring{\mathbf{W}}^g)$ for H_{10} case is below a pre-specified threshold level.

Summary of GPICS procedure. In general, GPICS procedure first initializes unmixing matrices by appropriate choices based on group structure assumptions. And then alternatingly updates unmixing matrices and time series parameters by fixing the others. The algorithm stops when the Amari distances of unmixing matrices between two successive iterations are below a pre-defined threshold. The algorithm is summarized in Algorithm 3.1.

Algorithm 3.1 GPICS algorithm

Initialize unmixing matrices $\mathbf{W}$ or $\mathbf{W}^g$ based on group structure assumptions.
Alternatingly do the following two steps until the Amari distance of $\mathbf{W}$ or maximum distances of $\{\mathbf{W}^g\}$ between two iteration steps are below the threshold.

1. Recover the source signals by $\mathring{\mathbf{S}}^{gi} = \mathring{\mathbf{W}}^g\mathbf{X}^{gi}$ or $\mathring{\mathbf{S}}^{gi} = \mathring{\mathbf{W}}\mathbf{X}^{gi}$. Estimate time series parameters for the jth source by Yule-Walker method on the jth row of $[\mathring{\mathbf{S}}^{g1}, \mathring{\mathbf{S}}^{g2}, \ldots, \mathring{\mathbf{S}}^{gn_g}]$ for the gth group of H_{01} case or $[\mathring{\mathbf{S}}^{11}, \mathring{\mathbf{S}}^{12}, \ldots, \mathring{\mathbf{S}}^{1n_1}, \ldots, \mathring{\mathbf{S}}^{Gn_G}]$ of the H_{00} and H_{10} cases.
2. Update unmixing matrices and Lagrange multipliers via Equation (3.9) for H_{00} and H_{01} cases and Equation (3.10) for the H_{10} case.

3.3.3 Dimension reduction of a group of images

In single subject fMRI ICA analysis, the input $\mathbf{X}_{M\times T}$ is dimension reduced and prewhitened (pre-processing) data of the real fMRI observation $\mathbf{Y}_{V\times T}$, i.e., $V \geq M$ and $\frac{1}{T}\mathbf{X}\mathbf{X}^T = \mathbf{I}_{M\times M}$. For GPICS, the inputs $\{\mathbf{X}^{gi}_{M\times T}\}$ also need dimension reduction and prewhitening so that the orthogonal unmixing matrices assumptions are satisfied. In this section, we present details on dimension reduction and prewhitening of raw observations $\{\mathbf{Y}^{gi}_{V\times T}\}$ according to the GPICS group structure assumptions. We start with a brief review of single subject fMRI data preprocessing and then extend it to group dimension reduction.

Suppose $\mathbf{Y}_{V\times T}$ has *singular value decomposition* (SVD) as $\mathbf{Y} = \mathbf{P}\mathbf{D}\mathbf{Q}^T$ where $\mathbf{P}$ and $\mathbf{Q}$ are orthogonal matrices of size $V \times r$ and $T \times r$ respectively, $\mathbf{D}$ is diagonal matrix of size $r \times r$

with the ith diagonal entry being singular value d_i and r is the rank of $\mathbf{Y}$. Let $\tilde{\mathbf{P}}_{V\times M}$ and $\tilde{\mathbf{Q}}_{T\times M}$ contain the first M columns of $\mathbf{P}$ and $\mathbf{Q}$ respectively and let diagonal matrix $\tilde{\mathbf{D}}_{M\times M}$ consist of the first M singular values as diagonal entries. The dimension reduction step is done by projecting columns of $\mathbf{Y}$ onto the subspace spanned by the columns of $\tilde{\mathbf{P}}$ by $\tilde{\mathbf{Y}} = \tilde{\mathbf{P}}\mathbf{Y} = \tilde{\mathbf{D}}\tilde{\mathbf{Q}}^T$. The prewhitening step is applied on the reduced data $\tilde{\mathbf{Y}}$ by $\sqrt{T}\tilde{\mathbf{D}}^{-1}\tilde{\mathbf{Y}} = \sqrt{T}\tilde{\mathbf{Q}}^T$. Hence, the pre-processed data $\mathbf{X}$ is actually $\mathbf{X} = \sqrt{T}\tilde{\mathbf{Q}}^T$. On the other hand, the goals of single subject ICA applied to fMRI data are to estimate spatial maps and associated temporal courses by decomposing $\mathbf{Y}$ as $\mathbf{Y}_{V\times T} \approx \mathbf{H}_{V\times M}\mathbf{S}_{M\times T}$. Therefore, after applying ICA on X by $\mathbf{X} = \mathbf{AS}$, we can recover spatial maps $\mathbf{H}$ by $\hat{\mathbf{H}} = \tilde{\mathbf{P}}\tilde{\mathbf{D}}\hat{\mathbf{A}}/\sqrt{T}$.

Single subject ICA pre-processes the raw data by projecting columns onto a subspace and the result is the scaled first M right singular vectors of the raw data. Motivated by this fact, GPICS does the pre-processing by approximately project columns of each data matrix onto a common subspace with orthogonal right subject specific components. Formally, GPICS do the groupwise dimension reduction by $\mathbf{Y}^{gi} \approx \tilde{\mathbf{P}}\tilde{\mathbf{D}}\tilde{\mathbf{Q}}^{giT}$ for H_{00} and H_{01} cases and $\mathbf{Y}^{gi} \approx \tilde{\mathbf{P}}^g\tilde{\mathbf{D}}^g\tilde{\mathbf{Q}}^{giT}$ for the H_{10} case where $\tilde{\mathbf{P}}_{V\times M}$ or $\tilde{\mathbf{P}}^g_{V\times M}$ are orthogonal matrices representing groupwise common left components, $\tilde{\mathbf{D}}_{M\times M}$ or $\tilde{\mathbf{D}}^g_{M\times M}$ consists of groupwise singular values, and $\tilde{\mathbf{Q}}^{gi}_{T\times M}$ are orthogonal matrices denoting subject specific right components. We shall discuss how to obtain the approximation later. The dimension reduced and prewhitened data $\mathbf{X}^{gi}$ are given as $\mathbf{X}^{gi} = \sqrt{T}\tilde{\mathbf{Q}}^{giT}$. After applying GPICS onto $\{\mathbf{X}^{gi}\}$, we can obtain group spatial maps by $\hat{\mathbf{H}} = \tilde{\mathbf{P}}\tilde{\mathbf{D}}\hat{\mathbf{A}}/\sqrt{T}$ for H_{00} and H_{01} cases or $\hat{\mathbf{H}}^g = \tilde{\mathbf{P}}^g\tilde{\mathbf{D}}^g\hat{\mathbf{A}}^g/\sqrt{T}$ for the H_{10} case.

We shall illustrate how to obtain group and subject-specific components only on the H_{10} structure because these components in the H_{00} and H_{01} cases can be derived in the same way by taking all raw data $\{\mathbf{Y}^{gi}\}$ as one single group. GPICS approximates $\mathbf{Y}^{gi}$ by $\mathbf{Y}^{gi} \approx \tilde{\mathbf{P}}^g\tilde{\mathbf{D}}^g\tilde{\mathbf{Q}}^{giT}$. The left components $\tilde{\mathbf{P}}^g$ and singular values $\tilde{\mathbf{D}}^g$ can be obtained by the *2DSVD* method (Ding and Ye, 2005) to small and moderate dimensions and *APVD* approach to massive dataset.

2DSVD first concatenates data matrices along temporal direction as $[\mathbf{Y}^{g1}, \mathbf{Y}^{g2}, \ldots, \mathbf{Y}^{gn_g}]$, followed by SVD decomposition $\mathbf{P}^g\mathbf{D}^g\mathbf{Q}^{gT}$ on this aggregation matrix, and then extracts the first M components and singular values from $\mathbf{P}^g$ and $\mathbf{D}^g$ to form $\tilde{\mathbf{P}}^g$ and $\tilde{\mathbf{D}}^g$. APVD is a two step SVD procedure. In the first step, APVD decomposes each data matrix through SVD by $\mathbf{Y}^{gi} = \mathbf{P}^{gi}\mathbf{D}^{gi}\mathbf{Q}^{giT}$ and extracts the first M left components and singular values from each subject. In the second step, APVD concatenates the scaled left singular vectors horizontally by $[\tilde{\mathbf{P}}^{g1}\tilde{\mathbf{D}}^{g1}, \tilde{\mathbf{P}}^{g2}\tilde{\mathbf{D}}^{g2}, \ldots, \tilde{\mathbf{P}}^{gn_g}\tilde{\mathbf{D}}^{gn_g}]$, followed by another SVD step on the aggregation matrix and extracts the first M left components and singular values as the final estimates of $\tilde{\mathbf{P}}^g$ and $\tilde{\mathbf{D}}^g$. By splitting one SVD on large matrix as in 2DSVD to several SVDs on relatively small matrices, APVD is efficiently scalable to large dataset.

After obtaining the group components $\tilde{\mathbf{P}}^g$ and $\tilde{\mathbf{D}}^g$,we propose to derive subject specific right components $\tilde{\mathbf{Q}}^{gi}$ as follows. Let $\tilde{\mathbf{p}}_j^g$ and $\tilde{\mathbf{q}}_j^{gi}$, $j = 1, 2, \ldots, M$, denote the jth column of $\tilde{\mathbf{P}}^g$ and $\tilde{\mathbf{Q}}^{gi}$ respectively. Let $\tilde{\mathbf{Q}}_j^{gi} = [\tilde{\mathbf{q}}_1^{gi}, \tilde{\mathbf{q}}_2^{gi}, \ldots, \tilde{\mathbf{q}}_j^{gi}]$ be the matrix containing the first j columns of $\tilde{\mathbf{Q}}^{gi}$ and let $\tilde{\mathbf{Q}}_0^{gi} = \mathbf{0}_{T\times T}$. Then, for $j = 1, 2, \ldots, M$, we can obtain columns $\tilde{\mathbf{q}}_j^{gi}$ of $\tilde{\mathbf{Q}}^{gi}$ sequencially by the recursive equation

$$\tilde{\mathbf{q}}_j^{gi} = \frac{\left(\mathbf{I}_{T\times T} - \tilde{\mathbf{Q}}_{j-1}^{gi}\tilde{\mathbf{Q}}_{j-1}^{giT}\right)\mathbf{Y}^{giT}\tilde{\mathbf{p}}_j^g}{\|\left(\mathbf{I}_{T\times T} - \tilde{\mathbf{Q}}_{j-1}^{gi}\tilde{\mathbf{Q}}_{j-1}^{giT}\right)\mathbf{Y}^{giT}\tilde{\mathbf{p}}_j^g\|}.$$

In summary, our GPICS procedure to fMRI data first reduces the dimension and prewhitens the raw data. And then appropriate GPICS algorithm shown in Algorithm 3.1 is applied to these pre-processed data based on group structure assumptions.

3.4 Simulations

In this section, we examine the performance of GPICS through simulations and compare our approach with the fastICA based group ICA approaches. The results will demonstrate that GPICS works better than fastICA based approaches in both blind source separation and brain

active region detections.

3.4.1 Blind source separation

In the first simulation, we study GPICS performance of recovering blind sources. We test the procedure under three different group structures. For each case, we generate 2 groups of data with 5 subjects in each group. Data samples are generated from the model $\mathbf{X}^{gi} = \mathbf{A}\mathbf{S}^{gi}$ for H_{00} and H_{01} cases and $\mathbf{X}^{gi} = \mathbf{A}^g\mathbf{S}^{gi}$ for the H_{10} case, $g = 1, 2$ and $j = 1, 2, \ldots, 5$, where $\mathbf{A}$ and $\mathbf{A}^g$ are of size 4×4 and $\mathbf{S}^{gi}$ has dimension 4×512. Hence, the number of ICs is $M = 4$ and the number of time points is $T = 512$.

In each simulation run, we randomly generate one orthogonal mixing matrix for the H_{00} and H_{01} cases and two for the H_{10} case. Let $\boldsymbol{\phi}_j^g$, $j = 1, \ldots, 4$, denote the coefficients for the jth source of group g. Each row of $\mathbf{S}^{gi}$ is generated from an AR model with coefficients given in Table 3.3. In the H_{00} case and H_{10} cases, $\boldsymbol{\phi}_j^1 = \boldsymbol{\phi}_j^2$, we take the set 1 coefficients for all subjects. In the H_{01} case, we take the set 1 and set 2 coefficients for the subjects in the first and second group respectively. The noise of all sources are generated from uniform$[-\sqrt{3}, \sqrt{3}]$.

	$\boldsymbol{\phi}_1$	$\boldsymbol{\phi}_2$	$\boldsymbol{\phi}_3$	$\boldsymbol{\phi}_4$
set 1	$[0.8]$	$[-0.6, -0.5]$	$[0.1, -0.8]$	$[-0.85, -0.7, 0.2]$
set 2	$[0.5, -0.12]$	$[0.6]$	$[-0.7, -0.3]$	$[0.3]$

Table 3.3: AR coefficients.

We estimate $\mathbf{A}$ or $\mathbf{A}^g$ and time series parameters by both GPICS and fastICA procedures. GPICS approach was described in Section 4.2. For the fastICA based procedures, in the H_{00} and H_{01} cases, we concatenate data matrices along temporal direction as $[\mathbf{X}^{11}, \mathbf{X}^{12}, \ldots, \mathbf{X}^{25}]$ and do an ICA decomposition via fastICA on this aggregation matrix to derive common mixing matrix. In the H_{10} case, we do fastICA separately on two aggregation matrices $[\mathbf{X}^{g1}, \mathbf{X}^{g2}, \ldots, \mathbf{X}^{g5}]$ for $g = \{1, 2\}$ to obtain group specific mixing matrices. The Amari distances between the estimates and the truth for both approaches are calculated. The temporal courses are recovered by $\hat{\mathbf{S}}^{gi} = \hat{\mathbf{W}}\mathbf{X}^{gi}$ or $\hat{\mathbf{S}}^{gi} = \hat{\mathbf{W}}^g\mathbf{X}^{gi}$. Then we calculate all pairs of correlation coefficients

between the estimated temporal courses and the truth. We did 100 runs for each group structure. The results are shown in Figure 3.1 for Amari distances and Figure 3.2 for temporal correlations.

Figure 3.1 shows the boxplots of Amari distance between the estimation mixing matrices and the truth for 100 simulation runs. Each subplot corresponds to one group structure. We can see that our GPICS method performs consistently better than fastICA based approaches in all group structures. Moreover, there is more variability found in the fastICA based procedures.

Figure 3.2 shows the boxplots of temporal correlations of estimated temporal courses and the truth. Each column corresponds to one group structure and each row represents one IC. In most cases GPICS has correlations nearly 1 and they are higher than fastICA in all cases. The latter case also exhibits a much larger variance in the range of correlations.

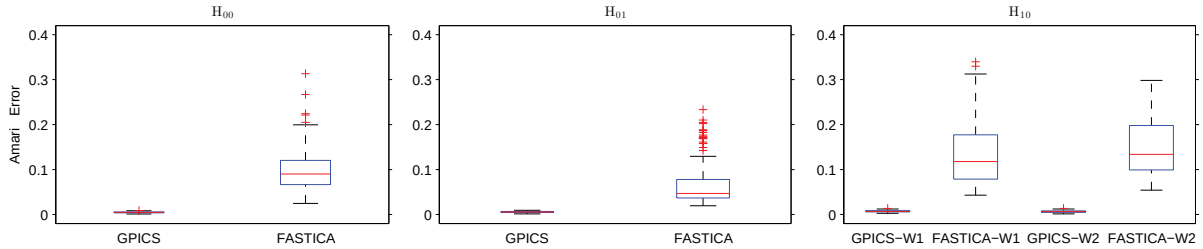


Figure 3.1: Boxplots of 100 simulation runs of Amari distances of estimated mixing matrices and the truth. The GPICS based estimates are consistently near the value zero while the estimates of fastICA based approaches vary greatly with means above zero.

3.4.2 Brain active region detection

In this simulation, we apply our GPICS method to a toy fMRI example to test the performance of identifying brain active regions under three group structures.

We first generate two sets of 3D images of size $20 \times 20 \times 10$ with each set consisting of 4 images. Figure 3.3 shows slices 2,4,6,8 and 10 for each image. Each slice in the figure is of size 20×20. For each image set, we then vectorize images in this set into column vectors of length $4,000$ and concatenate them horizontally. We represent the concatenation matrix by $\mathbf{H}^g_{4,000\times 4}$ where $g \in \{1, 2\}$.

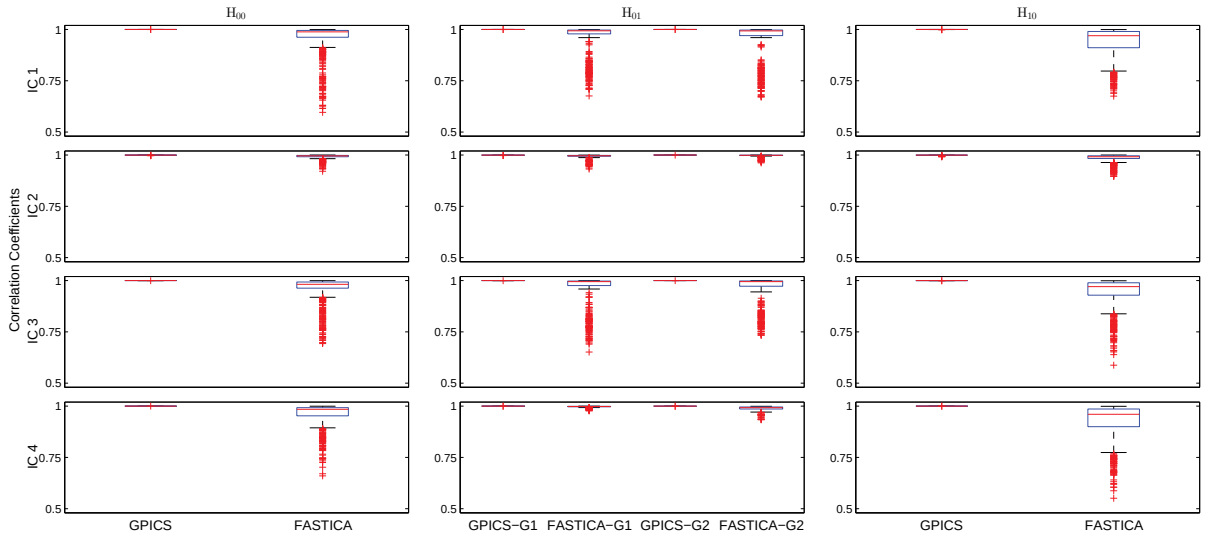


Figure 3.2: Boxplots of 100 simulation runs for temporal correlations of the estimation sources and the truth. The GPICS based estimates are consistently near the value 1 for perfect correlation while the fastICA estimates vary greatly with means below 1.

For each simulation, we generate 2 groups of data and each group has 5 subjects. The data are generated by the model $\mathbf{Y}^{gi} = \mathbf{H}^1\mathbf{S}^{gi}$ for the H_{00} and H_{01} cases or $\mathbf{Y}^{gi} = \mathbf{H}^g\mathbf{S}^{gi}$ for the H_{10} case where $g = 1, 2$, $i = 1, 2, \ldots, 5$ and $\mathbf{S}^{gi}$ is of size 4×128.

Temporal courses are simulated in a signal with additive noise form. The signal of the first IC is a boxcar type function. It starts with an interval of time points being 0, followed by the same interval of time points being 1, and continue this process for the rest. The time interval of boxcar type function for set 1 and 2 are 30 and 40 respectively. In the H_{00} and H_{10}, the signals are from set 1. In the H_{01}, we take signals from group $g = 1, 2$. The signal of the second and the third ICs are being sine functions with different frequencies and phases. The signals are summarized in Table 3.4.

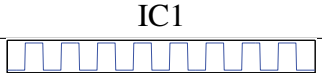
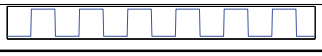
	IC1	IC2	IC3
set 1		$\sin(2\pi 1.17t + 1.61)$	$\sin(2\pi 0.3t + 1.45)$
set 2		$\sin(2\pi 2t + 1.3)$	$\sin(2\pi 2.1t + 1.1)$

Table 3.4: Two sets of signals for the first three ICs.

The AR coefficients for the noise part are given in Table 3.3. Similar to simulations in

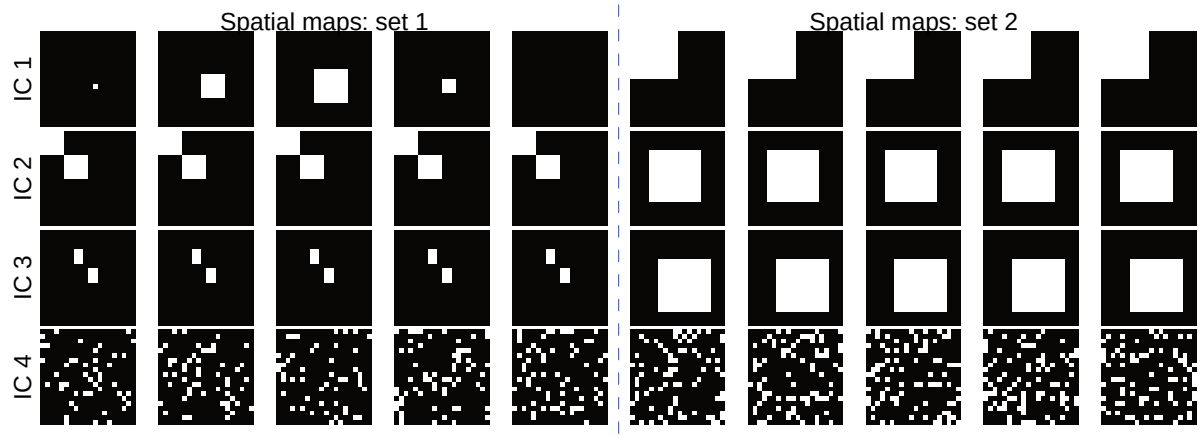


Figure 3.3: Simulated spatial maps.

Section 3.4.1, we use the first set of AR coefficients for the H_{00} and H_{10} cases and both sets for H_{01} case. The $\boldsymbol{\sigma}_j^g$ of each AR time series is derived from signal to noise ratio (SNR) by SNR $=$ (Signal variance)$/$(Noise variance) based on the corresponding AR model.

For each simulation run, We first do a group dimension reduction by prewhitening $\{\mathbf{Y}^{gi}\}$ to yield the pre-processed data matrix $\{\mathbf{X}^{gi}\}$. Then we carry out similar GPICS and fastICA based analysis on $\{\mathbf{X}^{gi}\}$ as shown in Section 3.4.1. Finally, we reconstruct the brain images $\hat{\mathbf{H}}^g$ by the estimated mixing matrices and the group common left components and singular values. To examine significant active regions, we standardize each column of $\mathbf{H}^g$ to obtain the z scores and those regions with $|z| > 2$ are determined as active regions. We did 100 runs for each simulation.

Figure 3.4 shows temporal correlations of 100 simulation runs with $T = 128$ and SNR $=$ 0.5. Figure 3.5 shows selective sliced of estimated brain active regions in the H_{00} case. We can see that GPICS has consistent higher correlations than the fastICA based approaches and recover cleaner active regions.

In order to examine the performance of identifying active regions with respect to the SNR and time points T, we further repeat the above simulations by varying SNR and T. We calculate the false positive (FP) and false negative (FN) rates and report the average over 100 simulations runs. In Figure 3.6, we fix the number of time points to be $T = 256$ and vary SNR as $0.5, 1, 2$

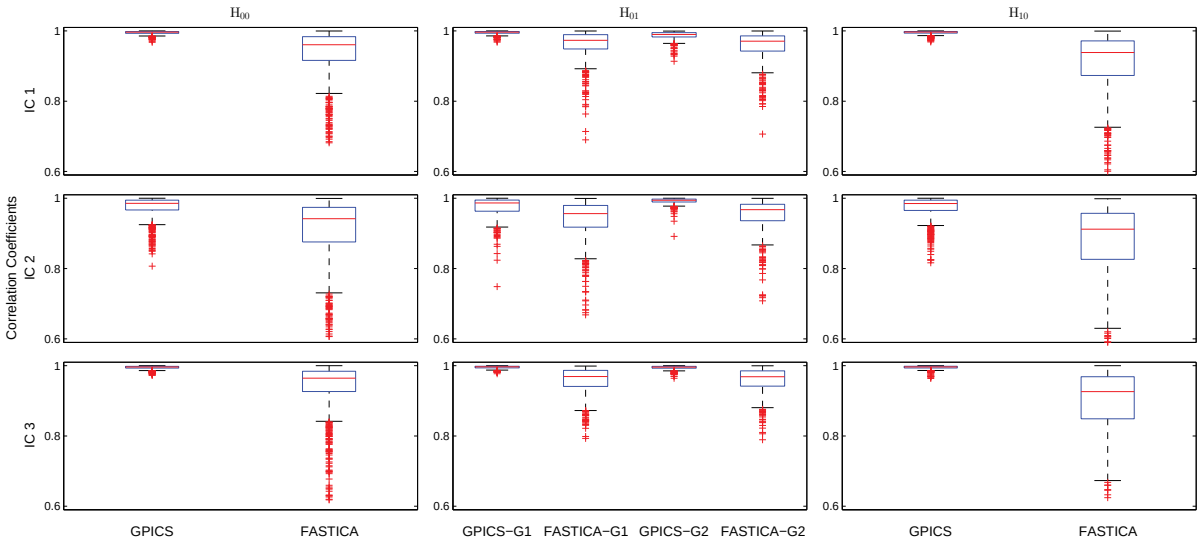


Figure 3.4: Temporal correlations of 100 simulation runs of the estimation temporal courses and the truth. The variability in the fastICA based approaches is clearly greater than the one with GPICS.

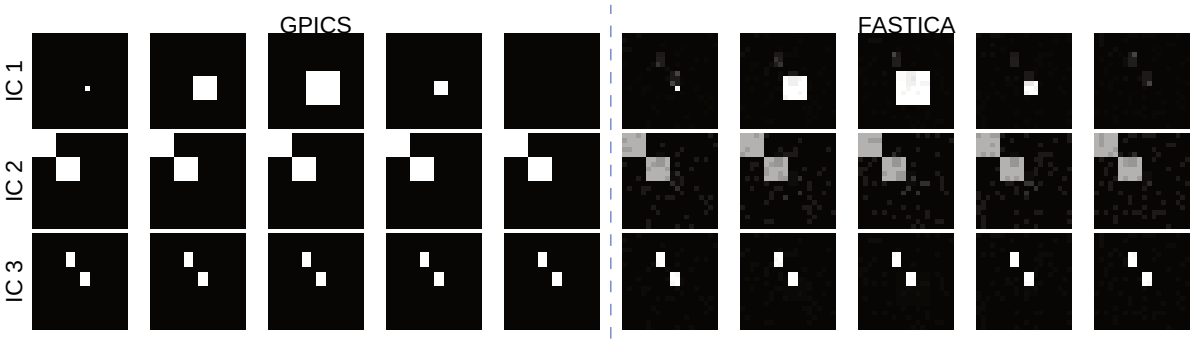


Figure 3.5: Estimated brain active regions for the case H_{00}

and 4. In Figure 3.7, we fix SNR $= 0.5$ and change T as $128, 256, 512$ and 1024. Due to space reason, we only report FNR and FPR for the first group in the H_{10} case. We see that GPICS works consistently better than the fastICA based approaches.

3.5 Real data analysis

In this section, we apply our GPICS approach to one of the open-access *resting-state fMRI* (rs-fMRI) datasets, the ADHD-200 Sample (Milham et al., 2012). The dataset contains a large collection of rs-fMRI and structural MRI scans of 491 typically developing controls and 285

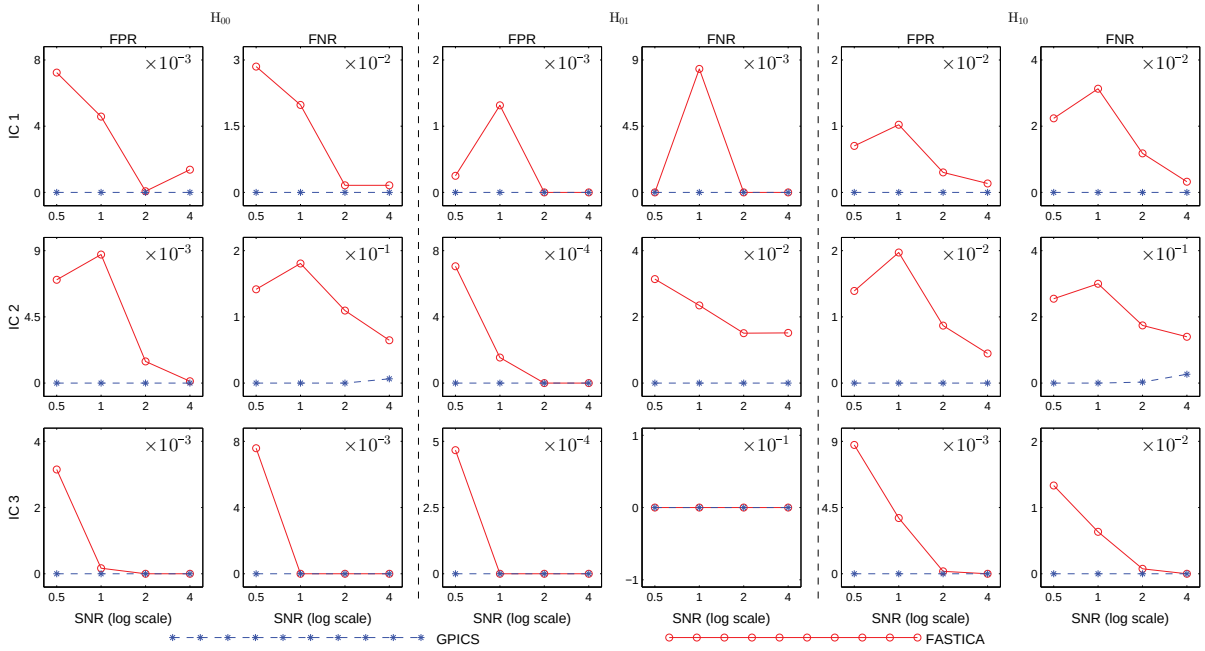


Figure 3.6: False positive and false negative rates as a function of SNR.

attention deficit hyperactivity disorder (ADHD) patients from 8 participating sites (for more details, see `http://fcon_1000.projects.nitrc.org/indi/adhd200/`).

For this study, we utilize the publicly available pre-processed data by the Athena pipeline (`http://neurobureau.projects.nitrc.org/ADHD200/Introduction.html`). Briefly, the pipeline to fMRI data includes slice timing correction, deobliquing, motion correction, registration into MNI152 standard brain space at 4mm $\times$ 4mm $\times$ 4mm resolution, nuisance variance removing, and a 6-mm full width at half maximum (FWHM) Gaussian filter spatial smoothing. Detailed information regarding the preprocessing steps can be found at `http://www.nitrc.org/plugins/mwiki/index.php/neurobureau:AthenaPipeline`. Each processed fMRI image is of size $49 \times 58 \times 47$. We further apply a mask onto each image to extract voxels inside the brain and results in a vector of length $30,316$. The number of time points of each subject is 172. Hence, each fMRI data matrix as the inputs of GPICS is of $30,316 \times 172$.

We choose 30 subjects from each of the three groups: typically developing controls group,

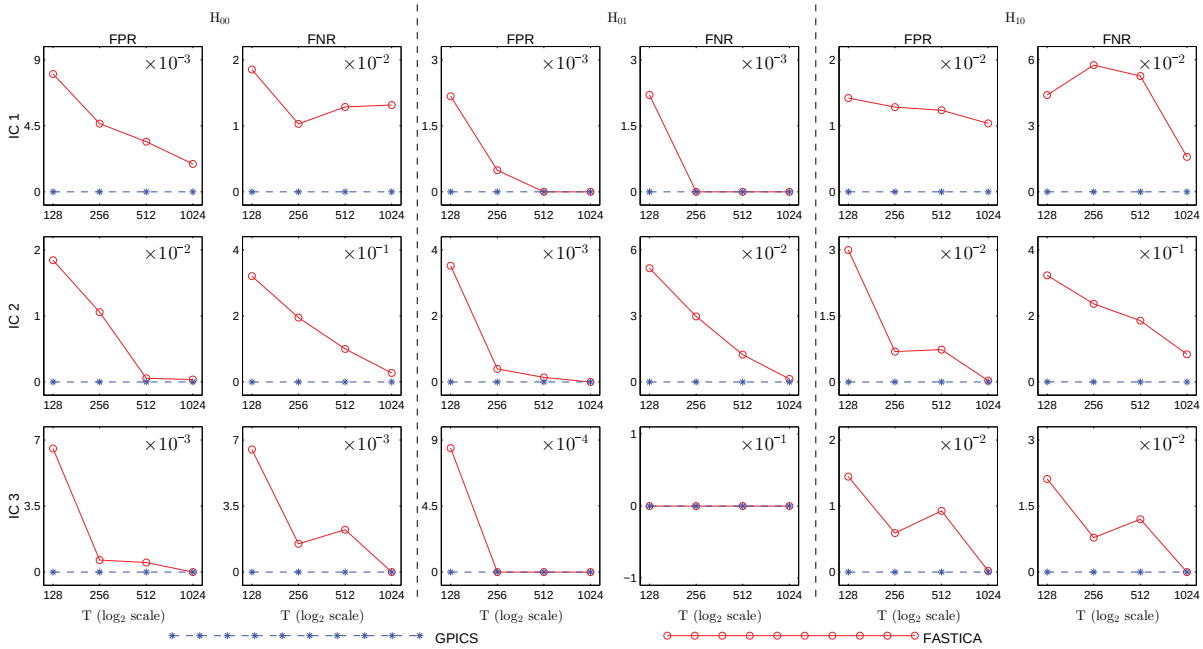


Figure 3.7: False rates as a function of the sample size.

ADHD combined subtype group and ADHD inattentive subtype group. We conduct GPICS separately on the three groups. For each group, we assume that subjects share the same spatial maps but different time series models (i.e., H_{01} case). This group structure assumption is consistent with the one commonly used in multisubject GICA approaches such as GIFT. Since the number of voxels is relatively large, we use APVD to do the dimension reduction instead of 2DSVD. H_{01} type GPICS algorithm is applied on each group and 20 ICs are obtained. In order to identify active voxels, we further calculate the z scores by standardizing each IC and those voxels with $|z| > 2$ are marked as active. Finally, we present identified voxels by overlaying their z scores onto an anatomical template.

For each group, the identified default mode network, visual network and auditory network are provided in Figure 3.8. Each column shows results of one group and each row is a type of brain network. For the default mode and auditory networks, signals are stronger in two ADHD subtypes than control group. While for the visual network, control group has larger absolute z scores than ADHD subtypes. It would be interesting to test the difference among control

groups and ADHD patients. We leave a detailed investigation to a later study.

Figures 3.9–3.11 show the three networks identified by the GPICS and the fastICA based approaches for the three subgroups. We can see that GPICS identifies the frontal part of the DMN in all subgroups. For the visual and auditory networks, GPICS and fastICA based approaches have comparable results.

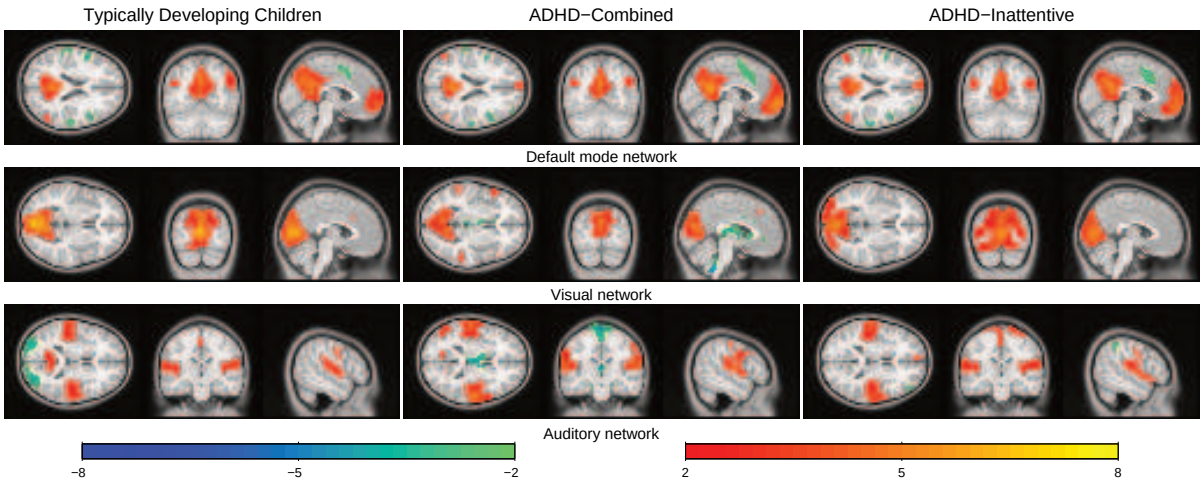


Figure 3.8: ADHD-200 GPICS results.

3.6 Conclusions

In this chapter, we present a new group ICA approach, GPICS. This approach takes the source temporal dependence into consideration by modeling each source via a parametric time series model. The parameters of the time series models and the unmixing matrices are estimated via the Whittle likelihood in the spectral domain. By applying the APVD approach developed in Chapter 2 as a pre-processing step, our method can deal with high dimensional neuroimaging data. The numerical performance of the GPICS approach was demonstrated by both simulations and a real data analysis.

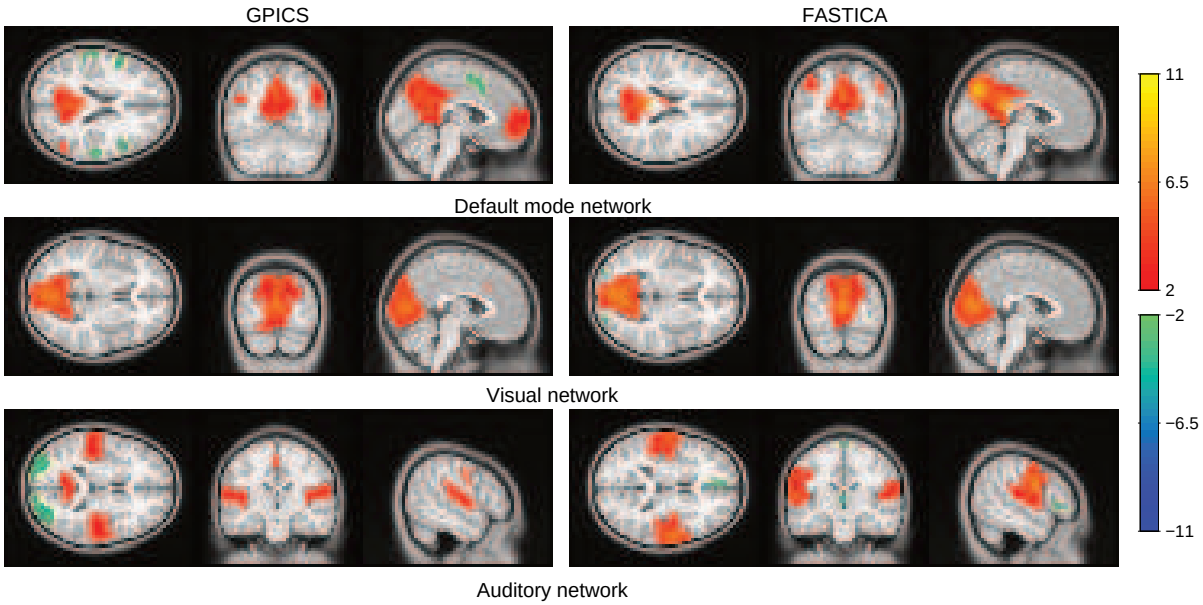


Figure 3.9: Three networks identified by GPICS and fastICA based approaches for typically developing children group.

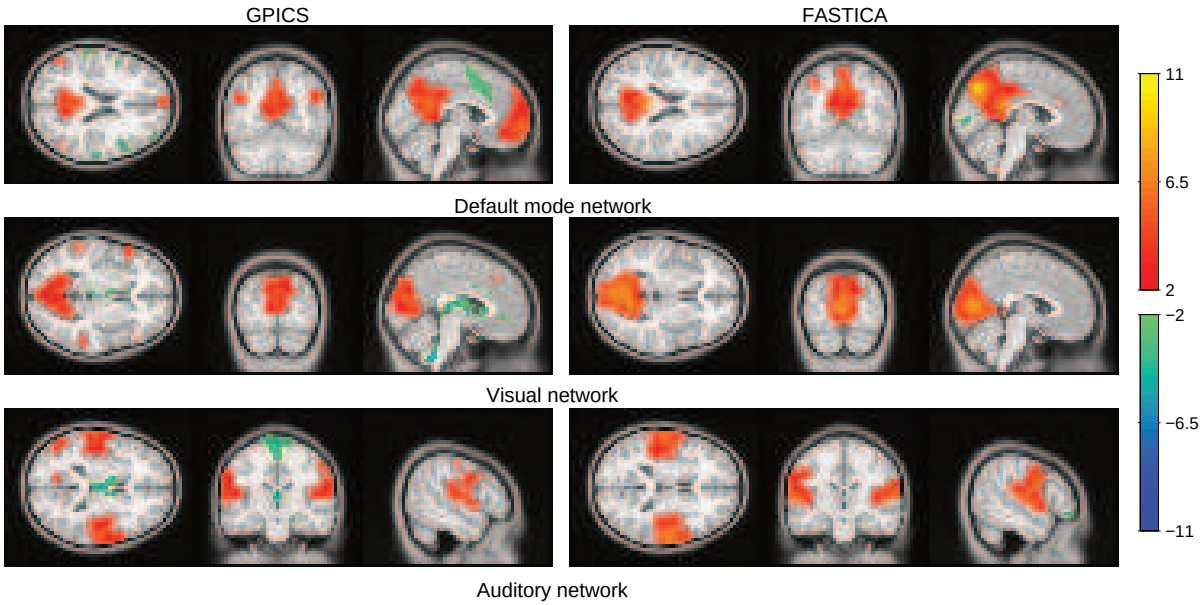


Figure 3.10: Three networks identified by GPICS and fastICA based approaches for ADHD-combined group.

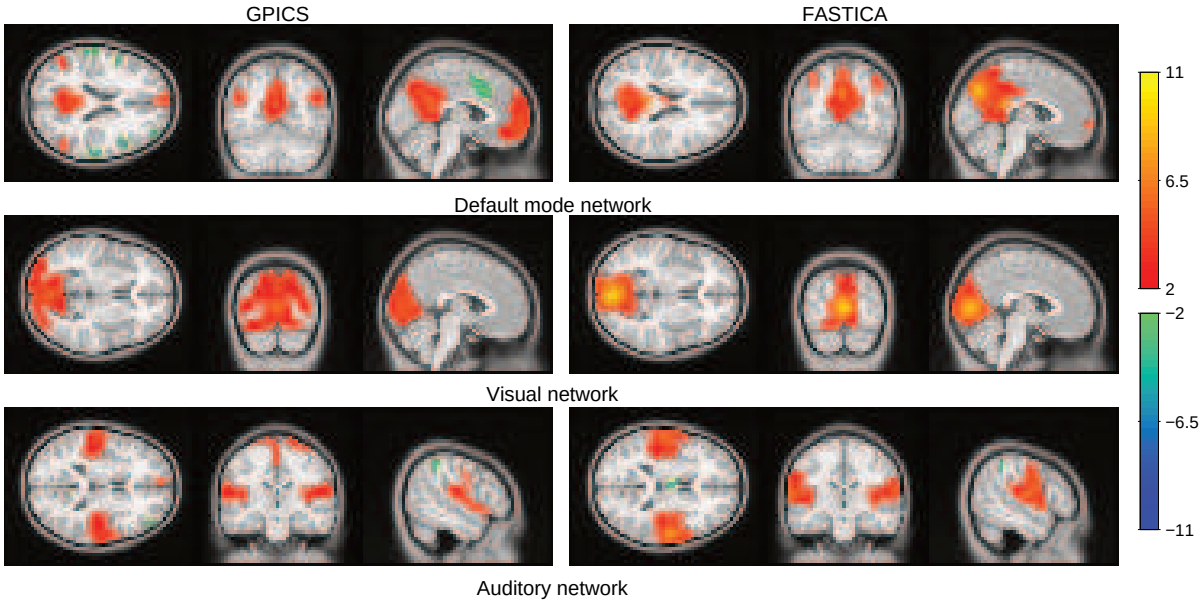


Figure 3.11: Three networks identified by GPICS and fastICA based approaches for ADHD-inattentive group.

CHAPTER 4: SCALAR-ON-MATRIX BILINEAR REGRESSION ANALYSIS

4.1 Introduction

Multiple linear regression is one of the most important statistical tools to establish the relationship between a scalar response variable and a vector of explanatory variables. In many modern applications, instead of vector-valued covariates, there has been an explosion of data sampled in the form of matrices. For instance, researchers often collect matrix-valued medical images and seek to study the association between such images and clinical traits of interest (Zhou et al., 2013). Intriguing problems as such are arising more and more frequently in other fields as well, including finance, economics, agriculture, chemistry, and biology.

One critical feature of the matrix-valued covariates is that there exist important structural information that should be incorporated into the regression analysis, such as column and/or row correlations, low rank properties, and so on. One simple idea is to vectorize the matrix covariates and then apply the classical linear regression. As tempting as it seems, such vectorization can result in a vector of (sometimes) daunting length, in comparison to sample size; more importantly, it destroys structural information intrinsic to the matrix form.

Hence, one interesting question is how to preserve the two-dimensional form of the matrix covariates. This problem has gained attentions from various researchers over the past few years, for example, in discriminant component analysis (Dyrholm et al., 2007), sufficient dimension reduction (Li et al., 2010), logistic regression analysis (Hung and Wang, 2013), scalar-on-matrix regression (Zhao and Leng, 2014), among others.

In this chapter, we consider the scalar-on-matrix regression setting, with a scalar response and matrix covariates. This name originates from imaging analysis, where the matrix covariates are images. Other names in the literature include matrix regression or matrix-variate regression.

In particular, we adopt the bilinear regression model of Zhao and Leng (2014), which assumes that the conditional expectation of the response relates to the matrix covariate $\mathbf{X}$ as $\boldsymbol{\alpha}^T\mathbf{X}\boldsymbol{\beta}$, where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the two coefficient vectors. The standard linear regression equates the conditional expectation of the response and a linear combination of the vector predictors. Given the fact that a matrix is organized in two directions (rows/columns), the above bilinear form is a natural way to extend the linear form to incorporate matrix covariates.

We firstly consider scenarios where the dimensions of the matrix covariates are smaller than the sample size. Our contributions are both methodological and theoretical. We propose two maximum likelihood based estimators for the model parameters. The first estimator is obtained through an iterative algorithm, while the second estimator is a truncated version of the first one, both of which are shown to work well numerically. We then consider the classical asymptotic framework, allowing the sample size to grow, and derive the asymptotic efficiency of both estimators, as well as the inefficiency of the linear regression estimator from the vectorization approach. These fundamental asymptotic results can provide further insights into estimators of higher order tensor data (Zhou et al., 2013). Furthermore, the asymptotic optimality of the truncated estimator may shed lights on other iterative estimators such as the one in the tensor regression model (Zhou et al., 2013). Because our results reassert the phenomenon that early stopping can merit the efficiency of an iterative estimator in the context of covariance matrix estimation (Werner et al., 2008) or precision matrix estimation (Zhou, 2014) of matrix-valued random variables.

We secondly propose a bilinear ridge estimator to deal with the case that the dimensions of the matrix covariates are comparable to or even larger than the sample size. This bilinear ridge estimator is obtained through optimizing the sum of the goodness-of-fit term and an appropriate penalty term. We note that the form of the penalty term, which is of its own interest, is important to study the theoretical properties of the estimator. We carefully choose the one that is a combination of the ridge type penalties on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ with appropriate scaling.

With this form, we establish an upper bound for the excess prediction error. The bound contains two terms: one is as a result of the ridge type regularization, while the other one is due to the randomness in the data.

Although having the same bilinear model as Zhao and Leng (2014), our setting is different from theirs. They are interested in cases where there are a limited number of observations in comparison with the dimensions of the matrix covariates, which are also of great interest. They make additional useful assumptions that the coefficient vectors are both sparse, and consider penalized regression techniques for model estimation.

The bilinear regression model in this chapter can be viewed as a special case of the generalized tensor regression model of Zhou et al. (2013), which considers tensor covariates and employs a multilinear form in the context of exponential family distributions. In addition, our model is a special case of the regularized matrix regression model of Zhou and Li (2014) when their coefficient matrix is assumed to be rank one.

The rest of this chapter is organized as follows. We first describe the scalar-on-matrix bilinear regression model in Section 4.2.1, and introduce the iterative estimator in Section 4.2.2 and its truncated version in Section 4.2.3. We then study the theoretical properties of both estimators in Section 4.3. We propose the bilinear ridge estimator and study its theoretical properties in 4.4. The numerical results of the plain bilinear estimators are presented in Section 4.5.

4.2 Scalar-on-matrix bilinear regression analysis

4.2.1 Bilinear regression model

Classical linear regression model assumes

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta}_0 + \epsilon_i, \quad i = 1, \ldots, n, \tag{4.1}$$

where $\mathbf{x}_i \in \mathbb{R}^p$ is the vector of covariates, $y_i \in \mathbb{R}$ is the scalar response, ϵ_i is the additive noise, and $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ is the unknown coefficient vector. Given n independent and identically distributed observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the goal is to estimate the true parameter vector $\boldsymbol{\beta}_0$.

When dealing with matrix covariates, we adopt the following bilinear regression model of Zhao and Leng (2014):

$$y_i = \boldsymbol{\alpha}_0^T \mathbf{X}_i \boldsymbol{\beta}_0 + \epsilon_i = \boldsymbol{\beta}_0^T \mathbf{X}_i^T \boldsymbol{\alpha}_0 + \epsilon_i, \quad i = 1, \ldots, n, \tag{4.2}$$

where the response $y_i \in \mathbb{R}$ and the random noise $\epsilon_i \in \mathbb{R}$ remain the same as in the traditional linear regression, with $\mathbf{X}_i \in \mathbb{R}^{p \times q}$ being the covariate matrix, and $\boldsymbol{\alpha}_0 \in \mathbb{R}^p$, $\boldsymbol{\beta}_0 \in \mathbb{R}^q$ being unknown parameter vectors of interest. Furthermore, we make the assumption that the noise is normally distributed: $\epsilon_i \sim \mathcal{N}(0, \tau^2)$. We propose two estimators for the unknown parameters $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ in Sections 4.2.2 and 4.2.3, and investigate their theoretical properties in Section 4.3.

The above bilinear model (4.2) has a natural connection with the conventional linear regression model (4.1). Define $\boldsymbol{\theta}_0 = \boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0$, where $\otimes$ denotes the Kronecker product. Then the bilinear model (4.2) can be re-expressed in the form of the general linear regression model (4.1) as

$$y_i = \mathrm{vec}^T(\mathbf{X}_i)\boldsymbol{\theta}_0 + \epsilon_i, \tag{4.3}$$

where $\mathrm{vec}(\mathbf{X}_i)$ denotes the vectorization operator by concatenating the columns of $\mathbf{X}_i$ one by one. The equivalence between (4.2) and (4.3) is due to the fact that $\boldsymbol{\alpha}_0^T \mathbf{X}_i \boldsymbol{\beta}_0 = \mathrm{vec}(\boldsymbol{\alpha}_0^T \mathbf{X}_i \boldsymbol{\beta}_0) = (\boldsymbol{\beta}_0^T \otimes \boldsymbol{\alpha}_0^T)\mathrm{vec}(\mathbf{X}_i) = (\boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0)^T \mathrm{vec}(\mathbf{X}_i) = \mathrm{vec}^T(\mathbf{X}_i)\boldsymbol{\theta}_0$, where the second and third equalities hold because of the properties of the Kronecker product and the vectorization operator. Furthermore, denote $\mathbb{X} = [\mathrm{vec}(\mathbf{X}_1), \mathrm{vec}(\mathbf{X}_2), \ldots, \mathrm{vec}(\mathbf{X}_n)]^T$, $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$ and $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \ldots, \epsilon_n]^T$. Then Model (4.3) can be written in the matrix form as

$$\mathbf{y} = \mathbb{X}\boldsymbol{\theta}_0 + \boldsymbol{\epsilon}. \tag{4.4}$$

In terms of parameter estimation, note that $\boldsymbol{\alpha}_0^T \mathbf{X}_i \boldsymbol{\beta}_0 = (\gamma \boldsymbol{\alpha}_0^T)\mathbf{X}_i(1/\gamma \boldsymbol{\beta}_0)$ for any non-zero constant γ, which suggests that the parameter vectors $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ are identifiable only up to a scale factor. Nevertheless, their Kronecker product $\boldsymbol{\theta}_0 = \boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0$ is identifiable. Therefore, for the rest of the chapter, we compare the performance of various estimators, $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ respectively, through their Kronecker product $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\beta}} \otimes \hat{\boldsymbol{\alpha}}$, by quantifying the discrepancy $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0$.

We want to point out one advantage of the bilinear regression model (4.2) in terms of number of regression coefficients involved. Although Model (4.2) and Model (4.3) are equivalent, the bilinear regression model (4.2) has $p+q$ regression coefficients, while the linear model (4.2) consists of pq coefficients if no assumption is made upon $\boldsymbol{\theta}_0$. When either p or q or both are large, this leads to parsimonious model description and estimation efficiency for the bilinear model. Later we demonstrate such efficiency gain theoretically in Section 4.3.3, and numerically in Section 4.5.3.

To study theoretical properties of the estimators derived in Section 4.3, we consider the random design setting where each covariate matrix $\mathbf{X}_i$ is assumed to follow a matrix normal distribution with zero mean; see Gupta and Nagar (2000); Kollo and von Rosen (2006) for more detailed information. The matrix normal distribution is a natural extension of the multivariate normal distribution from a random vector to a random matrix. Mathematically, a random matrix has a matrix normal distribution, denoted by

$$\mathbf{X}_i \sim \mathcal{N}(\mathbf{0}_{p\times q}, \boldsymbol{\Sigma}_{p\times p}, \boldsymbol{\Psi}_{q\times q}), \tag{4.5}$$

if and only if its vectorization has a multivariate normal distribution with the following covariance structure, $\mathrm{vec}(\mathbf{X}_i) \sim \mathcal{N}(\mathbf{0}_{pq}, \boldsymbol{\Psi} \otimes \boldsymbol{\Sigma})$,

Due to the equivalent definition and the property of the Kronecker product, one can easily see that the covariance of the ijth entry and the klth entry of the random matrix $\mathbf{X}$ is the product

of the ikth entry of Σ and the jlth entry of Ψ, i.e.

$$\mathrm{cov}(X_{ij}, X_{kl}) = \Sigma_{ik}\Psi_{jl},$$

which implies that the covariance can be decomposed as the product of two parts: the row covariance $\mathbf{\Sigma}$ and the column covariance $\mathbf{\Psi}$. Such row-column decomposition is reasonable for many applications, and is the most prevalent choice for distributions of a random matrix in recent literature (Allen and Tibshirani, 2012; Leng and Tang, 2012; Yin and Li, 2012; Zhou, 2014).

The reason that we choose the random design over the fixed design is two-fold. First, under the random design, the formula for the asymptotic covariance of the estimators are rather intuitive, as to be shown and discussed in Section 4.3, which offer nice interpretation and insights. On the other hand, if the fixed design were employed, the corresponding expressions would be rather involved and incomprehensible. This is due to the nice property of a normal distribution: its higher order moments are expressible through its first two moments. Second, it is well known that, for the standard linear regression, under the fixed design assumption, the covariance of the least squares estimator, which is also the maximum likelihood estimator, is $\tau^2(\mathbf{X}^T\mathbf{X})^{-1}$, which converges to $\tau^2\Sigma_{XX}^{-1}/n$, i.e. the same as its asymptotic covariance under the random design setting. Therefore, from this perspective, the choice of the design will not matter.

4.2.2 The maximum-likelihood based flip-flop estimator

We now derive the maximum likelihood estimator (MLE) under Model (4.2). Noting (4.5), the probability density function for the matrix normal $\mathbf{X}_i$ is

$$\begin{aligned} f(\mathbf{X}_i \mid \mathbf{\Sigma}, \mathbf{\Psi}) &= \frac{1}{(2\pi)^{pq/2}|\mathbf{\Psi} \otimes \mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}\mathrm{vec}^T(\mathbf{X}_i)(\mathbf{\Psi} \otimes \mathbf{\Sigma})^{-1}\mathrm{vec}(\mathbf{X}_i)\right) \\ &= \frac{1}{(2\pi)^{pq/2}|\mathbf{\Psi}|^{p/2}|\mathbf{\Sigma}|^{q/2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{X}_i\mathbf{\Psi}^{-1}\mathbf{X}_i^T)\right), \end{aligned}$$

which leads to the log-likelihood of $\mathbf{\Sigma}$ and $\mathbf{\Psi}$ given $\mathbf{X}_i$,

$$\ell(\mathbf{\Sigma}, \mathbf{\Psi}; \mathbf{X}_i) = -\frac{pq}{2}\log(2\pi) - \frac{q}{2}\log|\mathbf{\Sigma}| - \frac{p}{2}\log|\mathbf{\Psi}| - \frac{1}{2}\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{X}_i\mathbf{\Psi}^{-1}\mathbf{X}_i^T).$$

Then given n independent and identically distributed observations $\{\mathbf{X}_i, y_i\}_{i=1}^n$, the joint log-likelihood can be written as

$$\ell(\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0, \tau^2, \mathbf{\Sigma}, \mathbf{\Psi}; \mathbb{X}, \mathbf{y}) = -\frac{n}{2}\log(2\pi\tau^2) - \frac{\sum_{i=1}^n (y_i - \boldsymbol{\alpha}_0^T\mathbf{X}_i\boldsymbol{\beta}_0)^2}{2\tau^2} + \sum_{i=1}^n \ell(\mathbf{\Sigma}, \mathbf{\Psi}; \mathbf{X}_i). \quad (4.6)$$

In the above log-likelihood equation (4.6), the only term that contains $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ is the residual sum of squares $\sum_{i=1}^n (y_i - \boldsymbol{\alpha}_0^T\mathbf{X}_i\boldsymbol{\beta}_0)^2$. Hence, the maximum likelihood estimates of $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ can be obtained equivalently from the minimization of the residual sum of squares. Let $\hat{\boldsymbol{\alpha}}_{\mathrm{mle}}$ and $\hat{\boldsymbol{\beta}}_{\mathrm{mle}}$ denote the maximum likelihood estimates of $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ respectively. Mathematically, they can be derived from the following optimization problem,

$$(\hat{\boldsymbol{\alpha}}_{\mathrm{mle}}, \hat{\boldsymbol{\beta}}_{\mathrm{mle}}) = \underset{\boldsymbol{\alpha}\in\mathbb{R}^p,\ \boldsymbol{\beta}\in\mathbb{R}^q}{\mathrm{argmin}}\ h(\boldsymbol{\mu}), \quad (4.7)$$

where

$$h(\boldsymbol{\mu}) = h(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i=1}^n (y_i - \boldsymbol{\alpha}^T\mathbf{X}_i\boldsymbol{\beta})^2, \quad \text{with} \quad \boldsymbol{\mu} = [\boldsymbol{\alpha}^T\ \boldsymbol{\beta}^T]^T.$$

The objective function $h(\boldsymbol{\mu})$ is not convex with respect to $\boldsymbol{\mu}$, but is bi-convex with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The bi-convexity means that $h(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is convex with respect to $\boldsymbol{\alpha}$ (or $\boldsymbol{\beta}$) when $\boldsymbol{\beta}$ (or $\boldsymbol{\alpha}$) is fixed. A natural approach to solve a bi-convex optimization problem is to optimize over one while fixing the other. When $\boldsymbol{\beta}$ or $\boldsymbol{\alpha}$ is fixed, the bilinear model (4.2) reduces to the classical linear model with $\mathbf{X}_i\boldsymbol{\beta} \in \mathbb{R}^p$ or $\mathbf{X}_i^T\boldsymbol{\alpha} \in \mathbb{R}^q$ as the covariate vector and $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$ as the coefficient vector respectively. Therefore, we can solve for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ by iterating between the following two steps,

$$\begin{aligned} \boldsymbol{\alpha}(\boldsymbol{\beta}) = &\ (\tfrac{1}{n}\textstyle\sum_i^n \mathbf{X}_i\boldsymbol{\beta}\boldsymbol{\beta}^T\mathbf{X}_i^T)^{-1}(\tfrac{1}{n}\textstyle\sum_i^n \mathbf{X}_i\boldsymbol{\beta}y_i), \\ \boldsymbol{\beta}(\boldsymbol{\alpha}) = &\ (\tfrac{1}{n}\textstyle\sum_i^n \mathbf{X}_i^T\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{X}_i)^{-1}(\tfrac{1}{n}\textstyle\sum_i^n \mathbf{X}_i^T\boldsymbol{\alpha}y_i), \end{aligned} \tag{4.8}$$

where we have defined two operators $\boldsymbol{\alpha}(\cdot) : \mathbb{R}^q \mapsto \mathbb{R}^p$ and $\boldsymbol{\beta}(\cdot) : \mathbb{R}^p \mapsto \mathbb{R}^q$ that map $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ back and forth. Performing the mapping once decreases the value of the objective function defined in (4.7). In other words, the algorithm with iterations is a block descent algorithm.

To be more specific and to better distinguish between the current algorithm and the one to be proposed later in Section 4.2.3, the iterative algorithm works as follows. Let $\boldsymbol{\beta}_{(0)} = \boldsymbol{\beta}_{\text{init}}$ be the initialization of $\boldsymbol{\beta}$. We alternate the updates for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ by letting $\boldsymbol{\alpha}_{(2i+1)} = \boldsymbol{\alpha}(\boldsymbol{\beta}_{(2i)})$ and $\boldsymbol{\beta}_{(2i+2)} = \boldsymbol{\beta}(\boldsymbol{\alpha}_{(2i+1)})$ for $i = 0, 1, 2, \ldots$, and stop the updating process once $\|\boldsymbol{\alpha}_{(2i+1)} - \boldsymbol{\alpha}_{(2i-1)}\|_2^2$ and $\|\boldsymbol{\beta}_{(2i+2)} - \boldsymbol{\beta}_{(2i)}\|_2^2$ are both below a pre-specified tolerance level. The procedure is summarized below in Algorithm 4.1.

Differentiating the objective function (4.7) w.r.t. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and setting the derivatives to zero, one can easily verify that the global optimizer, the MLE, must satisfy the following stationary equations:

$$\hat{\boldsymbol{\alpha}}_{\text{mle}} = \boldsymbol{\alpha}(\hat{\boldsymbol{\beta}}_{\text{mle}}), \quad \text{and} \quad \hat{\boldsymbol{\beta}}_{\text{mle}} = \boldsymbol{\beta}(\hat{\boldsymbol{\alpha}}_{\text{mle}}), \tag{4.9}$$

where the operators $\boldsymbol{\alpha}(\cdot), \boldsymbol{\beta}(\cdot)$ are defined in (4.8). Since $\hat{\boldsymbol{\alpha}}_{\text{mle}}$ and $\hat{\boldsymbol{\beta}}_{\text{mle}}$ are functions of each other and intertwined, finding the MLE is a nontrivial task. Due to the non-convexity and biconvexity of the objective function, it may have several local minimums. The proposed

iterative procedure is not guaranteed to converge to the global minimizer, which is the MLE. Therefore, we name the estimates from the iterative algorithm after convergence, which are local stationary points, as the maximum-likelihood based *flip-flop estimator* and denote them by $\hat{\boldsymbol{\alpha}}_{\text{ff}}$, $\hat{\boldsymbol{\beta}}_{\text{ff}}$ and $\hat{\boldsymbol{\theta}}_{\text{ff}} = \hat{\boldsymbol{\beta}}_{\text{ff}} \otimes \hat{\boldsymbol{\alpha}}_{\text{ff}}$, respectively. The terminology *flip-flop* has been used extensively in research areas involving alternating procedures (Werner et al., 2008; Zhou, 2014).

Sample size. In order to uniquely update $\boldsymbol{\alpha}(\boldsymbol{\beta})$ and $\boldsymbol{\beta}(\boldsymbol{\alpha})$, we need the matrices $\sum_i^n \mathbf{X}_i \boldsymbol{\beta}\boldsymbol{\beta}^T \mathbf{X}_i^T$ and $\sum_i^n \mathbf{X}_i^T \boldsymbol{\alpha}\boldsymbol{\alpha}^T \mathbf{X}_i$ to be non-singular. These two matrices are sums of n rank-1 matrices $\mathbf{X}_i \boldsymbol{\beta}\boldsymbol{\beta}^T \mathbf{X}_i^T$ and $\mathbf{X}_i^T \boldsymbol{\alpha}\boldsymbol{\alpha}^T \mathbf{X}_i$ of size $p \times p$ and $q \times q$ respectively. Hence, this requires that n is at least as large as p and q, i.e.,

$$n \geq \max(p, q). \tag{4.10}$$

Recall that the sample size requirement for the general linear model to be identifiable is $n \geq pq$. We can see that the sample size requirement for the bilinear model can indeed be much smaller than that for the linear model when p or q is large.

Convergence. When $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$ is fixed, we write $h(\boldsymbol{\alpha}, \boldsymbol{\beta})$ as $h(\boldsymbol{\beta})$ or $h(\boldsymbol{\alpha})$ to reflect the fact that it is a function of $\boldsymbol{\beta}$ or $\boldsymbol{\alpha}$. Since $h(\boldsymbol{\alpha})$ and $h(\boldsymbol{\beta})$ are convex, $\boldsymbol{\alpha}(\boldsymbol{\beta})$ and $\boldsymbol{\beta}(\boldsymbol{\alpha})$ are the (global) minimizers of $h(\boldsymbol{\alpha})$ and $h(\boldsymbol{\beta})$. Hence, each updating step would reduce the value of the objective function $h(\boldsymbol{\alpha}, \boldsymbol{\beta})$. On the other hand, $h(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is bounded from below by 0. Hence, Algorithm 4.1 will converge as long as the sample size n satisfies the above requirement (4.10).

4.2.3 A truncated flip-flop estimator

The aforementioned flip-flop estimator involves an iterative procedure. When p and q are large, the computational load can be very intensive. For faster computation, we propose the following *truncated flip-flop estimator* which is partially motivated by the separable covariance matrix estimation problem (Werner et al., 2008).

Truncation means that we stop the flip-flop Algorithm 4.1 at $\boldsymbol{\beta}_{(2)}$ and $\boldsymbol{\alpha}_{(3)}$. Let $\hat{\boldsymbol{\alpha}}_{\text{tf}}$, $\hat{\boldsymbol{\beta}}_{\text{tf}}$, and $\hat{\boldsymbol{\theta}}_{\text{tf}} = \hat{\boldsymbol{\beta}}_{\text{tf}} \otimes \hat{\boldsymbol{\alpha}}_{\text{tf}}$ denote the truncated flip-flop estimators. In particular, they are defined as

Algorithm 4.1 The Flip-flop Estimation Algorithm

1. Randomly initialize $\boldsymbol{\beta}$ by $\boldsymbol{\beta}_{\text{init}}$, denote $\boldsymbol{\beta}_{(0)} = \boldsymbol{\beta}_{\text{init}}$ and let $i = 0$.

2. Given i, do the following

$$\begin{aligned}
\boldsymbol{\alpha}_{(2i+1)} &= \boldsymbol{\alpha}(\boldsymbol{\beta}_{(2i)}) = (\frac{1}{n}\sum_{i}^{n}\mathbf{X}_i\boldsymbol{\beta}_{(2i)}\boldsymbol{\beta}_{(2i)}^T\mathbf{X}_i^T)^{-1}(\frac{1}{n}\sum_{i}^{n}\mathbf{X}_i\boldsymbol{\beta}_{(2i)}y_i), \\
\boldsymbol{\beta}_{(2i+2)} &= \boldsymbol{\beta}(\boldsymbol{\alpha}_{(2i+1)}) = (\frac{1}{n}\sum_{i}^{n}\mathbf{X}_i^T\boldsymbol{\alpha}_{(2i+1)}\boldsymbol{\alpha}_{(2i+1)}^T\mathbf{X}_i)^{-1}(\frac{1}{n}\sum_{i}^{n}\mathbf{X}_i^T\boldsymbol{\alpha}_{(2i+1)}y_i).
\end{aligned}$$

3. If $\|\boldsymbol{\alpha}_{(2i+1)} - \boldsymbol{\alpha}_{(2i-1)}\|_2^2$ and $\|\boldsymbol{\beta}_{(2i+2)} - \boldsymbol{\beta}_{(2i)}\|_2^2$ are both larger than a pre-specified threshold, update i to be $i+1$ and go to step (2). Otherwise, output $\hat{\boldsymbol{\alpha}}_{\text{ff}} = \boldsymbol{\alpha}_{(2i+1)}$ and $\hat{\boldsymbol{\beta}}_{\text{ff}} = \boldsymbol{\beta}_{(2i+2)}$.

follows,

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{\text{tf}} &= \boldsymbol{\beta}_{(2)} = \boldsymbol{\beta}(\boldsymbol{\alpha}(\boldsymbol{\beta}_{\text{init}})), \\
\hat{\boldsymbol{\alpha}}_{\text{tf}} &= \boldsymbol{\alpha}_{(3)} = \boldsymbol{\alpha}(\hat{\boldsymbol{\beta}}_{\text{tf}}) = \boldsymbol{\alpha}(\boldsymbol{\beta}(\boldsymbol{\alpha}(\boldsymbol{\beta}_{\text{init}})),
\end{aligned}$$

where $\boldsymbol{\beta}_{\text{init}}$ is an initialization of $\boldsymbol{\beta}_0$.

This truncated flip-flop estimator only consists of three iteration steps. Hence, it can significantly improve the computational speed especially when p and q are large. Asymptotically, the theoretical studies in Section 4.3 show that the truncated flip-flop estimator achieves the same asymptotic covariance matrix as the flip-flop estimator, both of which are much more efficient than the linear regression estimator. As for finite sample performance, the simulation studies in Section 4.5 confirm that the truncated flip-flop estimation performs comparably as the flip-flop estimator.

4.3 Asymptotic properties

4.3.1 Consistency and asymptotic covariance

Henceforth, we denote the asymptotic covariance matrix of any estimator $\hat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}_0$ by $\mathrm{acov}(\hat{\boldsymbol{\theta}}) = \lim_{n\to\infty} \mathrm{cov}(\hat{\boldsymbol{\theta}})$. The asymptotic property of the flip-flop estimator is given by the following Theorem 3. Since the flip-flop estimator is not necessarily the maximum likelihood estimator (MLE), the proof of the theorem does not directly follow from the property of MLE. We obtain the asymptotic covariance of the flip-flop estimator through deriving the first order expansion of the stationary equations (4.9) and computing the covariance of the leading terms.

Theorem 3 *Let $\hat{\boldsymbol{\theta}}_{\mathrm{ff}}$ denote the flip-flop estimator of $\boldsymbol{\theta}_0$ given by Algorithm 4.1 under Model (4.2) and Assumption (4.5). Then $\hat{\boldsymbol{\theta}}_{\mathrm{ff}}$ is a consistent estimator of $\boldsymbol{\theta}_0$, and its asymptotic covariance is given by*

$$\begin{aligned}\mathrm{acov}(\hat{\boldsymbol{\theta}}_{\mathrm{ff}}) &= \frac{\tau^2}{n}\big((\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}\boldsymbol{\Psi}^{-1}\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T) + (\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T)\otimes\boldsymbol{\Sigma}^{-1} \\ &\quad -(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T)\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T)\big)\,.\end{aligned}$$

The right hand side (RHS) consists of three terms, which can be interpreted as follows.

1. To gain insight of the first term, we recall that for the classical linear regression model (4.1) with random design $\mathbf{x}_i \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, the ordinary least square/MLE estimator $\hat{\boldsymbol{\beta}}$ is consistent and has asymptotic covariance $\frac{\tau^2}{n}\boldsymbol{\Sigma}^{-1}$. In the bilinear model (4.2), suppose that $\boldsymbol{\alpha}_0$ is known and we are interested in estimating $\boldsymbol{\beta}_0$. Then the covariates become $\mathbf{a}_i = \mathbf{X}_i^T\boldsymbol{\alpha}_0$ with a multivariate normal distribution $\mathcal{N}(\mathbf{0}, (\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)\boldsymbol{\Psi})$, due to the property of the matrix normal assumption (4.5) on $\mathbf{X}_i$. In addition, the bilinear model reduces to the classical linear regression model $y_i = \mathbf{a}_i^T\boldsymbol{\beta}_0 + \epsilon_i$. It is straightforward to realize that the MLE of $\boldsymbol{\beta}_0$ when $\boldsymbol{\alpha}_0$ is known, say $\hat{\boldsymbol{\beta}}_{\alpha_0}$, is consistent and has asymptotic co-

variance $\mathrm{acov}(\hat{\boldsymbol{\beta}}_{\alpha_0}) = (\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}\boldsymbol{\Psi}^{-1}\tau^2/n$. Under the assumption that $\boldsymbol{\alpha}_0$ is known, the estimator for $\boldsymbol{\theta}_0 = \boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0$, say $\hat{\boldsymbol{\theta}}_{\alpha_0}$, is simply the Kronecker product of the estimator of $\boldsymbol{\beta}_0$ and the known $\boldsymbol{\alpha}_0$, i.e., $\hat{\boldsymbol{\theta}}_{\alpha_0} = \hat{\boldsymbol{\beta}}_{\alpha_0} \otimes \boldsymbol{\alpha}_0$. Due to the property of the Kronecker product, it can be easily seen that $\hat{\boldsymbol{\theta}}_{\alpha_0}$ is consistent and has asymptotic covariance $\mathrm{acov}(\hat{\boldsymbol{\theta}}_{\alpha_0}) = (\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}\boldsymbol{\Psi}^{-1} \otimes (\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T)\tau^2/n$, which is identical to the first term of the RHS.

2. Similarly, exchanging the roles of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in the above discussion produces the asymptotic covariance of estimating $\boldsymbol{\theta}_0$ when $\boldsymbol{\beta}_0$ is known, which is the second term on the RHS.

3. The last term $(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T) \otimes (\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T)$ is a positive semi-definite matrix and is subtracted from the first two terms. The appearance of this term reveals that the asymptotic covariance of the estimator of the bilinear model is *not* the simple sum of the asymptotic covariances for estimating one of the unknown coefficient vectors while assuming the knowledge of the other, it is less than the sum as a matter of fact. This is somewhat surprising at first sight. Yet, when we recall that $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ are only identifiable up to a scale factor and that the first two terms on the RHS of Theorem 3 estimate the scaling factor twice, it becomes sensible to eliminate the redundancy, which is the purpose of the third term.

   The third term corresponds to the case when we know the directions of both of the two coefficient vectors $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ but not their scale, i.e., the bilinear model reduces to a simply linear regression model without intercept,

$$y_i = \nu\boldsymbol{\alpha}_0^T\mathbf{X}_i\boldsymbol{\beta}_0 + \epsilon_i,$$

   where $\nu \in \mathbb{R}$ is the only unknown scalar parameter, and $\boldsymbol{\alpha}_0^T\mathbf{X}_i\boldsymbol{\beta}_0$ is the one dimensional predictor that has a univariate normal distribution $\mathcal{N}(0, (\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0))$. Again by

the property of the MLE of the classical linear regression model in the random design setting, we have the consistency of the estimator $\hat{\nu}$ and its asymptotic variance $\mathrm{acov}(\hat{\nu}) = (\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}\tau^2/n$. It follows that $\hat{\boldsymbol{\theta}}_{\boldsymbol{\alpha}_0,\boldsymbol{\beta}_0}$, the estimator under the current setup, should be $\hat{\nu}\boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0$, which is a consistent estimator of $\boldsymbol{\theta}_{\boldsymbol{\alpha}_0,\boldsymbol{\beta}_0} = \nu\boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0$ with asymptotic covariance matrix $\mathrm{acov}(\hat{\boldsymbol{\theta}}_{\boldsymbol{\alpha}_0,\boldsymbol{\beta}_0}) = \tau^2/n(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T) \otimes (\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T)$, which is the same as the third term on the RHS.

As discussed above, the three-term decomposition of the asymptotic covariance of the flip-flop estimator is intuitive and meaningful. It helps us better understand the nature and difficulty of the bilinear problem.

We want to emphasize that Theorem 3 holds for all flip-flop estimators whenever Algorithm 4.1 converges, which may or may not converge to the global minimum of the objective function, and hence may or may not reach the MLE. In other words, Theorem 3 provides some theoretical justification and ensures that one initialization, no matter how to initialize, is sufficient in the sense of achieving least asymptotic covariance, although in practice people may still try a few initial points and hope to obtain the global optimal, which however cannot be guaranteed.

Similarly, the following Theorem 4 states the asymptotic properties of the truncated estimator.

Theorem 4 *Let $\hat{\boldsymbol{\theta}}_{\mathrm{tf}}$ denote the truncated flip-flop estimator of $\boldsymbol{\theta}_0$ under Model (4.2) and Assumption (4.5). Then $\hat{\boldsymbol{\theta}}_{\mathrm{tf}}$ is consistent for $\boldsymbol{\theta}_0$, and its asymptotic covariance is given by*

$$\begin{aligned}\mathrm{acov}(\hat{\boldsymbol{\theta}}_{\mathrm{tf}}) &= \frac{\tau^2}{n}\left((\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}\boldsymbol{\Psi}^{-1} \otimes (\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T) + (\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T) \otimes \boldsymbol{\Sigma}^{-1}\right. \\ &\quad \left. -(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T) \otimes (\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T)\right).\end{aligned}$$

Theorem 4 has two implications that are worth noting.

First of all, the asymptotic covariances of the flip-flop estimator and the truncated flip-flop estimator are identical. Even though the truncated estimator terminates the flip-flop algorithm

after merely three iteration steps, it is asymptotically as efficient as the iterative algorithm reaching convergence. Therefore, computationally, the truncated flip-flop estimator is much more appealing for large sample size n, because each iteration step involves the inversion of either a $p \times p$ or $q \times q$ matrix, which can be rather time-consuming for relatively large p or q.

Secondly, although the truncated flip-flop estimator consists of only three steps, its performance does not depend the initialization after all, which is surprising to some extent. In fact, the effects of the initialization on the estimations of $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ cancel each other after three steps. In comparison, if the flip-flop algorithm is discontinued after two steps, i.e., we consider the new estimator $\hat{\boldsymbol{\theta}} = \boldsymbol{\beta}(\boldsymbol{\alpha}(\boldsymbol{\beta}_{\text{init}})) \otimes \boldsymbol{\alpha}(\boldsymbol{\beta}_{\text{init}})$, the cancellation will not occur and it can be shown that the asymptotic covariance of this estimator would rely upon the initialization $\boldsymbol{\beta}_{\text{init}}$.

The same two phenomena arise in the context of covariance matrix estimation (Werner et al., 2008) and precision matrix estimation (Zhou, 2014) under the assumption of separable covariance matrix. Although our bilinear regression problem is apparently very distinct from theirs, we all adopt the usage of matrix variate normal distribution. Hence our conjecture is that this separable covariance assumption is the origin of the above interesting phenomena.

In spite of the connection and the analogous phenomena among the three problems, there exist certain significant differences in the technical results. We will show in the next theorem that our flip-flop estimator achieves the information lower bound. In comparison, Werner et al. (2008) did not show that the asymptotic covariance matched the lower bound for covariance matrix estimation, and Zhou (2014) did not provide any lower bound.

4.3.2 Cramér-Rao lower bound and the MLE

The Cramér-Rao lower bound (CRLB) establishes a lower bound on the variance of an estimator. Ideally, the inverse of the Fisher information matrix with respect to $\boldsymbol{\theta}$, which has the special structural assumption $\boldsymbol{\beta} \otimes \boldsymbol{\alpha}$, can be used directly as the CRLB. Since the bilinear regression model is expressed in terms of $\boldsymbol{\mu} = (\boldsymbol{\alpha}^T, \boldsymbol{\beta}^T)^T$, it is natural to use the chain rule

and the Fisher information matrix with respect to $\boldsymbol{\mu}$ to derive the CRLB. Nonetheless, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are identifiable only up to a scale factor in the bilinear model, the Fisher information matrix with respect to $\boldsymbol{\mu}$ is rank deficient and hence non-invertible. By using results on parameter estimation with singular information matrices in Stoica and Marzetta (2001b), we can obtain the desired CRLB in the following theorem.

Theorem 5 *The Cramér-Rao lower bound for any unbiased estimator $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}_0$ under Model (4.2) and Assumption (4.5) is given by*

$$\operatorname{acov}(\hat{\boldsymbol{\theta}}) \succeq \mathbf{A}\mathbf{J}(\boldsymbol{\mu})^{\dagger}\mathbf{A}^T, \tag{4.11}$$

where the matrix inequality $\mathbf{A} \succeq \mathbf{B}$ is understood to mean that the matrix $\mathbf{A} - \mathbf{B}$ is positive semi-definite, † denotes the Moore-Penrose pseudoinverse, $\mathbf{A}$ is the Jacobian matrix

$$\mathbf{A} = \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\mu}^T} = [\boldsymbol{\beta} \otimes \mathbf{I}_{p\times p} \quad \mathbf{I}_{q\times q} \otimes \boldsymbol{\alpha}], \tag{4.12}$$

and $\mathbf{J}(\boldsymbol{\mu})$ is the Fisher information matrix when the model is parameterized by $\boldsymbol{\mu}$:

$$\mathbf{J}(\boldsymbol{\mu}) = \frac{n}{\tau^2}\begin{pmatrix} (\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)\boldsymbol{\Sigma} & \boldsymbol{\Sigma}\boldsymbol{\alpha}_0\boldsymbol{\beta}_0^T\boldsymbol{\Psi} \\ \boldsymbol{\Psi}\boldsymbol{\beta}_0\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma} & (\boldsymbol{\alpha}_0\boldsymbol{\Sigma}\boldsymbol{\alpha}_0^T)\boldsymbol{\Psi} \end{pmatrix}. \tag{4.13}$$

Moreover, when the expressions (4.12) and (4.13) are plugged back into the general formula (4.11) and the pseudoinverse is completely computed, we have the following explicit CRLB,

$$\begin{aligned} \operatorname{acov}(\hat{\boldsymbol{\theta}}) \;\succeq\; & \frac{\tau^2}{n}\big((\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}\boldsymbol{\Psi}^{-1} \otimes (\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T) + (\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T) \otimes \boldsymbol{\Sigma}^{-1} \\ & -(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T) \otimes (\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T)\big). \end{aligned} \tag{4.14}$$

Note that the CRLB in Theorem 5, Equation (4.14), is attained by both the flip-flop estimator and the truncated estimator asymptotically, which shows that both estimators are efficient.

We next examine whether the maximum likelihood estimator (MLE), i.e., the global optimizer of the objective function (4.7) if obtainable numerically, achieves the CRLB as well. It is well known that the MLE possesses a few attractive limiting properties, efficiency being one of them. However, when over-parameterization occurs, we encounter the same problem of the singularity of the Fisher information matrix as in the derivation of the CRLB. Nevertheless, $\boldsymbol{\theta}_0$ is identifiable and estimable, which enables us to apply the techniques described in Shapiro (1986) to derive the asymptotic distribution of the MLE.

Theorem 6 *Suppose Model (4.2) and Assumption (4.5) hold. Let* $\hat{\boldsymbol{\theta}}_{\mathrm{mle}} = \hat{\boldsymbol{\beta}}_{\mathrm{mle}} \otimes \hat{\boldsymbol{\alpha}}_{\mathrm{mle}}$ *where* $\hat{\boldsymbol{\beta}}_{\mathrm{mle}}$ *and* $\hat{\boldsymbol{\beta}}_{\mathrm{mle}}$ *are defined in (4.7). Then we have the following convergence in distribution*

$$\hat{\boldsymbol{\theta}}_{\mathrm{mle}} - \boldsymbol{\theta}_0 \to \mathcal{N}(\mathbf{0}, \mathbf{A}(\mathbf{A}^T \mathbf{J}(\boldsymbol{\theta}) \mathbf{A})^{\dagger} \mathbf{A}^T),$$

where $\mathbf{A}$ *is the Jacobian matrix defined in Theorem 5, and* $\mathbf{J}(\boldsymbol{\theta})$ *is the Fisher information matrix with respect to* $\boldsymbol{\theta}$ *when no structural assumption* $\boldsymbol{\theta} = \boldsymbol{\beta} \otimes \boldsymbol{\alpha}$ *is assumed, i.e.,* $\mathbf{J}(\boldsymbol{\theta}) = \frac{n}{\tau^2} \boldsymbol{\Psi} \otimes \boldsymbol{\Sigma}$. *Furthermore, the asymptotic covariance of the maximum likelihood estimator, when fully expanded, coincides with the CRLB in Theorem 5.*

Theorems 3-6 altogether depict the following overall picture. The MLE is the most favorable choice due to its statistical advantages, including obtaining the lowest asymptotic mean squared error among all consistent estimators. However, it is computationally intractable, which makes us resort to the flip-flop estimator. It turns out that the flip-flop estimator as a surrogate can achieve the information lower bound, even though it is only a stationary point. More surprisingly, the truncated flip-flop estimator, another alternative to the MLE, which is computationally even more competitive, is as efficient as the MLE and the flip-flop estimator.

4.3.3 Asymptotic efficiency

All of the three estimators, the flip-flop, the truncated flip-flop, and the maximum likelihood estimators, are asymptotically optimal. It is also interesting to investigate their performance compared to the general linear regression model with the vector covariate being the vectorization of the original matrix covariate. Let $\hat{\boldsymbol{\theta}}_{\mathrm{lm}}$ denote the estimator of $\boldsymbol{\theta}_0$ in the linear regression model (4.4). It is well known that $\hat{\boldsymbol{\theta}}_{\mathrm{lm}}$ is a consistent estimator of $\boldsymbol{\theta}_0$, with its asymptotic covariance as $\mathrm{acov}(\hat{\boldsymbol{\theta}}_{\mathrm{lm}}) = \tau^2/n\boldsymbol{\Psi}^{-1} \otimes \boldsymbol{\Sigma}^{-1}$. Then the following theorem states that the three estimators of $\boldsymbol{\theta}_0$ mentioned above are asymptotically more efficient than the linear regression estimator.

Theorem 7 *Let $\hat{\boldsymbol{\theta}}_{\mathrm{ff}}$, $\hat{\boldsymbol{\theta}}_{\mathrm{tf}}$, $\hat{\boldsymbol{\theta}}_{\mathrm{mle}}$ and $\hat{\boldsymbol{\theta}}_{\mathrm{lm}}$ be the flip-flop, truncated flip-flop, maximum likelihood, and linear regression estimators of $\boldsymbol{\theta}_0$ under Model (4.2) and Assumption* (4.5)*. Then we have $\mathrm{acov}(\hat{\boldsymbol{\theta}}_{\mathrm{lm}}) \succeq \mathrm{acov}(\hat{\boldsymbol{\theta}}_{\mathrm{ff}}) = \mathrm{acov}(\hat{\boldsymbol{\theta}}_{\mathrm{tf}}) = \mathrm{acov}(\hat{\boldsymbol{\theta}}_{\mathrm{mle}})$, where the matrix inequality is defined in Theorem 5.*

Theorem 7 shows the efficiency gain when taking the Knocker product structure into consideration, assuming that the bilinear model is true and the matrix covariate follows a matrix normal distribution. Simulation results in Section 4.5 further demonstrate that the two bilinear estimators, $\hat{\boldsymbol{\theta}}_{\mathrm{ff}}$ and $\hat{\boldsymbol{\theta}}_{\mathrm{tf}}$, both of which are computable in reality, perform better than the linear estimator $\hat{\boldsymbol{\theta}}_{\mathrm{lm}}$ for finite sample size.

4.4 Bilinear ridge regression

The flip-flop and truncated flip-flop estimators require that $n > \max(p, q)$. In practice, n is sometimes comparable to or even smaller than p or q. To overcome the limitation of the sample size, we propose a bilinear ridge estimator in this section.

4.4.1 Bilinear ridge estimator

With ridge penalties on both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the bilinear ridge estimator are the solutions to the following optimization problem,

$$\begin{aligned}&\text{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^p,\boldsymbol{\beta}\in\mathbb{R}^q}\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\boldsymbol{\alpha}^T\mathbf{X}_i\boldsymbol{\beta}\right)^2\\&+\lambda_\alpha\|\boldsymbol{\beta}\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\alpha}\|_2^2+\lambda_\beta\|\boldsymbol{\alpha}\|_{\boldsymbol{\Sigma}}^2\|\boldsymbol{\beta}\|_2^2+\lambda_\alpha\lambda_\beta\|\boldsymbol{\alpha}\|_2^2\|\boldsymbol{\beta}\|_2^2,\end{aligned}\tag{4.15}$$

where $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Sigma}}^2=\boldsymbol{\alpha}^T\boldsymbol{\Sigma}\boldsymbol{\alpha}$ and $\|\boldsymbol{\beta}\|_{\boldsymbol{\Psi}}^2=\boldsymbol{\beta}^T\boldsymbol{\Psi}\boldsymbol{\beta}$ are two vector norms.

The objective function in this optimization problem (4.15) is also a bi-convex function. Similar to the optimization problem (4.7), we solve this bi-convex optimization problem through an iterative approach. We denote the estimators by $(\hat{\boldsymbol{\alpha}}_\lambda,\hat{\boldsymbol{\beta}}_\lambda)$ and name it as flip-flip bilinear ridge estimator. In parallel to the operators in (4.8), we define the following two operators for the flip-flop bilinear ridge estimator,

$$\begin{aligned}\boldsymbol{\alpha}_\lambda(\boldsymbol{\beta})=&\ \left(\tfrac{1}{n}\textstyle\sum_i^n\mathbf{X}_i\boldsymbol{\beta}\boldsymbol{\beta}^T\mathbf{X}_i^T+\lambda_\alpha\|\boldsymbol{\beta}\|_{\boldsymbol{\Psi}}^2\mathbf{I}+\lambda_\beta\|\boldsymbol{\beta}\|_2^2\boldsymbol{\Sigma}+\lambda_\alpha\lambda_\beta\|\boldsymbol{\beta}\|_2^2\mathbf{I}\right)^{-1}\left(\tfrac{1}{n}\textstyle\sum_i^n\mathbf{X}_i\boldsymbol{\beta}y_i\right),\\\boldsymbol{\beta}_\lambda(\boldsymbol{\alpha})=&\ \left(\tfrac{1}{n}\textstyle\sum_i^n\mathbf{X}_i^T\boldsymbol{\alpha}\boldsymbol{\alpha}^T\mathbf{X}_i+\lambda_\alpha\|\boldsymbol{\alpha}\|_2^2\boldsymbol{\Psi}+\lambda_\beta\|\boldsymbol{\alpha}\|_{\boldsymbol{\Sigma}}^2\mathbf{I}+\lambda_\alpha\lambda_\beta\|\boldsymbol{\alpha}\|_2^2\mathbf{I}\right)^{-1}\left(\tfrac{1}{n}\textstyle\sum_i^n\mathbf{X}_i^T\boldsymbol{\alpha}y_i\right),\end{aligned}\tag{4.16}$$

With the above two operators, the iterative procedure to solve the optimization problem (4.15) is now given in Algorithm 4.2.

Algorithm 4.2 The Flip-flop Bilinear Ridge Estimation Algorithm

1. Initialize $\boldsymbol{\beta}_{(0)}$ by $\boldsymbol{\beta}_{\text{init}}$.
2. Given i, do the following

   $$\boldsymbol{\alpha}_{\text{r}(2i+1)}=\boldsymbol{\alpha}_\lambda(\boldsymbol{\beta}_{\text{r}(2i)}),\quad\text{and}\quad\boldsymbol{\beta}_{\text{r}(2i+2)}=\boldsymbol{\beta}_\lambda(\boldsymbol{\alpha}_{\text{r}(2i+1)}),$$

   where the operators $\boldsymbol{\alpha}_\lambda(\cdot)$ and $\boldsymbol{\beta}_\lambda(\cdot)$ are given in (4.16).
3. If the errors of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ between two iteration steps are below the threshold, the algorithm stops. Otherwise, go to step 2.

4.4.2 Theoretical properties of the flip-flop bilinear ridge estimator

In this section, we study the theoretical properties of the flip-flop bilinear ridge estimator through the excess prediction error. We start with introducing some more notations.

The eigen-decompositions of $\mathbf{\Sigma}$ and $\mathbf{\Psi}$ are denoted as follows,

$$\mathbf{\Sigma} = \sum_{j=1}^{p} \mu_j \mathbf{u}_j \mathbf{u}_j^T \quad \text{and} \quad \mathbf{\Psi} = \sum_{k=1}^{q} \nu_k \mathbf{v}_k \mathbf{v}_k^T,$$

where $\mu_1 \geq \mu_2 \geq \cdots \geq 0$ and $\nu_1 \geq \nu_2 \geq \cdots \geq 0$ are eigen-values of $\mathbf{\Sigma}$ and $\mathbf{\Psi}$ respectively, and $\mathbf{u}_j$ for $j = 1, 2, \ldots, p$ and $\mathbf{v}_k$ for $k = 1, 2, \ldots, q$ are eigen-vectors of $\mathbf{\Sigma}$ and $\mathbf{\Psi}$ respectively.

Define the following two quantities,

$$d_{\boldsymbol{\alpha}} = \sum_{j=1}^{p} \left(\frac{\mu_j}{\mu_j + \lambda_\alpha} \right)^2, \quad \text{and} \quad d_{\boldsymbol{\beta}} = \sum_{k=1}^{q} \left(\frac{\nu_k}{\nu_k + \lambda_\beta} \right)^2. \tag{4.17}$$

It can be seen that $d_{\boldsymbol{\alpha}} \leq p$ and $d_{\boldsymbol{\beta}} \leq q$. The $d_{\boldsymbol{\alpha}}$ and $d_{\boldsymbol{\beta}}$ are called effective dimensions in literature (e.g., Hsu et al., 2014) and they play an important role in the development of the convergence rate of the excess prediction error.

Define $\bar{\boldsymbol{\alpha}}$ and $\bar{\boldsymbol{\beta}}$ as follows.

$$\begin{aligned} \bar{\boldsymbol{\alpha}} &= (\mathbf{\Sigma} + \lambda_\alpha \mathbf{I})^{-1} \mathbf{\Sigma} \boldsymbol{\alpha}_0, \\ \bar{\boldsymbol{\beta}} &= (\mathbf{\Psi} + \lambda_\beta \mathbf{I})^{-1} \mathbf{\Psi} \boldsymbol{\beta}_0. \end{aligned}$$

Note that $\bar{\boldsymbol{\alpha}}$ and $\bar{\boldsymbol{\beta}}$ does not depend on the training data $\{(\mathbf{X}_i, y_i), i = 1, 2, \ldots, n\}$ and hence are deterministic. They characterize the limiting behavior of $\hat{\boldsymbol{\alpha}}_\lambda$ and $\hat{\boldsymbol{\beta}}_\lambda$ which will be seen in the proof of Theorem 8.

Now we show the theoretical properties of the flip-flop bilinear ridge estimators through the excess prediction error which is a quantity that measures the prediction accuracy. Let $(\tilde{\mathbf{X}}, \tilde{y})$ be a new pair of measurements independent of $(\mathbf{X}_i, y_i)$ for $i = 1, 2, \ldots, n$. Suppose $\hat{\boldsymbol{\alpha}}_\lambda$ and $\hat{\boldsymbol{\beta}}_\lambda$

are known and take expectation with respect to $\tilde{\mathbf{X}}$ and $\tilde{y}$, the excess prediction error is given by

$$\mathrm{E}\left((\tilde{y}-\hat{\boldsymbol{\alpha}}_\lambda^T\tilde{\mathbf{X}}\hat{\boldsymbol{\beta}}_\lambda)^2-(\tilde{y}-\boldsymbol{\alpha}_0^T\tilde{\mathbf{X}}\boldsymbol{\beta}_0)^2\right)=\|\hat{\boldsymbol{\beta}}_\lambda\otimes\hat{\boldsymbol{\alpha}}_\lambda-\boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|^2_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}.$$

The following theorem shows the rate of convergence of the excess prediction error.

Theorem 8 *Suppose λ_α and λ_β satisfy*

$$n^{-1}d_{\boldsymbol{\alpha}}=o(1),\quad \textit{and}\quad n^{-1}d_{\boldsymbol{\beta}}=o(1),$$

where $d_{\boldsymbol{\alpha}}$ and $d_{\boldsymbol{\beta}}$ are given in (4.17). Then the excess prediction error has the following upper bound,

$$\begin{aligned}&\mathrm{E}\|\hat{\boldsymbol{\beta}}_\lambda\otimes\hat{\boldsymbol{\alpha}}_\lambda-\boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|^2_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}\\ \leq\ & 2\|\bar{\boldsymbol{\beta}}\otimes\bar{\boldsymbol{\alpha}}-\boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|^2_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}\\ &+O\left(\frac{d_{\boldsymbol{\alpha}}+d_{\boldsymbol{\beta}}}{n}(\tau^2+\|\bar{\boldsymbol{\beta}}\otimes\bar{\boldsymbol{\alpha}}-\boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|^2_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}})\right).\end{aligned}\tag{4.18}$$

There are no explicit conditions for the matrix dimensions p and q, but implicit conditions through $d_{\boldsymbol{\alpha}}$ and $d_{\boldsymbol{\beta}}$, the effective dimensions. The conditions $n^{-1}d_{\boldsymbol{\alpha}}=o(1)$ and $n^{-1}d_{\boldsymbol{\beta}}=o(1)$ state that the excess prediction error converges as long as $d_{\boldsymbol{\alpha}}$ and $d_{\boldsymbol{\beta}}$ grow slower than n.

The upper bound contains two terms. The first term, $\|\bar{\boldsymbol{\beta}}\otimes\bar{\boldsymbol{\alpha}}-\boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|^2_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}$, on the right hand side is deterministic. The second term is because of the random design and the noise. It describes that the rate of convergence is $n^{-1}(d_{\boldsymbol{\alpha}}+d_{\boldsymbol{\beta}})$, which is the sum of effects from the row and column dimensions.

4.5 Simulations

In this section, we show the results of numerical experiments to compare the performance of the proposed flip-flop and truncated flip-flop estimators with the linear regression estimator.

4.5.1 Simulation setup

Throughout this section, the data are generated according to Model (4.2). We consider the following four simulation setups.

- **Model I** Both $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ follow the $N(\mathbf{0}, \mathbf{I})$ distribution, i.e., all of the entries of $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ are independent standard normal and they are mutually independent. $\boldsymbol{\Sigma}$ and $\boldsymbol{\Psi}$ are both identity matrices.

- **Model II** $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ are simulated in the way as in Model I. The ij-th entry of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Psi}$ are $\sigma_{ij} = 0.3^{|i-j|}$ and $\psi_{ij} = 0.5^{|i-j|}$ respectively. In this way, the magnitude of the off-diagonal entries decay as they move away from the diagonal, which ensures that the entries of $\mathbf{X}_i$ that are further apart are less correlated. This auto-regressive type of correlation structure is reasonable for many real applications.

- **Model III** $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ are simulated as smooth curves: the ith entry of $\boldsymbol{\alpha}_0$ is $\cos(2\pi i/p)$ and the jth entry of $\boldsymbol{\beta}_0$ is $\sin(2\pi j/q)$ respectively. $\boldsymbol{\Sigma}$ and $\boldsymbol{\Psi}$ are simulated the same way as in Model II.

- **Model IV** $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ are set to have linear forms: $\boldsymbol{\alpha}_0 = [1, 2, \ldots, p]^T$ and $\boldsymbol{\beta}_0 = [1, 2, \ldots, q]^T$. $\boldsymbol{\Sigma}$ and $\boldsymbol{\Psi}$ are the same as in Models II and III.

Moreover, $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ are normalized so that they have unit length: $\|\boldsymbol{\alpha}_0\|_2 = 1$ and $\|\boldsymbol{\beta}_0\|_2 = 1$. Hence, the four models are comparable to each other.

As for the covariate matrix $\mathbf{X}_i$, we generate it from $\mathbf{X}_i = \boldsymbol{\Sigma}^{1/2}\mathbf{Z}_i\boldsymbol{\Psi}^{1/2}$, where the entries of $\mathbf{Z}_i$ are i.i.d. standard normal random variables. From the basic property of matrix normal distributions, $\mathbf{X}_i$ follows a matrix normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}, \boldsymbol{\Psi})$.

We choose to fix the value of p to be 10, and let the number of columns q range in $\{10, 20, 40\}$, so that different aspect ratios are included. Regarding to the noise part ϵ_i, we take it to be $N(0, \tau^2)$. The choice of τ can be back traced in a way that makes the signal

to noise ratio snr range in $\{.5, 1, 2, 4\}$, where snr is defined as $snr = \text{var}(\boldsymbol{\alpha}_0^T \mathbf{X}_i \boldsymbol{\beta}_0)/\tau^2 = (\boldsymbol{\alpha}_0^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_0)(\boldsymbol{\beta}_0^T \boldsymbol{\Psi} \boldsymbol{\beta}_0)/\tau^2$ for a given setup with specific $\boldsymbol{\alpha}_0$, $\boldsymbol{\beta}_0$, $\boldsymbol{\Sigma}$, $\boldsymbol{\Psi}$ and τ^2.

For each simulation setup, we vary the size of the training data to be $\{1000, 2000, 5000, 10000\}$, and fix the size of the test data at 1000. We ran 100 simulations, apply each algorithm under comparison, and summarize the results using the following two performance measures. For estimation accuracy, we calculate the l_2 distance between the estimated coefficient vector and the truth,

$$D = \|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\|_2.$$

For out-of-sample prediction accuracy, we use the test data to evaluate the mean squared prediction error

$$MSPE = \mathsf{average}(y_i - \widehat{y}_i)^2.$$

4.5.2 The effects of algorithm initialization

Both the flip-flop and truncated flip-flop estimators start from a random initialization. A natural question to address is whether the initialization affects the estimation accuracy, and how we should choose the initials. We study this question under Model I, with $snr = 1$, $q = 20$, and the sample size n ranging in $\{1000, 2000, 5000, 10000\}$. For each value of n, we simulated n training samples and randomly generated 100 initials. The results of the ℓ_2 distances between the estimates and the truth are shown in Figure 4.1.

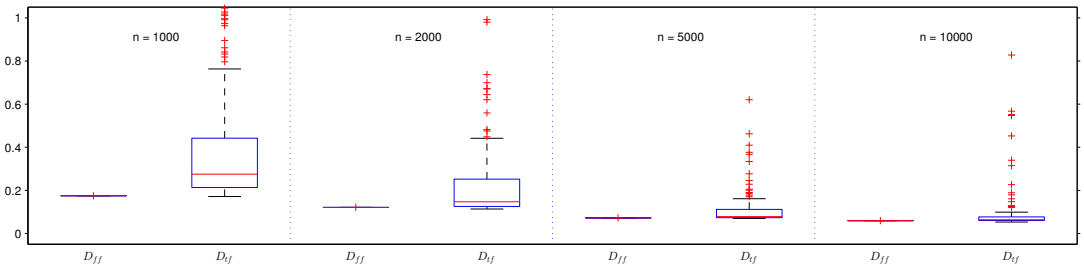


Figure 4.1: The simulation results for 100 random initials under Model I, with $snr = 1$ and $q = 20$. D is the ℓ_2 norm between the true and estimated coefficient vectors.

Fig. 4.1 shows that the flip-flop estimator, D_{ff}, does not depend on the initial. For each n, the 100 randomly generated initials always result in the same ℓ_2 distance. This is primarily because the flip-flop estimator is a convergent result of the algorithm.

On the contrary, the truncated flip-flop estimator, D_{tf}, highly depends on the initial, and this dependence decreases with increasing sample size n. Recall that the truncated flip-flop algorithm stops after three iterations. If the iterative algorithm does not converge within three steps, the truncated estimates are very much determined by the point we start the algorithm. As the sample size n increases, the flip-flop algorithm converges faster, or needs fewer steps to converge. This makes the truncated estimator closer to the truth, and thus reduces the dependence of the truncated estimator on the initial value.

In practice, we recommend randomly generating an initial for the iterative procedure but trying a few different initials for the truncated one. In all the following simulations, we randomly generate 10 initials for the truncated algorithm, and choose the one with the smallest mean squared error to be the truncated flip-flop estimate.

4.5.3 Simulation results

Representative results are reported in Tables 4.1-4.3. The simulation results for the other combinations are quite similar to the ones we report here and thus are omitted.

Table 4.1 summarizes the results under different models with increasing sample size n, while fixing the signal to noise ratio $snr = 1$ and the dimension $q = 20$. Table 4.2 studies how the estimation and prediction accuracy vary with decreasing aspect ratio p/q (increasing q) and fixed $snr = 2$ and $n = 2000$. Table 4.3 reports the effects of snr with fixed $n = 5000$ and $q = 10$.

Under all 4 simulation models, the flip-flop and truncated flip-flop estimators work comparable to each other, and both significantly outperform the standard linear regression estimator for both estimation and prediction. As the sample size n increases (Table 4.1), all methods

	D_{ff}	D_{tf}	D_{lm}	$MSPE_{ff}$	$MSPE_{tf}$	$MSPE_{lm}$
Model I						
n=1000	0.171 (0.022)	0.180 (0.023)	0.497 (0.026)	1.031 (0.046)	1.034 (0.046)	1.258 (0.064)
n=2000	0.119 (0.016)	0.123 (0.018)	0.332 (0.018)	1.012 (0.043)	1.013 (0.044)	1.109 (0.052)
n=5000	0.076 (0.010)	0.076 (0.010)	0.203 (0.010)	1.003 (0.046)	1.003 (0.046)	1.040 (0.049)
n=10000	0.054 (0.007)	0.054 (0.007)	0.143 (0.007)	0.993 (0.041)	0.994 (0.041)	1.010 (0.042)
Model II						
n=1000	0.189 (0.027)	0.202 (0.031)	0.618 (0.039)	0.830 (0.037)	0.833 (0.038)	1.013 (0.052)
n=2000	0.159 (0.024)	0.161 (0.024)	0.478 (0.030)	1.107 (0.047)	1.108 (0.047)	1.213 (0.057)
n=5000	0.085 (0.012)	0.086 (0.012)	0.257 (0.016)	0.839 (0.038)	0.839 (0.038)	0.870 (0.041)
n=10000	0.065 (0.009)	0.065 (0.009)	0.181 (0.011)	0.828 (0.034)	0.828 (0.034)	0.842 (0.035)
Model III						
n=1000	0.315 (0.049)	0.321 (0.050)	1.296 (0.085)	3.657 (0.156)	3.661 (0.158)	4.414 (0.219)
n=2000	0.227 (0.035)	0.228 (0.035)	0.865 (0.056)	3.581 (0.146)	3.582 (0.147)	3.922 (0.185)
n=5000	0.140 (0.022)	0.140 (0.022)	0.530 (0.033)	3.544 (0.179)	3.544 (0.179)	3.669 (0.179)
n=10000	0.095 (0.015)	0.095 (0.015)	0.372 (0.025)	3.542 (0.170)	3.542 (0.170)	3.607 (0.170)
Model IV						
n=1000	0.331 (0.051)	0.337 (0.050)	1.473 (0.097)	4.724 (0.188)	4.727 (0.188)	5.704 (0.284)
n=2000	0.227 (0.035)	0.229 (0.036)	0.983 (0.063)	4.620 (0.196)	4.623 (0.196)	5.068 (0.239)
n=5000	0.145 (0.022)	0.145 (0.021)	0.603 (0.038)	4.582 (0.229)	4.582 (0.229)	4.741 (0.232)
n=10000	0.104 (0.015)	0.104 (0.015)	0.423 (0.028)	4.581 (0.221)	4.581 (0.221)	4.660 (0.219)

Table 4.1: Simulation results with fixed $snr = 1$ and $q = 20$

	D_{ff}	D_{tf}	D_{lm}	$MSPE_{ff}$	$MSPE_{tf}$	$MSPE_{lm}$
Model I						
q=10	0.070 (0.011)	0.070 (0.011)	0.164 (0.012)	0.502 (0.024)	0.502 (0.024)	0.524 (0.025)
q=20	0.084 (0.011)	0.087 (0.012)	0.235 (0.013)	0.506 (0.021)	0.506 (0.021)	0.554 (0.026)
q=40	0.111 (0.013)	0.120 (0.014)	0.353 (0.015)	0.513 (0.021)	0.515 (0.021)	0.625 (0.026)
Model II						
q=10	0.082 (0.014)	0.083 (0.015)	0.221 (0.020)	0.479 (0.023)	0.479 (0.023)	0.500 (0.024)
q=20	0.112 (0.017)	0.114 (0.018)	0.338 (0.021)	0.554 (0.023)	0.554 (0.024)	0.606 (0.028)
q=40	0.159 (0.021)	0.167 (0.022)	0.548 (0.026)	0.634 (0.026)	0.636 (0.026)	0.772 (0.033)
Model III						
q=10	0.108 (0.021)	0.109 (0.022)	0.358 (0.032)	1.304 (0.057)	1.304 (0.057)	1.361 (0.060)
q=20	0.160 (0.024)	0.161 (0.025)	0.612 (0.039)	1.791 (0.073)	1.791 (0.073)	1.961 (0.092)
q=40	0.227 (0.026)	0.232 (0.027)	0.980 (0.038)	2.046 (0.087)	2.048 (0.088)	2.499 (0.113)
Model IV						
q=10	0.120 (0.023)	0.120 (0.024)	0.452 (0.040)	2.076 (0.091)	2.076 (0.091)	2.167 (0.095)
q=20	0.160 (0.025)	0.162 (0.026)	0.695 (0.045)	2.310 (0.098)	2.311 (0.098)	2.534 (0.119)
q=40	0.230 (0.026)	0.237 (0.027)	1.080 (0.042)	2.488 (0.113)	2.492 (0.114)	3.036 (0.138)

Table 4.2: Simulation results with fixed $snr = 2$ and $n = 2000$

improve their estimation and prediction accuracy. From Table 4.2, we can see that increasing q will raise the problem complexity and thus decrease the performance. Table 4.3 shows that all estimators increase accuracy with increasing signal to noise ratio snr.

4.6 Discussion

The rank-1 bilinear combination $\boldsymbol{\alpha}^T \mathbf{X} \boldsymbol{\beta}$ is a simple and direct extension of the traditional linear form. We have proposed two estimators for the bilinear regression model, and demonstrated that they outperform the linear model estimator both theoretically and numerically. We

	D_{ff}	D_{tf}	D_{lm}	$MSPE_{ff}$	$MSPE_{tf}$	$MSPE_{lm}$
Model I						
snr=0.5	0.087 (0.015)	0.088 (0.015)	0.205 (0.016)	1.999 (0.085)	1.999 (0.085)	2.032 (0.086)
snr=1.0	0.062 (0.010)	0.062 (0.010)	0.145 (0.011)	1.000 (0.043)	1.000 (0.043)	1.016 (0.043)
snr=2.0	0.044 (0.007)	0.044 (0.007)	0.102 (0.008)	0.500 (0.021)	0.500 (0.021)	0.508 (0.021)
snr=4.0	0.031 (0.005)	0.031 (0.005)	0.072 (0.006)	0.250 (0.011)	0.250 (0.011)	0.254 (0.011)
Model II						
snr=0.5	0.120 (0.023)	0.120 (0.022)	0.360 (0.035)	3.319 (0.141)	3.320 (0.141)	3.372 (0.143)
snr=1.0	0.085 (0.016)	0.085 (0.016)	0.254 (0.025)	1.660 (0.070)	1.660 (0.070)	1.686 (0.071)
snr=2.0	0.060 (0.011)	0.060 (0.011)	0.180 (0.017)	0.830 (0.035)	0.830 (0.035)	0.843 (0.036)
snr=4.0	0.042 (0.008)	0.042 (0.008)	0.127 (0.012)	0.415 (0.018)	0.415 (0.018)	0.422 (0.018)
Model III						
snr=0.5	0.136 (0.024)	0.136 (0.023)	0.438 (0.039)	5.201 (0.238)	5.201 (0.237)	5.288 (0.240)
snr=1.0	0.096 (0.017)	0.096 (0.017)	0.310 (0.028)	2.600 (0.119)	2.600 (0.119)	2.644 (0.120)
snr=2.0	0.068 (0.012)	0.068 (0.012)	0.219 (0.020)	1.300 (0.059)	1.300 (0.059)	1.322 (0.060)
snr=4.0	0.048 (0.008)	0.048 (0.008)	0.155 (0.014)	0.650 (0.030)	0.650 (0.030)	0.661 (0.030)
Model IV						
snr=0.5	0.152 (0.030)	0.152 (0.030)	0.553 (0.050)	8.277 (0.375)	8.278 (0.375)	8.424 (0.383)
snr=1.0	0.107 (0.021)	0.107 (0.021)	0.391 (0.035)	4.139 (0.187)	4.139 (0.187)	4.212 (0.191)
snr=2.0	0.076 (0.015)	0.076 (0.015)	0.277 (0.025)	2.069 (0.094)	2.069 (0.094)	2.106 (0.096)
snr=4.0	0.053 (0.010)	0.054 (0.010)	0.196 (0.018)	1.035 (0.047)	1.035 (0.047)	1.053 (0.048)

Table 4.3: Simulation results with fixed $n = 5000$ and $q = 10$

also proposed one bilinear ridge estimator to deal with the case that the dimensions are comparable to the sample size. Nevertheless, in some real applications, the underlying data structure is complex, and thus the rank-1 combination may not be flexible enough to capture all the information. In those cases, a multi-rank model is necessary, for example, one can consider an additive bilinear form, $\sum_i \boldsymbol{\alpha}_i^T \mathbf{X} \boldsymbol{\beta}_i$, for the regression mean. The theoretical and numerical performances of the multi-rank bilinear model need further investigation.

4.7 Proofs of theorems

In this section, we first provide proofs of Theorem 3 - Theorem 7 which are under the classical asymptotic setting, i.e., p and q are fixed and $n \to \infty$. We then provide technical details of the Theorem 8 under the setting that n, p and $q \to \infty$. We start with the proof of Theorem 4 since the proof of Theorem 3 will adopt the same idea as the proof of Theorem 4.

Proof of Theorem 4. We first prove the consistency property of the estimator $\hat{\boldsymbol{\theta}}_{\text{tf}}$. Suppose the initialization is given by $\boldsymbol{\beta}_{\text{init}}$. Let $\boldsymbol{\alpha}_{(1)} = \boldsymbol{\alpha}(\boldsymbol{\beta}_{\text{init}})$ denote the estimate of $\boldsymbol{\alpha}_0$ after the first

iteration step. Then, plugging into the generative model, we get

$$\begin{aligned}\boldsymbol{\alpha}_{(1)} &= (\frac{1}{n}\sum_i \mathbf{X}_i\boldsymbol{\beta}_{\text{init}}\boldsymbol{\beta}_{\text{init}}^T\mathbf{X}_i^T)^{-1}(\frac{1}{n}\sum_i \mathbf{X}_i\boldsymbol{\beta}_{\text{init}}\boldsymbol{\beta}_0^T\mathbf{X}_i^T\boldsymbol{\alpha}_0 + \frac{1}{n}\sum_i \mathbf{X}_i\boldsymbol{\beta}_{\text{init}}\epsilon_i) \\ &\stackrel{\text{def}}{=} (\mathbf{A}_1)^{-1}(\mathbf{B}_1\boldsymbol{\alpha}_0 + \mathbf{c}_1),\end{aligned}$$

where we have defined the following quantities, whose limits can be obtained by the law of large numbers,

$$\begin{aligned}\mathbf{A}_1 &= \frac{1}{n}\sum_i \mathbf{X}_i\boldsymbol{\beta}_{\text{init}}\boldsymbol{\beta}_{\text{init}}^T\mathbf{X}_i^T \stackrel{w.p.1}{\longrightarrow} \text{tr}(\boldsymbol{\beta}_{\text{init}}\boldsymbol{\beta}_{\text{init}}^T\boldsymbol{\Psi})\boldsymbol{\Sigma} = (\boldsymbol{\beta}_{\text{init}}^T\boldsymbol{\Psi}\boldsymbol{\beta}_{\text{init}})\boldsymbol{\Sigma}, \\ \mathbf{B}_1 &= \frac{1}{n}\sum_i \mathbf{X}_i\boldsymbol{\beta}_{\text{init}}\boldsymbol{\beta}_0^T\mathbf{X}_i^T \stackrel{w.p.1}{\longrightarrow} \text{tr}(\boldsymbol{\beta}_{\text{init}}\boldsymbol{\beta}_0^T\boldsymbol{\Psi})\boldsymbol{\Sigma} = (\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{\text{init}})\boldsymbol{\Sigma}, \\ \mathbf{c}_1 &= \frac{1}{n}\sum_i \mathbf{X}_i\boldsymbol{\beta}_{\text{init}}\epsilon_i \stackrel{w.p.1}{\longrightarrow} \mathbf{0},\end{aligned}$$

where $w.p.1$ is a shorthand for with probability 1. It follows that

$$\boldsymbol{\alpha}_{(1)} \stackrel{w.p.1}{\longrightarrow} \frac{\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{\text{init}}}{\boldsymbol{\beta}_{\text{init}}^T\boldsymbol{\Psi}\boldsymbol{\beta}_{\text{init}}}\boldsymbol{\alpha}_0 \stackrel{\text{def}}{=} \gamma\boldsymbol{\alpha}_0 \stackrel{\text{def}}{=} \boldsymbol{\alpha}_\star.$$

Here, $\gamma = \frac{\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{\text{init}}}{\boldsymbol{\beta}_{\text{init}}^T\boldsymbol{\Psi}\boldsymbol{\beta}_{\text{init}}}$ is a scaler that depends on the initialization and will show up frequently throughout the proof because $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$ are only identifiable up to a scaler and the flip-flop algorithm eventually converges to $\boldsymbol{\alpha}_\star = \gamma\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_\star = \gamma^{-1}\boldsymbol{\beta}_0$, which makes $\boldsymbol{\beta}_\star \otimes \boldsymbol{\alpha}_\star = \boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0$.

Let $\boldsymbol{\beta}_{(2)} = \boldsymbol{\beta}(\boldsymbol{\alpha}_{(1)}) = \boldsymbol{\beta}(\boldsymbol{\alpha}(\boldsymbol{\beta}_{\text{init}}))$ and $\boldsymbol{\alpha}_{(3)} = \boldsymbol{\alpha}(\boldsymbol{\beta}_{(2)}) = \boldsymbol{\alpha}(\boldsymbol{\beta}(\boldsymbol{\alpha}(\boldsymbol{\beta}_{\text{init}})))$ denote the updates of $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ after two and three steps respectively. Similar to the analysis of $\boldsymbol{\alpha}_{(1)}$ we have

$$\boldsymbol{\beta}_{(2)} \stackrel{w.p.1}{\longrightarrow} \frac{\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\gamma\boldsymbol{\alpha}_0}{\gamma\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\gamma\boldsymbol{\alpha}_0}\boldsymbol{\beta}_0 = \gamma^{-1}\boldsymbol{\beta}_0 \stackrel{\text{def}}{=} \boldsymbol{\beta}_\star,$$

and

$$\boldsymbol{\alpha}_{(3)} \xrightarrow{w.p.1} \frac{\boldsymbol{\beta}_0^T \boldsymbol{\Psi} \gamma^{-1} \boldsymbol{\beta}_0}{\gamma^{-1} \boldsymbol{\beta}_0^T \boldsymbol{\Psi} \gamma^{-1} \boldsymbol{\beta}_0} \boldsymbol{\alpha}_0 = \gamma \boldsymbol{\alpha}_0 = \boldsymbol{\alpha}_\star.$$

Then, it follows that

$$\hat{\boldsymbol{\theta}}_{\mathrm{tf}} = \boldsymbol{\beta}_{(2)} \otimes \boldsymbol{\alpha}_{(3)} \xrightarrow{w.p.1} \boldsymbol{\beta}_\star \otimes \boldsymbol{\alpha}_\star = \boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0.$$

This completes the proof of consistency.

Next, we derive the first-order expansion of $\hat{\boldsymbol{\theta}}_{\mathrm{tf}}$ around $\boldsymbol{\theta}_0$, which is the key to the computation of the variance. We use symbol $\approx$ to represent "equal up to first order". From now on, the notation convention is that the subscript star $*$ means the limit and the tilde˜means the first order term.

As in the consistency part, we start from the analysis of $\boldsymbol{\alpha}_{(1)}$. Let $\mathbf{A}_{1*} = (\boldsymbol{\beta}_{\mathrm{init}}^T \boldsymbol{\Psi} \boldsymbol{\beta}_{\mathrm{init}}) \boldsymbol{\Sigma}$, $\mathbf{B}_{1*} = (\boldsymbol{\beta}_0^T \boldsymbol{\Psi} \boldsymbol{\beta}_{\mathrm{init}}) \boldsymbol{\Sigma}$. By the law of large numbers we know that $\mathbf{A}_1 \xrightarrow{w.p.1} \mathbf{A}_{1*}$, $\mathbf{B}_1 \xrightarrow{w.p.1} \mathbf{B}_{1*}$ and hence we can write $\tilde{\mathbf{A}}_1 = \mathbf{A}_1 - \mathbf{A}_{1*}$ and $\tilde{\mathbf{B}}_1 = \mathbf{B}_1 - \mathbf{B}_{1*}$. Because of the matrix inversion lemma

$$(\mathbf{A} + \tilde{\mathbf{A}})^{-1} \approx \mathbf{A}^{-1} - \mathbf{A}^{-1} \tilde{\mathbf{A}} \mathbf{A}^{-1}, \tag{4.19}$$

when $\mathbf{A}_{1*}$ and $\mathbf{A}_1 = \mathbf{A}_{1*} + \tilde{\mathbf{A}}_1$ are non-singular matrices and $\tilde{\mathbf{A}}_1 = o(1)$, we have

$$\mathbf{A}_1^{-1} \approx \mathbf{A}_{1*}^{-1} - \mathbf{A}_{1*}^{-1} \tilde{\mathbf{A}}_1 \mathbf{A}_{1*}^{-1}.$$

Therefore,

$$\begin{aligned} \boldsymbol{\alpha}_{(1)} &= \mathbf{A}_1^{-1}(\mathbf{B}_1 \boldsymbol{\alpha}_0 + \mathbf{c}_1) \approx (\mathbf{A}_{1*}^{-1} - \mathbf{A}_{1*}^{-1} \tilde{\mathbf{A}}_1 \mathbf{A}_{1*}^{-1})(\mathbf{B}_{1*} \boldsymbol{\alpha}_0 + \tilde{\mathbf{B}}_1 \boldsymbol{\alpha}_0 + \mathbf{c}_1) \\ &\approx \boldsymbol{\alpha}_\star + \mathbf{A}_{1*}^{-1} \tilde{\mathbf{B}}_1 \boldsymbol{\alpha}_0 + \mathbf{A}_{1*}^{-1} \mathbf{c}_1 - \mathbf{A}_{1*}^{-1} \tilde{\mathbf{A}}_1 \boldsymbol{\alpha}_\star \end{aligned} \tag{4.20}$$

We further let $\tilde{\boldsymbol{\alpha}}_{(1)} = \mathbf{A}_{1*}^{-1} \tilde{\mathbf{B}}_1 \boldsymbol{\alpha}_0 + \mathbf{A}_{1*}^{-1} \mathbf{c}_1 - \mathbf{A}_{1*}^{-1} \tilde{\mathbf{A}}_1 \boldsymbol{\alpha}_\star$ to denote the first-order term of $\boldsymbol{\alpha}_{(1)}$

and hence $\boldsymbol{\alpha}_{(1)} \approx \boldsymbol{\alpha}_\star + \tilde{\boldsymbol{\alpha}}_{(1)}$.

For $\boldsymbol{\beta}_{(2)}$, we have

$$\begin{aligned}\boldsymbol{\beta}_{(2)} &= (\frac{1}{n}\sum_i \mathbf{X}_i^T \boldsymbol{\alpha}_{(1)}\boldsymbol{\alpha}_{(1)}^T \mathbf{X}_i)^{-1}(\frac{1}{n}\sum_i \mathbf{X}_i^T \boldsymbol{\alpha}_{(1)}\boldsymbol{\alpha}_0^T \mathbf{X}_i\boldsymbol{\beta}_0 + \frac{1}{n}\sum_i \mathbf{X}_i^T \boldsymbol{\alpha}_{(1)}\epsilon_i) \\ &\stackrel{\text{def}}{=} (\mathbf{A}_2)^{-1}(\mathbf{B}_2\boldsymbol{\beta}_0 + \mathbf{c}_2), \end{aligned} \tag{4.21}$$

where

$$\mathbf{A}_2 = \frac{1}{n}\sum_i \mathbf{X}_i^T \boldsymbol{\alpha}_{(1)}\boldsymbol{\alpha}_{(1)}^T \mathbf{X}_i, \quad \mathbf{B}_2 = \frac{1}{n}\sum_i \mathbf{X}_i^T \boldsymbol{\alpha}_{(1)}\boldsymbol{\alpha}_0^T \mathbf{X}_i,$$

and

$$\mathbf{c}_2 = \frac{1}{n}\sum_i \mathbf{X}_i^T \boldsymbol{\alpha}_{(1)}\epsilon_i \approx \frac{1}{n}\sum_i \mathbf{X}_i^T \boldsymbol{\alpha}_\star \epsilon_i.$$

In order to obtain the first order expansions of matrices $\mathbf{A}_2$ and $\mathbf{B}_2$, we need the following notations and results. Let

$$\mathbf{F} = \frac{1}{n}\sum_i \mathbf{X}_i^T \otimes \mathbf{X}_i^T, \quad \text{and} \quad \mathbf{F}_* = \text{vec}(\boldsymbol{\Psi})\text{vec}^T(\boldsymbol{\Sigma}).$$

By the basic fact that $\text{E}(\mathbf{X}_i^T \otimes \mathbf{X}_i^T) = \text{vec}(\boldsymbol{\Psi})\text{vec}^T(\boldsymbol{\Sigma})$ and the law of large numbers, we have $\mathbf{F} \stackrel{w.p.1}{\longrightarrow} \mathbf{F}_*$ and hence $\tilde{\mathbf{F}} \stackrel{\text{def}}{=} \mathbf{F} - \mathbf{F}_* = o(1)$. Now we vectorize matrices $\mathbf{A}_2$ and $\mathbf{B}_2$ to obtain their first order expansions.

$$\begin{aligned}\text{vec}(\mathbf{A}_2) &= \frac{1}{n}\sum_i (\mathbf{X}_i^T \otimes \mathbf{X}_i^T)\text{vec}(\boldsymbol{\alpha}_{(1)}\boldsymbol{\alpha}_{(1)}^T) \\ &\approx (\mathbf{F}_* + \tilde{\mathbf{F}})\text{vec}\left((\boldsymbol{\alpha}_\star + \tilde{\boldsymbol{\alpha}}_{(1)})(\boldsymbol{\alpha}_\star + \tilde{\boldsymbol{\alpha}}_{(1)})^T\right).\end{aligned}$$

Keeping the first-order terms leads to

$$\text{vec}(\mathbf{A}_2) \approx \mathbf{F}_*\text{vec}(\boldsymbol{\alpha}_\star\boldsymbol{\alpha}_\star^T) + \mathbf{F}_*\text{vec}(\tilde{\boldsymbol{\alpha}}_{(1)}\boldsymbol{\alpha}_\star^T) + \mathbf{F}_*\text{vec}(\boldsymbol{\alpha}_\star\tilde{\boldsymbol{\alpha}}_{(1)}^T) + \tilde{\mathbf{F}}\text{vec}(\boldsymbol{\alpha}_\star\boldsymbol{\alpha}_\star^T).$$

Let $\mathbf{D} = \frac{1}{n}\sum_i \mathbf{X}_i^T \boldsymbol{\alpha}_\star \boldsymbol{\alpha}_\star^T \mathbf{X}_i$ and $\mathbf{D}_* = (\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_\star)\boldsymbol{\Psi}$. Since $\mathrm{E}(\mathbf{X}_i^T \boldsymbol{\alpha}_\star \boldsymbol{\alpha}_\star^T \mathbf{X}_i) = (\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_\star)\boldsymbol{\Psi}$, we have $\mathbf{D} \xrightarrow{w.p.1} \mathbf{D}_*$. Then $\tilde{\mathbf{D}} \stackrel{\text{def}}{=} \mathbf{D} - \mathbf{D}_* = o(1)$. Now unvectorizing $\mathrm{vec}(\mathbf{A}_2)$ gives

$$\mathbf{A}_2 \approx (\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_\star)\boldsymbol{\Psi} + 2(\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \tilde{\boldsymbol{\alpha}}_{(1)})\boldsymbol{\Psi} + \tilde{\mathbf{D}},$$

and its inverse by making use of (4.19)

$$\mathbf{A}_2^{-1} \approx (\boldsymbol{\alpha}_\star^T \Sigma \boldsymbol{\alpha}_\star)^{-1}\boldsymbol{\Psi}^{-1} - (\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_\star)^{-2}\boldsymbol{\Psi}^{-1}\left(2(\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \tilde{\boldsymbol{\alpha}}_{(1)})\boldsymbol{\Psi} + \tilde{\mathbf{D}}\right)\boldsymbol{\Psi}^{-1}. \tag{4.22}$$

Applying similar treatment of $\mathbf{A}_2$ to $\mathbf{B}_2$, we have

$$\mathbf{B}_2 \approx \frac{1}{\gamma}(\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_\star)\boldsymbol{\Psi} + \frac{1}{\gamma}(\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \tilde{\boldsymbol{\alpha}}_{(1)})\boldsymbol{\Psi} + \frac{1}{\gamma}\tilde{\mathbf{D}}. \tag{4.23}$$

By Equation (4.21), (4.22) and (4.23), the first order expansion of $\boldsymbol{\beta}_{(2)}$ is given by

$$\boldsymbol{\beta}_{(2)} \approx \boldsymbol{\beta}_\star - \frac{\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \tilde{\boldsymbol{\alpha}}_{(1)}}{\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_\star}\boldsymbol{\beta}_\star + \frac{\boldsymbol{\Psi}^{-1}(\frac{1}{n}\sum_i \mathbf{X}_i^T \boldsymbol{\alpha}_\star \epsilon_i)}{\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_\star}. \tag{4.24}$$

An identical analysis as $\boldsymbol{\beta}_{(2)}$ for $\boldsymbol{\alpha}_{(3)}$ produces

$$\boldsymbol{\alpha}_{(3)} \approx \boldsymbol{\alpha}_\star - \frac{\boldsymbol{\beta}_\star^T \boldsymbol{\Psi} \tilde{\boldsymbol{\beta}}_{(2)}}{\boldsymbol{\beta}_\star^T \boldsymbol{\Psi} \boldsymbol{\beta}_\star}\boldsymbol{\alpha}_\star + \frac{\boldsymbol{\Sigma}^{-1}(\frac{1}{n}\sum_i \mathbf{X}_i \boldsymbol{\beta}_\star \epsilon_i)}{\boldsymbol{\beta}_\star^T \boldsymbol{\Psi} \boldsymbol{\beta}_\star}. \tag{4.25}$$

The first-order expansion of $\hat{\boldsymbol{\theta}}_{\mathrm{tf}}$ around $\boldsymbol{\theta}_0$ is

$$\hat{\boldsymbol{\theta}}_{\mathrm{tf}} - \boldsymbol{\theta}_0 = \boldsymbol{\beta}_{(2)} \otimes \boldsymbol{\alpha}_{(3)} - \boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0 \approx (\boldsymbol{\beta}_{(2)} - \boldsymbol{\beta}_\star) \otimes \boldsymbol{\alpha}_{(3)} + \boldsymbol{\beta}_{(2)} \otimes (\boldsymbol{\alpha}_{(3)} - \boldsymbol{\alpha}_\star). \tag{4.26}$$

Plugging Equations (4.20), (4.24) and (4.25) into Equation (4.26), we have the following first-

order approximation for $\hat{\boldsymbol{\theta}}_{\rm tf}$,

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\rm tf} &\approx \boldsymbol{\theta}_0 + \frac{\boldsymbol{\Psi}^{-1}(\frac{1}{n}\sum_i \mathbf{X}_i^T\boldsymbol{\alpha}_0\epsilon_i)}{\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0}\otimes\boldsymbol{\alpha}_0 + \boldsymbol{\beta}_0\otimes\frac{\boldsymbol{\Sigma}^{-1}(\frac{1}{n}\sum_i \mathbf{X}_i\boldsymbol{\beta}_0\epsilon_i)}{\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0} \\ &\quad -\frac{\boldsymbol{\beta}_0^T(\frac{1}{n}\sum_i \mathbf{X}_i^T\boldsymbol{\alpha}_0\epsilon_i)}{(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)}(\boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0). \qquad (4.27)\end{aligned}$$

One feature of the expansion is that it does not involve the initialization any more because of some cancellations.

Finally, we analyze the asymptotic covariance matrix of $\hat{\boldsymbol{\theta}}_{\rm tf}$. Write the last three terms in Equation (4.27) as follows

$$\begin{aligned}\mathbf{s}_1 &= (\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}(\boldsymbol{\Psi}^{-1}\frac{1}{n}\sum_i \mathbf{X}_i^T\boldsymbol{\alpha}_0\epsilon_i)\otimes\boldsymbol{\alpha}_0, \\ \mathbf{s}_2 &= (\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}\boldsymbol{\beta}_0\otimes(\boldsymbol{\Sigma}^{-1}\frac{1}{n}\sum_i \mathbf{X}_i\boldsymbol{\beta}_0\epsilon_i), \\ \mathbf{s}_3 &= (\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}(\frac{1}{n}\sum_i \boldsymbol{\beta}_0^T\mathbf{X}_i^T\boldsymbol{\alpha}_0\epsilon_i)\boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0.\end{aligned}$$

Note that

$$\begin{aligned}{\rm acov}(\hat{\boldsymbol{\theta}}_{\rm tf}) &\approx {\rm E}(\hat{\boldsymbol{\theta}}_{\rm tf}-\boldsymbol{\theta}_0)(\hat{\boldsymbol{\theta}}_{\rm tf}-\boldsymbol{\theta}_0)^T = {\rm E}(\mathbf{s}_1+\mathbf{s}_2-\mathbf{s}_3)(\mathbf{s}_1+\mathbf{s}_2-\mathbf{s}_3)^T \\ &\approx {\rm E}(\mathbf{s}_1\mathbf{s}_1^T+\mathbf{s}_2\mathbf{s}_2^T+\mathbf{s}_3\mathbf{s}_3^T+\mathbf{s}_1\mathbf{s}_2^T-\mathbf{s}_1\mathbf{s}_3^T-\mathbf{s}_2\mathbf{s}_3^T+\mathbf{s}_2\mathbf{s}_1^T-\mathbf{s}_3\mathbf{s}_1^T-\mathbf{s}_3\mathbf{s}_2^T).\end{aligned}$$

Below we will calculate the expectation of each term on the right hand side.

The first term is

$$\mathbf{s}_1\mathbf{s}_1^T = (\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-2}\left((\frac{1}{n}\sum_i \boldsymbol{\Psi}^{-1}\mathbf{X}_i^T\boldsymbol{\alpha}_0\epsilon_i)(\frac{1}{n}\sum_j \epsilon_j\boldsymbol{\alpha}_0^T\mathbf{X}_j\boldsymbol{\Psi}^{-1})\right)\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T).$$

By the independence of ϵ_i and $\mathbf{X}_i$, we have

$$\mathrm{E}(\epsilon_i^2 \mathbf{X}_i^T \boldsymbol{\alpha}_0 \boldsymbol{\alpha}_0^T \mathbf{X}_i) = \tau^2 (\boldsymbol{\alpha}_0^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_0) \boldsymbol{\Psi}. \tag{4.28}$$

Again by the independence of $\mathbf{X}_i$ and ϵ_i and Equation (4.28), the expectation is given by

$$\mathrm{E}(\mathbf{s}_1 \mathbf{s}_1^T) = (\boldsymbol{\alpha}_0^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_0)^{-1} \frac{\tau^2}{n} \boldsymbol{\Psi}^{-1} \otimes (\boldsymbol{\alpha}_0 \boldsymbol{\alpha}_0^T).$$

Similar arguments as the $\mathbf{s}_1 \mathbf{s}_1^T$ term together with the fact

$$\mathrm{E}(\epsilon_i^2 \mathbf{X}_i \boldsymbol{\beta}_0 \boldsymbol{\beta}_0^T \mathbf{X}_i^T) = \tau^2 (\boldsymbol{\beta}_0^T \boldsymbol{\Psi} \boldsymbol{\beta}_0) \boldsymbol{\Sigma},$$

produce

$$\mathrm{E}(\mathbf{s}_2 \mathbf{s}_2^T) = (\boldsymbol{\beta}_0^T \boldsymbol{\Psi} \boldsymbol{\beta}_0)^{-1} \frac{\tau^2}{n} (\boldsymbol{\beta}_0 \boldsymbol{\beta}_0^T) \otimes \boldsymbol{\Sigma}^{-1}.$$

For the $\mathbf{s}_3 \mathbf{s}_3^T$ term, we have

$$\mathbf{s}_3 \mathbf{s}_3^T = (\boldsymbol{\alpha}_0^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_0)^{-2} (\boldsymbol{\beta}_0^T \boldsymbol{\Psi} \boldsymbol{\beta}_0)^{-2} (\frac{1}{n} \sum_i \boldsymbol{\beta}_0^T \mathbf{X}_i^T \boldsymbol{\alpha}_0 \epsilon_i)^2 (\boldsymbol{\beta}_0 \boldsymbol{\beta}_0^T) \otimes \boldsymbol{\alpha}_0 \boldsymbol{\alpha}_0^T.$$

By equation (4.28), it follows

$$\mathrm{E}(\frac{1}{n} \sum_i \boldsymbol{\beta}_0^T \mathbf{X}_i^T \boldsymbol{\alpha}_0 \epsilon_i)^2 = \frac{\tau^2}{n} (\boldsymbol{\alpha}_0^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_0) (\boldsymbol{\beta}_0^T \boldsymbol{\Psi} \boldsymbol{\beta}_0),$$

and hence

$$\mathrm{E}(\mathbf{s}_3 \mathbf{s}_3^T) = (\boldsymbol{\alpha}_0^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_0)^{-1} (\boldsymbol{\beta}_0^T \boldsymbol{\Psi} \boldsymbol{\beta}_0)^{-1} \frac{\tau^2}{n} (\boldsymbol{\beta}_0 \boldsymbol{\beta}_0^T) \otimes (\boldsymbol{\alpha}_0 \boldsymbol{\alpha}_0^T).$$

For the $\mathbf{s}_1 \mathbf{s}_2^T$ term, we have

$$\mathbf{s}_1 \mathbf{s}_2^T = (\boldsymbol{\alpha}_0^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_0)^{-1} (\boldsymbol{\beta}_0^T \boldsymbol{\Psi} \boldsymbol{\beta}_0)^{-1} \left((\frac{1}{n} \sum_i \boldsymbol{\Psi}^{-1} \mathbf{X}_i^T \boldsymbol{\alpha}_0 \epsilon_i \boldsymbol{\beta}_0^T) \otimes (\frac{1}{n} \sum_j \epsilon_j \boldsymbol{\alpha}_0 \boldsymbol{\beta}_0^T \mathbf{X}_j^T \boldsymbol{\Sigma}^{-1}) \right).$$

Note that

$$(\boldsymbol{\Psi}^{-1}\mathbf{X}_i^T\boldsymbol{\alpha}_0\boldsymbol{\beta}_0^T)\otimes(\boldsymbol{\alpha}_0\boldsymbol{\beta}_0^T\mathbf{X}_i^T\boldsymbol{\Sigma}^{-1}) = (\boldsymbol{\Psi}\otimes\boldsymbol{\alpha}_0\boldsymbol{\beta}_0^T)(\mathbf{X}_i^T\otimes\mathbf{X}_i^T)(\boldsymbol{\alpha}_0\boldsymbol{\beta}_0^T\otimes\boldsymbol{\Sigma}^{-1}),$$

then it follows

$$\begin{aligned}
& \mathrm{E}(\boldsymbol{\Psi}^{-1}\mathbf{X}_i^T\boldsymbol{\alpha}_0\boldsymbol{\beta}_0^T)\otimes(\boldsymbol{\alpha}_0\boldsymbol{\beta}_0^T\mathbf{X}_i^T\boldsymbol{\Sigma}^{-1}) \\
= \;& (\boldsymbol{\Psi}^{-1}\otimes\boldsymbol{\alpha}_0\boldsymbol{\beta}_0^T)(\mathrm{vec}(\boldsymbol{\Psi})\mathrm{vec}^T(\boldsymbol{\Sigma}))(\boldsymbol{\alpha}_0\boldsymbol{\beta}_0^T\otimes\boldsymbol{\Sigma}^{-1}) \\
= \;& (\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T)\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T),
\end{aligned}$$

and hence

$$\mathrm{E}(\mathbf{s}_1\mathbf{s}_2^T) = (\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}\frac{\tau^2}{n}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T)\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T).$$

For the $\mathbf{s}_1\mathbf{s}_3^T$ term, we have

$$\begin{aligned}
& \mathbf{s}_1\mathbf{s}_3^T \\
= \;& (\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-2}(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}(\frac{1}{n}\sum_i\boldsymbol{\beta}_0^T\mathbf{X}_i^T\boldsymbol{\alpha}_0\epsilon_i)(\frac{1}{n}\sum_j\epsilon_j\boldsymbol{\Psi}^{-1}\mathbf{X}_j^T\boldsymbol{\alpha}_0\boldsymbol{\beta}_0^T)\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T) \\
= \;& (\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-2}(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}\left(\frac{1}{n}\sum_j\epsilon_j\boldsymbol{\Psi}^{-1}\mathbf{X}_j^T\boldsymbol{\alpha}_0(\frac{1}{n}\sum_i\boldsymbol{\alpha}_0^T\mathbf{X}_i\epsilon_i)\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T\right)\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T).
\end{aligned}$$

By the independence of $\mathbf{X}_i$ and ϵ_j, it follows

$$\mathrm{E}(\frac{1}{n}\sum_j\epsilon_j\boldsymbol{\Psi}^{-1}\mathbf{X}_j^T\boldsymbol{\alpha}_0(\frac{1}{n}\sum_i\boldsymbol{\alpha}_0^T\mathbf{X}_i\epsilon_i)) = \frac{\tau^2}{n}(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)I,$$

and hence

$$\mathrm{E}(\mathbf{s}_1\mathbf{s}_3^T) = (\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}\frac{\tau^2}{n}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T)\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T).$$

By the same treatment of the $\mathrm{s}_1\mathrm{s}_3^T$ term, the expectation of the $\mathrm{s}_2\mathrm{s}_3^T$ term is given by

$$\mathrm{E}(\mathbf{s}_2\mathbf{s}_3^T) = (\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}\frac{\tau^2}{n}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T)\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T).$$

By symmetry, we have $\mathrm{E}(\mathbf{s}_1\mathbf{s}_2^T) = \mathrm{E}(\mathbf{s}_2\mathbf{s}_1^T)$, $\mathrm{E}(\mathbf{s}_1\mathbf{s}_3^T) = \mathrm{E}(\mathbf{s}_3\mathbf{s}_1^T)$ and $\mathrm{E}(\mathbf{s}_2\mathbf{s}_3^T) = \mathrm{E}(\mathbf{s}_3\mathbf{s}_2^T)$. The asymptotic covariance matrix of $\hat{\boldsymbol{\theta}}_{\mathrm{tf}}$ then is given as follows,

$$\begin{aligned}
\mathrm{acov}(\hat{\boldsymbol{\theta}}_{\mathrm{tf}}) &\approx \mathrm{E}(\mathbf{s}_1\mathbf{s}_1^T + \mathbf{s}_2\mathbf{s}_2^T + \mathbf{s}_3\mathbf{s}_3^T + \mathbf{s}_1\mathbf{s}_2^T - \mathbf{s}_1\mathbf{s}_3^T - \mathbf{s}_2\mathbf{s}_3^T + \mathbf{s}_2\mathbf{s}_1^T - \mathbf{s}_3\mathbf{s}_1^T - \mathbf{s}_3\mathbf{s}_2^T) \\
&= (\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}\frac{\tau^2}{n}\boldsymbol{\Psi}^{-1}\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T) + (\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}\frac{\tau^2}{n}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T)\otimes\boldsymbol{\Sigma}^{-1} \\
&\quad -(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}\frac{\tau^2}{n}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T)\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T),
\end{aligned}$$

which completes the proof of Theorem 4.

Proof of Theorem 3 From the proof of Theorem 4, we know that

$$\hat{\boldsymbol{\alpha}}_{\mathrm{ff}} \xrightarrow{w.p.1} \boldsymbol{\alpha}_\star, \quad \hat{\boldsymbol{\beta}}_{\mathrm{ff}} \xrightarrow{w.p.1} \boldsymbol{\beta}_\star.$$

Hence,

$$\hat{\boldsymbol{\theta}}_{\mathrm{ff}} = \hat{\boldsymbol{\beta}}_{\mathrm{ff}}\otimes\hat{\boldsymbol{\alpha}}_{\mathrm{ff}} \xrightarrow{w.p.1} \boldsymbol{\beta}_\star\otimes\boldsymbol{\alpha}_\star = \boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0 = \boldsymbol{\theta}_0,$$

which means that $\hat{\boldsymbol{\theta}}_{\mathrm{ff}}$ is a consistent estimator of $\boldsymbol{\theta}_0$. Then

$$\tilde{\boldsymbol{\alpha}}_{\mathrm{ff}} \stackrel{\mathrm{def}}{=} \hat{\boldsymbol{\alpha}}_{\mathrm{ff}} - \boldsymbol{\alpha}_\star = o(1), \quad \text{and} \quad \tilde{\boldsymbol{\beta}}_{\mathrm{ff}} \stackrel{\mathrm{def}}{=} \hat{\boldsymbol{\beta}}_{\mathrm{ff}} - \boldsymbol{\beta}_\star = o(1).$$

Note that when the algorithm converges, we have the following

$$\hat{\boldsymbol{\alpha}}_{\mathrm{ff}} = \boldsymbol{\alpha}(\hat{\boldsymbol{\beta}}_{\mathrm{ff}}), \quad \text{and} \quad \hat{\boldsymbol{\beta}}_{\mathrm{ff}} = \boldsymbol{\alpha}(\hat{\boldsymbol{\beta}}_{\mathrm{ff}}).$$

By the exactly same treatment of $\boldsymbol{\beta}_{(2)}$ in the proof of Theorem 4 and replacing $\boldsymbol{\alpha}_{(1)}$ by $\hat{\boldsymbol{\alpha}}_{\mathrm{ff}}$, we

have

$$\hat{\boldsymbol{\beta}}_{\text{ff}} \approx \boldsymbol{\beta}_\star - \frac{\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \tilde{\boldsymbol{\alpha}}_{\text{ff}}}{\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_\star} \boldsymbol{\beta}_\star + \frac{\boldsymbol{\Psi}^{-1}(\frac{1}{n}\sum_i \mathbf{X}_i^T \boldsymbol{\alpha}_\star \epsilon_i)}{\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_\star}, \tag{4.29}$$

and similarly,

$$\hat{\boldsymbol{\alpha}}_{\text{ff}} \approx \boldsymbol{\alpha}_\star - \frac{\boldsymbol{\beta}_\star^T \boldsymbol{\Psi} \tilde{\boldsymbol{\beta}}_{\text{ff}}}{\boldsymbol{\beta}_\star^T \boldsymbol{\Psi} \boldsymbol{\beta}_\star} \boldsymbol{\alpha}_\star + \frac{\boldsymbol{\Sigma}^{-1}(\frac{1}{n}\sum_i \mathbf{X}_i \boldsymbol{\beta}_\star \epsilon_i)}{\boldsymbol{\beta}_\star^T \boldsymbol{\Psi} \boldsymbol{\beta}_\star}.$$

Then the first order approximation of $\hat{\boldsymbol{\theta}}_{\text{ff}}$ is given by

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{\text{ff}} - \boldsymbol{\theta}_0 &= \hat{\boldsymbol{\beta}}_{\text{ff}} \otimes \hat{\boldsymbol{\alpha}}_{\text{ff}} - \boldsymbol{\theta}_0 \\
&\approx \frac{\boldsymbol{\Sigma}^{-1}(\frac{1}{n}\sum_i \mathbf{X}_i \boldsymbol{\beta}_\star \epsilon_i)}{\boldsymbol{\beta}_\star^T \boldsymbol{\Psi} \boldsymbol{\beta}_\star} \otimes \boldsymbol{\alpha}_\star + \boldsymbol{\beta}_\star \otimes \frac{\boldsymbol{\Sigma}^{-1}(\frac{1}{n}\sum_i \mathbf{X}_i \boldsymbol{\beta}_\star \epsilon_i)}{\boldsymbol{\beta}_\star^T \boldsymbol{\Psi} \boldsymbol{\beta}_\star} \\
&\quad -(\frac{\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \tilde{\boldsymbol{\alpha}}_{\text{ff}}}{\boldsymbol{\alpha}_\star^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_\star} + \frac{\boldsymbol{\beta}_\star^T \boldsymbol{\Psi} \tilde{\boldsymbol{\beta}}_{\text{ff}}}{\boldsymbol{\beta}_\star^T \boldsymbol{\Psi} \boldsymbol{\beta}_\star})(\boldsymbol{\beta}_\star \otimes \boldsymbol{\alpha}_\star).
\end{aligned} \tag{4.30}$$

Plugging Equation (4.29) into Equation (4.30), we have

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{\text{ff}} &\approx \boldsymbol{\theta}_0 + \frac{\boldsymbol{\Psi}^{-1}(\frac{1}{n}\sum_i \mathbf{X}_i^T \boldsymbol{\alpha}_0 \epsilon_i)}{\boldsymbol{\alpha}_0^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_0} \otimes \boldsymbol{\alpha}_0 + \boldsymbol{\beta}_0 \otimes \frac{\boldsymbol{\Sigma}^{-1}(\frac{1}{n}\sum_i \mathbf{X}_i \boldsymbol{\beta}_0 \epsilon_i)}{\boldsymbol{\beta}_0^T \boldsymbol{\Psi} \boldsymbol{\beta}_0} \\
&\quad -\frac{\boldsymbol{\beta}_0^T(\frac{1}{n}\sum_i \mathbf{X}_i^T \boldsymbol{\alpha}_0 \epsilon_i)}{(\boldsymbol{\alpha}_0^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_0)(\boldsymbol{\beta}_0^T \boldsymbol{\Psi} \boldsymbol{\beta}_0)}(\boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0).
\end{aligned} \tag{4.31}$$

The right hand side of Equation (4.31) is the same as the one in Equation (4.27). And hence the asymptotic covariance matrix of $\hat{\boldsymbol{\theta}}_{\text{ff}}$ is the same as $\hat{\boldsymbol{\theta}}_{\text{tf}}$. This completes the proof of Theorem 3.

Proof of Theorem 5 The Fisher information matrix $\mathbf{J}(\boldsymbol{\mu})$ is given by

$$\mathbf{J}(\boldsymbol{\mu}) = -\mathrm{E}_{\boldsymbol{\mu}_0} \frac{\partial^2 l}{\partial \boldsymbol{\mu} \partial \boldsymbol{\mu}^T} = -\mathrm{E}_{\boldsymbol{\mu}_0} \begin{pmatrix} \frac{\partial^2 l}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\alpha}^T} & \frac{\partial^2 l}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\beta}^T} \\ \frac{\partial^2 l}{\partial \boldsymbol{\beta} \partial \boldsymbol{\alpha}^T} & \frac{\partial^2 l}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} \end{pmatrix}.$$

Calculating the derivatives and taking the expectation, we have

$$\mathbf{J}(\boldsymbol{\mu}) = \frac{n}{\tau^2} \begin{pmatrix} (\boldsymbol{\beta}_0^T \boldsymbol{\Psi} \boldsymbol{\beta}_0) \boldsymbol{\Sigma} & \boldsymbol{\Sigma} \boldsymbol{\alpha}_0 \boldsymbol{\beta}_0^T \boldsymbol{\Psi} \\ \boldsymbol{\Psi} \boldsymbol{\beta}_0 \boldsymbol{\alpha}_0^T \boldsymbol{\Sigma} & (\boldsymbol{\alpha}_0 \boldsymbol{\Sigma} \boldsymbol{\alpha}_0^T) \boldsymbol{\Psi} \end{pmatrix}.$$

The CRLB for any unbiased estimator $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}_0$ is given by (Stoica and Marzetta, 2001a)

$$\mathrm{acov}(\hat{\boldsymbol{\theta}}) \succeq \mathbf{A}\mathbf{J}(\boldsymbol{\mu})^{\dagger}\mathbf{A}^T.$$

The matrix $\mathbf{J}(\boldsymbol{\mu})^{\dagger}$ can be calculated from the formula for partitioned matrices (e.g., Rohde, 1965). Together with Equation (4.12), we have

$$\begin{aligned}\mathrm{acov}(\hat{\boldsymbol{\theta}}) \succeq\ & (\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}\frac{\tau^2}{n}\boldsymbol{\Psi}^{-1}\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T) + (\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}\frac{\tau^2}{n}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T)\otimes\boldsymbol{\Sigma}^{-1} \\ & -(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)^{-1}(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)^{-1}\frac{\tau^2}{n}(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T)\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T).\end{aligned}$$

Proof of Theorem 6 By plugging in the expressions for $\mathbf{A}$ and $\mathbf{J}(\boldsymbol{\theta})$, we have

$$\mathbf{A}^T\mathbf{J}(\boldsymbol{\theta})\mathbf{A} = \frac{n}{\tau^2}\begin{pmatrix} (\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)\boldsymbol{\Sigma} & \boldsymbol{\Sigma}\boldsymbol{\alpha}_0\boldsymbol{\beta}_0^T\boldsymbol{\Psi} \\ \boldsymbol{\Psi}\boldsymbol{\beta}_0\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma} & (\boldsymbol{\alpha}_0\boldsymbol{\Sigma}\boldsymbol{\alpha}_0^T)\boldsymbol{\Psi}\end{pmatrix}.$$

The above expression is the same as the $\mathbf{J}(\boldsymbol{\mu})$ in Theorem 5. Hence, the rest of the proof of Theorem 6 is identical to the proof of Theorem 5.

Proof of Theorem 7 From general linear regression analysis, it is well known that the asymptotic covariance of the linear estimator $\hat{\boldsymbol{\theta}}_{\mathrm{lm}}$ is given by

$$\mathrm{acov}(\hat{\boldsymbol{\theta}}_{\mathrm{lm}}) = \frac{\tau^2}{n}\boldsymbol{\Psi}^{-1}\otimes\boldsymbol{\Sigma}^{-1}.$$

To show the asymptotic efficiency of $\hat{\boldsymbol{\theta}}_{\mathrm{ff}}$ and $\hat{\boldsymbol{\theta}}_{\mathrm{tf}}$, we only need to show $\mathrm{acov}(\hat{\boldsymbol{\theta}}_{\mathrm{lm}})-\mathrm{acov}(\hat{\boldsymbol{\theta}}_{\mathrm{ff}}) \succeq \mathbf{0}$ and $\mathrm{acov}(\hat{\boldsymbol{\theta}}_{\mathrm{lm}}) - \mathrm{acov}(\hat{\boldsymbol{\theta}}_{\mathrm{tf}}) \succeq \mathbf{0}$. This is equivalent to show that

$$\boldsymbol{\Psi}^{-1}\otimes\boldsymbol{\Sigma}^{-1} + \frac{(\boldsymbol{\beta}_0^T\boldsymbol{\beta}_0)\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T)}{(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)} - \frac{(\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T)\otimes\boldsymbol{\Sigma}^{-1}}{\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0} - \frac{\boldsymbol{\Psi}^{-1}\otimes(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T)}{\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0} \succeq \mathbf{0}.$$

The left hand side equals

$$\left(\boldsymbol{\Psi}^{-1} - \frac{\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T}{\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0}\right) \otimes \left(\boldsymbol{\Sigma}^{-1} - \frac{\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T}{\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0}\right).$$

From the basic fact that the Kronecker product of two positive semi-definite matrices is still positive semi-definite, we only need to show that

$$\boldsymbol{\Psi}^{-1} - \frac{\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T}{\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0} \succeq \mathbf{0}, \quad \text{and} \quad \boldsymbol{\Sigma}^{-1} - \frac{\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T}{\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0} \succeq \mathbf{0}.$$

For every $\boldsymbol{\eta} \in \mathbb{R}^q$, by Cauchy-Schwarz inequality we have

$$(\boldsymbol{\eta}^T\boldsymbol{\beta}_0)^2 \leq (\boldsymbol{\eta}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\eta})(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0),$$

and hence it follows that

$$\boldsymbol{\eta}^T\left(\boldsymbol{\Psi}^{-1} - \frac{\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T}{\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0}\right)\boldsymbol{\eta} \geq 0,$$

which means that the matrix $\boldsymbol{\Psi}^{-1} - \frac{\boldsymbol{\beta}_0\boldsymbol{\beta}_0^T}{\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0}$ is positive semi-definite.

By similar arguments we can show that the matrix $\boldsymbol{\Sigma}^{-1} - \frac{\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T}{\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_0}$ is also positive semi-definite. This completes the proof of Theorem 7.

Proof of Theorem 8 The proof of this theorem adopts a similar idea as the proof of Theorem 3. Hence, we use a similar set of notations as used in the proof of Theorem 3 but with a subscript r to represent the bilinear ridge estimator.

Let $\boldsymbol{\alpha}_{r(1)} = \boldsymbol{\alpha}_\lambda(\boldsymbol{\beta}_{\text{init}})$, $\boldsymbol{\beta}_{r(2)} = \boldsymbol{\beta}_\lambda(\boldsymbol{\alpha}_{r(1)})$ and $\boldsymbol{\alpha}_{r(3)} = \boldsymbol{\alpha}_\lambda(\boldsymbol{\beta}_{r(2)})$. As discovered in the Theorem 4, the asymptotic result of the truncated estimator does not depend on the initial $\boldsymbol{\beta}_{\text{init}}$. We will first show that the upper bound of the truncated flip-flop bilinear ridge estimator also does not depend on the initial which is a similar result as in Theorem 4.

Let us first define the following quantities which characterize the limiting behavior of $\boldsymbol{\alpha}_{r(3)}$

and $\boldsymbol{\beta}_{r(2)}$.

$$\gamma_r = \left(\|\boldsymbol{\beta}_{\text{init}}\|_{\boldsymbol{\Psi}}^2 + \lambda_\beta \|\boldsymbol{\beta}_{\text{init}}\|_2^2\right)^{-1} (\boldsymbol{\beta}_0^T \boldsymbol{\Psi} \boldsymbol{\beta}_{\text{init}}), \quad \boldsymbol{\alpha}_{r\star} = \gamma_r \bar{\boldsymbol{\alpha}}, \quad \text{and} \quad \boldsymbol{\beta}_{r\star} = \gamma_r^{-1} \bar{\boldsymbol{\beta}}.$$

Next, we introduce the following four quantities which will be used in the analysis. Let

$$\begin{aligned}
\mathbf{D}_r &= n^{-1} \sum_{i=1}^n \mathbf{X}_i^T \boldsymbol{\alpha}_{r\star} \boldsymbol{\alpha}_{r\star}^T \mathbf{X}_i, \\
\mathbf{E}_r &= n^{-1} \sum_{i=1}^n \mathbf{X}_i^T \boldsymbol{\alpha}_{r\star} \boldsymbol{\alpha}_0^T \mathbf{X}_i, \\
\mathbf{F}_r &= n^{-1} \sum_{i=1}^n \mathbf{X}_i \boldsymbol{\beta}_{r\star} \boldsymbol{\beta}_{r\star}^T \mathbf{X}_i^T, \\
\mathbf{G}_r &= n^{-1} \sum_{i=1}^n \mathbf{X}_i \boldsymbol{\beta}_{r\star} \boldsymbol{\beta}_0^T \mathbf{X}_i^T.
\end{aligned}$$

Their expectations are

$$\begin{aligned}
\mathbf{D}_{r\star} = \mathrm{E}(\mathbf{D}_r) = \|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}}^2 \boldsymbol{\Psi}, \quad \mathbf{E}_{r\star} = \mathrm{E}(\mathbf{E}_r) = (\boldsymbol{\alpha}_{r\star}^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_0) \boldsymbol{\Psi}, \\
\mathbf{F}_{r\star} = \mathrm{E}(\mathbf{F}_r) = \|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2 \boldsymbol{\Sigma}, \quad \mathbf{G}_{r\star} = \mathrm{E}(\mathbf{G}_r) = (\boldsymbol{\beta}_{r\star}^T \boldsymbol{\Psi} \boldsymbol{\beta}_0) \boldsymbol{\Sigma}.
\end{aligned}$$

We can further write $\tilde{\mathbf{D}}_r = \mathbf{D}_r - \mathbf{D}_{r\star}$, $\tilde{\mathbf{E}}_r = \mathbf{E}_r - \mathbf{E}_{r\star}$, $\tilde{\mathbf{F}}_r = \mathbf{F}_r - \mathbf{F}_{r\star}$, and $\tilde{\mathbf{G}}_r = \mathbf{G}_r - \mathbf{G}_{r\star}$, which are the leading error terms.

The excess prediction error for the estimator $(\boldsymbol{\alpha}_{r(3)}, \boldsymbol{\beta}_{r(2)})$ is

$$\begin{aligned}
& \mathrm{E}\left((\tilde{y} - \boldsymbol{\alpha}_{r(3)}^T \tilde{\mathbf{X}} \boldsymbol{\beta}_{r(2)})^2 - (\tilde{y} - \boldsymbol{\alpha}_0^T \tilde{\mathbf{X}} \boldsymbol{\beta}_0)^2\right) \\
= \; & \mathrm{E}\left((\boldsymbol{\beta}_{r(2)}^T \otimes \boldsymbol{\alpha}_{r(3)}^T - \boldsymbol{\beta}_0^T \otimes \boldsymbol{\alpha}_0^T) \mathrm{vec}(\tilde{\mathbf{X}})\right)^2 \\
= \; & \|\boldsymbol{\beta}_{r(2)} \otimes \boldsymbol{\alpha}_{r(3)} - \boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi} \otimes \boldsymbol{\Sigma}}^2 \\
= \; & \|\boldsymbol{\beta}_{r(2)} \otimes \boldsymbol{\alpha}_{r(3)} - \boldsymbol{\beta}_{r\star} \otimes \boldsymbol{\alpha}_{r\star} + \boldsymbol{\beta}_{r\star} \otimes \boldsymbol{\alpha}_{r\star} - \boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi} \otimes \boldsymbol{\Sigma}}^2 \\
\leq \; & 2\|\boldsymbol{\beta}_{r(2)} \otimes \boldsymbol{\alpha}_{r(3)} - \boldsymbol{\beta}_{r\star} \otimes \boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Psi} \otimes \boldsymbol{\Sigma}}^2 + 2\|\bar{\boldsymbol{\beta}} \otimes \bar{\boldsymbol{\alpha}} - \boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi} \otimes \boldsymbol{\Sigma}}^2. \qquad (4.32)
\end{aligned}$$

The second term on the right hand side is deterministic. The upper bound of the expectation of the first term is given in the following lemma.

Lemma 1 *Suppose λ_α and λ_β satisfy*

$$n^{-1}d_{\boldsymbol{\alpha}} = o(1), \quad \textit{and} \quad n^{-1}d_{\boldsymbol{\beta}} = o(1).$$

Then

$$\begin{aligned} & \mathrm{E}\|\boldsymbol{\beta}_{r(2)} \otimes \boldsymbol{\alpha}_{r(3)} - \boldsymbol{\beta}_{r\star} \otimes \boldsymbol{\alpha}_{r\star}\|^2_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}} \\ \leq \; & O\left(\frac{d_{\boldsymbol{\alpha}} + d_{\boldsymbol{\beta}}}{n}(\tau^2 + \|\bar{\boldsymbol{\beta}} \otimes \bar{\boldsymbol{\alpha}} - \boldsymbol{\beta}_0 \otimes \boldsymbol{\alpha}_0\|^2_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}})\right). \end{aligned}$$

Lemma 1 together with Equation (4.32) give us the upper bound on the expectation of the excess prediction error for the estimator $(\boldsymbol{\alpha}_{r(3)}, \boldsymbol{\beta}_{r(2)})$ which is the right hand side of (4.18). Similar to the flip-flop and the trucated flip-flop plain bilinear estimators, the flip-flop bilinear ridge estimator has the same upper bound as its truncated version. This completes the proof of Theorem 8.

4.8 Proofs of lemmas

Proof of Lemma 1 We first derive the first order approximations of $\boldsymbol{\alpha}_{r(1)}$, $\boldsymbol{\beta}_{r(2)}$ and $\boldsymbol{\alpha}_{r(3)}$. Given an initial $\boldsymbol{\beta}_{\text{init}}$, let $\boldsymbol{\alpha}_{r(1)} = \boldsymbol{\alpha}_\lambda(\boldsymbol{\beta}_{\text{init}})$ be the updated value of $\boldsymbol{\alpha}$ after the first iteration step. By the definition of the operator $\boldsymbol{\alpha}_\lambda$, we have

$$\boldsymbol{\alpha}_{r(1)} = \mathbf{A}_{r1}^{-1}(\mathbf{B}_{r1}\boldsymbol{\alpha}_0 + \mathbf{c}_{r1}),$$

where

$$\begin{aligned}
\mathbf{A}_{r1} &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i\boldsymbol{\beta}_{\text{init}}\boldsymbol{\beta}_{\text{init}}^T\mathbf{X}_i^T + \lambda_\alpha\|\boldsymbol{\beta}_{\text{init}}\|_{\boldsymbol{\Psi}}^2\mathbf{I} + \lambda_\beta\|\boldsymbol{\beta}_{\text{init}}\|_2^2\boldsymbol{\Sigma} + \lambda_\alpha\lambda_\beta\|\boldsymbol{\beta}_{\text{init}}\|_2^2\mathbf{I}, \\
\mathbf{B}_{r1} &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i\boldsymbol{\beta}_{\text{init}}\boldsymbol{\beta}_0^T\mathbf{X}_i^T, \\
\mathbf{c}_{r1} &= \frac{1}{n}\sum_{i=1}^{n}\epsilon_i\mathbf{X}_i\boldsymbol{\beta}_{\text{init}}.
\end{aligned}$$

Now we derive the first order approximation of $\mathbf{A}_{r1}$, $\mathbf{B}_{r1}$ and $\mathbf{c}_{r1}$, and hence $\boldsymbol{\alpha}_{r(1)}$. The expectation of $\mathbf{A}_{r1}$ is given as follows,

$$\mathbf{A}_{r1*} = \mathrm{E}(\mathbf{A}_{r1}) = \left(\|\boldsymbol{\beta}_{\text{init}}\|_{\boldsymbol{\Psi}}^2 + \lambda_\beta\|\boldsymbol{\beta}_{\text{init}}\|_2^2\right)(\boldsymbol{\Sigma} + \lambda_\alpha\mathbf{I}).$$

Therefore, we can write $\mathbf{A}_{r1}$ as

$$\mathbf{A}_{r1} = \mathbf{A}_{r1*} + \tilde{\mathbf{A}}_{r1},$$

where $\tilde{\mathbf{A}}_{r1}$ is the term of small order.

Similarly, $\mathbf{B}_{r1}$ can be expressed as

$$\mathbf{B}_{r1} = \mathbf{B}_{r1*} + \tilde{\mathbf{B}}_{r1},$$

where $\mathbf{B}_{r1*} = \mathrm{E}(\mathbf{B}_{r1}) = (\boldsymbol{\beta}_{\text{init}}^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)\boldsymbol{\Sigma}$.

Applying the matrix inversion lemma (4.19) and only keeping the first order terms, it follows that

$$\boldsymbol{\alpha}_{r(1)} \approx \boldsymbol{\alpha}_{r\star} + \tilde{\boldsymbol{\alpha}}_{r(1)}, \tag{4.33}$$

where

$$\begin{aligned}\tilde{\boldsymbol{\alpha}}_{r(1)} &= \mathbf{A}_{r1*}^{-1}\tilde{\mathbf{B}}_{r1}\boldsymbol{\alpha}_0 + \mathbf{A}_{r1*}^{-1}\mathbf{c}_{r1} - \mathbf{A}_{r1*}^{-1}\tilde{\mathbf{A}}_{r1}\boldsymbol{\alpha}_{r\star} \\ &= \|\boldsymbol{\beta}_{\text{init}}\|_{\boldsymbol{\Psi}+\lambda_\beta\mathbf{I}}^{-2}(\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}(\tilde{\mathbf{B}}_{r1}\boldsymbol{\alpha}_0 - \tilde{\mathbf{A}}_{r1}\boldsymbol{\alpha}_{r\star} + \mathbf{c}_{r1}). \end{aligned} \tag{4.34}$$

Next, let $\boldsymbol{\beta}_{r(2)}$ be the updated value of $\boldsymbol{\beta}$ after the second iteration step, i.e., $\boldsymbol{\beta}_{r(2)} = \boldsymbol{\beta}_\lambda(\boldsymbol{\alpha}_\lambda(\boldsymbol{\beta}_{\text{init}}))$. By plugging $\boldsymbol{\alpha}_{r(1)}$ into the $\boldsymbol{\beta}_\lambda(\cdot)$ operator, we have

$$\boldsymbol{\beta}_{r(2)} = \mathbf{A}_{r2}^{-1}(\mathbf{B}_{r2}\boldsymbol{\beta}_0 + \mathbf{c}_{r2}),$$

where

$$\begin{aligned}\mathbf{A}_{r2} &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i^T\boldsymbol{\alpha}_{r(1)}\boldsymbol{\alpha}_{r(1)}^T\mathbf{X}_i + \lambda_\alpha\|\boldsymbol{\alpha}_{r(1)}\|_2^2\boldsymbol{\Psi} \\ &\quad +\lambda_\beta\|\boldsymbol{\alpha}_{r(1)}\|_2^2\mathbf{I} + \lambda_\alpha\lambda_\beta\|\boldsymbol{\alpha}_{r(1)}\|_2^2\mathbf{I}, \\ \mathbf{B}_{r2} &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i^T\boldsymbol{\alpha}_{r(1)}\boldsymbol{\alpha}_0^T\mathbf{X}_i, \\ \mathbf{c}_{r2} &= \frac{1}{n}\sum_{i=1}^{n}\epsilon_i\mathbf{X}_i^T\boldsymbol{\alpha}_{r(1)}. \end{aligned} \tag{4.35}$$

We analyze each term in $\mathbf{A}_{r2}$ and obtain their first order expansions. Let the four terms of $\mathbf{A}_{r2}$ in (4.35) be denoted as follows,

$$\mathbf{A}_{r21} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i^T\boldsymbol{\alpha}_{r(1)}\boldsymbol{\alpha}_{r(1)}^T\mathbf{X}_i, \quad \mathbf{A}_{r22} = \lambda_\alpha\|\boldsymbol{\alpha}_{r(1)}\|_2^2\boldsymbol{\Psi},$$

$$\mathbf{A}_{r23} = \lambda_\beta\|\boldsymbol{\alpha}_{r(1)}\|_2^2\mathbf{I}, \quad \text{and} \quad \mathbf{A}_{r24} = \lambda_\alpha\lambda_\beta\|\boldsymbol{\alpha}_{r(1)}\|_2^2\mathbf{I}.$$

With the notations of $\mathbf{D}_r$ and $\mathbf{E}_r$, by plugging $\boldsymbol{\alpha}_{r(1)}$ in (4.33) into the above expression and

keeping only the first order terms, we can obtain the following first order approximations,

$$\mathbf{A}_{r21} \approx \|\boldsymbol{\alpha}_{r\star}\|^2_{\boldsymbol{\Sigma}}\boldsymbol{\Psi} + 2(\tilde{\boldsymbol{\alpha}}_{r(1)}\boldsymbol{\Sigma}\boldsymbol{\alpha}_{r\star})\boldsymbol{\Psi} + \tilde{\mathbf{D}}_r, \mathbf{A}_{r22} \approx \lambda_\alpha\|\boldsymbol{\alpha}_{r\star}\|^2_2\boldsymbol{\Psi} + 2\lambda_\alpha(\boldsymbol{\alpha}^T_{r\star}\tilde{\boldsymbol{\alpha}}_{r(1)})\boldsymbol{\Psi},$$

$$\mathbf{A}_{r23} \approx \lambda_\beta\|\boldsymbol{\alpha}_{r\star}\|^2_{\boldsymbol{\Sigma}}\mathbf{I} + 2\lambda_\beta(\boldsymbol{\alpha}^T_{r\star}\boldsymbol{\Sigma}\tilde{\boldsymbol{\alpha}}_{r(1)})\mathbf{I}, \mathbf{A}_{r24} \approx \lambda_\alpha\lambda_\beta\|\boldsymbol{\alpha}_{r\star}\|^2_2 + 2\lambda_\alpha\lambda_\beta(\boldsymbol{\alpha}^T_{r(1)}\boldsymbol{\alpha}_{r\star})\mathbf{I}.$$

By collecting the leading terms and first order terms from $\mathbf{A}_{r21}$, $\mathbf{A}_{r22}$, $\mathbf{A}_{r23}$ and $\mathbf{A}_{r24}$, and denoting the leading term of $\mathbf{A}_{r2}$ by $\mathbf{A}_{r2*}$ and the first order term by $\tilde{\mathbf{A}}_{r2}$, the following holds:

$$\begin{aligned}
\mathbf{A}_{r2*} &= (\|\boldsymbol{\alpha}_{r\star}\|^2_{\boldsymbol{\Sigma}} + \lambda_\alpha\|\boldsymbol{\alpha}_{r\star}\|^2_2)(\boldsymbol{\Psi} + \lambda_\beta\mathbf{I}), && (4.36)\\
\tilde{\mathbf{A}}_{r2} &= 2(\tilde{\boldsymbol{\alpha}}^T_{r(1)}\boldsymbol{\Sigma}\boldsymbol{\alpha}_{r\star} + \lambda_\alpha(\boldsymbol{\alpha}^T_{r\star}\tilde{\boldsymbol{\alpha}}_{r(1)}))(\boldsymbol{\Psi} + \lambda_\beta\mathbf{I}) + \tilde{\mathbf{D}}_r. && (4.37)
\end{aligned}$$

Denote the leading term and the first order term of $\mathbf{B}_{r2}$ by $\mathbf{B}_{r2*}$ and $\tilde{\mathbf{B}}_{r2}$ respectively. Then we have

$$\begin{aligned}
\mathbf{B}_{r2*} &= (\boldsymbol{\alpha}^T_{r\star}\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)\boldsymbol{\Psi}, && (4.38)\\
\tilde{\mathbf{B}}_{r2} &= (\tilde{\boldsymbol{\alpha}}^T_{r(1)}\boldsymbol{\Sigma}\boldsymbol{\alpha}_0)\boldsymbol{\Psi} + \tilde{\mathbf{E}}_r. && (4.39)
\end{aligned}$$

By Equations (4.19), (4.36), (4.38), the first order approximation of $\boldsymbol{\beta}_{r(2)}$ can be expressed as

$$\boldsymbol{\beta}_{r(2)} \approx \boldsymbol{\beta}_{r\star} + \mathbf{A}^{-1}_{r2*}\mathbf{c}_{r2} + (\mathbf{A}^{-1}_{r2*}\tilde{\mathbf{B}}_{r2}\boldsymbol{\beta}_0 - \mathbf{A}^{-1}_{r2*}\tilde{\mathbf{A}}_{r2}\boldsymbol{\beta}_{r\star}) = \boldsymbol{\beta}_{r\star} + \tilde{\boldsymbol{\beta}}_{r(2)}. \quad (4.40)$$

We further define the following two quantities,

$$\begin{aligned}
\mathbf{c}_{r\alpha} &= \frac{1}{n}\sum_{i=1}^n \mathbf{X}^T_i\boldsymbol{\alpha}_{r\star}\epsilon_i, && (4.41)\\
\mathbf{c}_{r\beta} &= \frac{1}{n}\sum_{i=1}^n \mathbf{X}_i\boldsymbol{\beta}_{r\star}\epsilon_i. && (4.42)
\end{aligned}$$

The leading error term in $\mathbf{c}_{r2}$ is $\mathbf{c}_{r\alpha}$ and $\frac{1}{n}\sum_{i=1}^n \mathbf{X}^T_i\tilde{\boldsymbol{\alpha}}_{r(1)}\epsilon_i$ is of second order.

Together with plugging (4.36), (4.37), (4.38), (4.39) into (4.40) and further replacing $\tilde{\boldsymbol{\alpha}}_{r(1)}$ by (4.34), one achieves

$$\begin{aligned}\tilde{\boldsymbol{\beta}}_{r(2)} &= -\|\boldsymbol{\alpha}_{r\star}\|^{-2}_{\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I}}\|\boldsymbol{\beta}_{\text{init}}\|^{-2}_{\boldsymbol{\Psi}+\lambda_\beta\mathbf{I}}\boldsymbol{\alpha}_{r\star}^T(\tilde{\mathbf{B}}_{r1}\boldsymbol{\alpha}_0 - \tilde{\mathbf{A}}_{r1}\boldsymbol{\alpha}_{r\star} + \mathbf{c}_{r1})\boldsymbol{\beta}_{r\star} \\ &\quad +\|\boldsymbol{\alpha}_{r\star}\|^{-2}_{\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I}}(\boldsymbol{\Psi}+\lambda_\beta\mathbf{I})^{-1}(\tilde{\mathbf{E}}_r\boldsymbol{\beta}_0 - \tilde{\mathbf{D}}_r\boldsymbol{\beta}_{r\star} + \mathbf{c}_{r\alpha}). \qquad (4.43)\end{aligned}$$

By identical argument to derive (4.43), we can obtain $\tilde{\boldsymbol{\alpha}}_{r(3)}$. The first order error for estimating the Kronecker product $\boldsymbol{\beta}_{r\star} \otimes \boldsymbol{\alpha}_{r\star}$ after some cancellation is given as follows,

$$\begin{aligned}&\boldsymbol{\beta}_{r\star} \otimes \tilde{\boldsymbol{\alpha}}_{r(3)} + \tilde{\boldsymbol{\beta}}_{r(2)} \otimes \boldsymbol{\alpha}_{r\star} \\ =\ & \|\boldsymbol{\beta}_{r\star}\|^{-2}_{\boldsymbol{\Psi}+\lambda_\beta\mathbf{I}}\boldsymbol{\beta}_{r\star} \otimes (\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0 - \tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star} + \mathbf{c}_{r\beta}) \\ &+\|\boldsymbol{\alpha}_{r\star}\|^{-2}_{\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I}}(\boldsymbol{\Psi}+\lambda_\beta\mathbf{I})^{-1}(\tilde{\mathbf{E}}_r\boldsymbol{\beta}_0 - \tilde{\mathbf{D}}_r\boldsymbol{\beta}_{r\star} + \mathbf{c}_{r\alpha}) \otimes \boldsymbol{\alpha}_{r\star} \\ &-\|\boldsymbol{\beta}_{r\star}\|^{-2}_{\boldsymbol{\Psi}+\lambda_\beta\mathbf{I}}\|\boldsymbol{\alpha}_{r\star}\|^{-2}_{\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I}}[\boldsymbol{\alpha}_{r\star}^T(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0 - \tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star} + \mathbf{c}_{r\beta})](\boldsymbol{\beta}_{r\star} \otimes \boldsymbol{\alpha}_{r\star}). \qquad (4.44)\end{aligned}$$

In what follows, we will denote the three terms on the right hand of the equation by s_1, s_2, s_3 respectively.

The excess prediction error can be bounded as

$$\begin{aligned}&\|\boldsymbol{\beta}_{r(2)} \otimes \boldsymbol{\alpha}_{r(3)} - \boldsymbol{\beta}_{r\star} \otimes \boldsymbol{\alpha}_{r\star}\|^2_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}} \\ =\ & \|s_1 + s_2 + s_3\|^2_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}} \\ \leq\ & 3\|s_1\|^2_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}} + 3\|s_2\|^2_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}} + 3\|s_3\|^2_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}. \qquad (4.45)\end{aligned}$$

Our task is to bound the three terms separately. The following lemmas give the upper bound for these terms.

Lemma 2 *Let s_1 and s_2 be defined in* (4.44)*. We have*

$$E\|s_1\|^2_{\mathbf{\Psi}\otimes\mathbf{\Sigma}} \leq 2n^{-1}d_\alpha(\tau^2 + \|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star} - \boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|^2_{\mathbf{\Psi}\otimes\mathbf{\Sigma}})(1+o(1)), \tag{4.46}$$

$$E\|s_2\|^2_{\mathbf{\Psi}\otimes\mathbf{\Sigma}} \leq 2n^{-1}d_\beta(\tau^2 + \|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star} - \boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|^2_{\mathbf{\Psi}\otimes\mathbf{\Sigma}})(1+o(1)). \tag{4.47}$$

Lemma 3 *Let s_3 be defined in* (4.44)*. The following holds,*

$$E\|s_3\|^2_{\mathbf{\Psi}\otimes\mathbf{\Sigma}} \leq 2n^{-1}(\tau^2 + \|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star} - \boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|^2_{\mathbf{\Psi}\otimes\mathbf{\Sigma}}). \tag{4.48}$$

Now Equations (4.45), (4.46), (4.47) and (4.48) complete the proof of Lemma 1.

Proof of Lemma 2 By the Kronecker product property, we have

$$\begin{aligned}
& \|s_1\|^2_{\mathbf{\Psi}\otimes\mathbf{\Sigma}} \\
= \;& \|\boldsymbol{\beta}_{r\star}\|^{-4}_{\mathbf{\Psi}+\lambda_\beta\mathbf{I}}\|\boldsymbol{\beta}_{r\star}\|^2_{\mathbf{\Psi}}\|(\mathbf{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0 - \tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star} + \mathbf{c}_{r\beta})\|^2_{\mathbf{\Sigma}} \\
\leq \;& 2\|\boldsymbol{\beta}_{r\star}\|^{-4}_{\mathbf{\Psi}+\lambda_\beta\mathbf{I}}\|\boldsymbol{\beta}_{r\star}\|^2_{\mathbf{\Psi}}\|(\mathbf{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}\mathbf{c}_{r\beta}\|^2_{\mathbf{\Sigma}} \\
& +2\|\boldsymbol{\beta}_{r\star}\|^{-4}_{\mathbf{\Psi}+\lambda_\beta\mathbf{I}}\|\boldsymbol{\beta}_{r\star}\|^2_{\mathbf{\Psi}}\|(\mathbf{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0 - \tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star})\|^2_{\mathbf{\Sigma}} \\
\overset{def.}{=} \;& s_{11} + s_{12}.
\end{aligned}$$

Recall the definition of $\mathbf{c}_{r\beta}$ in (4.42), it follows

$$\begin{aligned}
& E\|(\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}\mathbf{c}_{r\beta}\|_{\boldsymbol{\Sigma}}^2 \\
= \; & E\|(\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}\frac{1}{n}\sum_{i=1}^n \mathbf{X}_i\boldsymbol{\beta}_{r\star}\epsilon_i\|_{\boldsymbol{\Sigma}}^2 \\
= \; & n^{-2}E\left((\sum_{i=1}^n \epsilon_i\boldsymbol{\beta}_{r\star}^T\mathbf{X}_i^T)(\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}(\sum_{i=1}^n \mathbf{X}_i\boldsymbol{\beta}_{r\star}\epsilon_i)\right) \\
= \; & \tau^2 n^{-2}E\left((\sum_{i=1}^n \boldsymbol{\beta}_{r\star}^T\mathbf{X}_i^T)(\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}(\sum_{i=1}^n \mathbf{X}_i\boldsymbol{\beta}_{r\star})\right) \\
= \; & \tau^2 n^{-1}E\left(\boldsymbol{\beta}_{r\star}^T\mathbf{X}^T(\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}\mathbf{X}\boldsymbol{\beta}_{r\star}\right) \\
= \; & \tau^2 n^{-1}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2 \mathrm{tr}\left((\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I})^{-1}\boldsymbol{\Sigma}\right) \\
= \; & \tau^2 n^{-1}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2 \sum_{i=1}^n \frac{\lambda_i^2}{(\lambda_i+\lambda_\alpha)^2} \\
= \; & \frac{\tau^2 d_{\boldsymbol{\alpha}}}{n}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2, \qquad (4.49)
\end{aligned}$$

where the third equality uses the fact that $\mathbf{X}$ and ϵ are independent, the fifth equality comes from the property of matrix normal distribution, and the last one is based on the definition of $d_{\boldsymbol{\alpha}}$.

Hence, (4.49) implies that

$$\begin{aligned}
Es_{11} &= 2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}+\lambda_\beta\mathbf{I}}^{-4}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^4\frac{\tau^2 d_{\boldsymbol{\alpha}}}{n} \\
&= \frac{2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^4}{(\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2+\lambda_\beta\|\boldsymbol{\beta}_{r\star}\|_2^2)^2}\frac{\tau^2 d_{\boldsymbol{\alpha}}}{n} \\
&\leq 2\frac{\tau^2 d_{\boldsymbol{\alpha}}}{n}.
\end{aligned}$$

As for s_{12}, we know that

$$
\begin{aligned}
& E\|(\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0-\tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star})\|_{\boldsymbol{\Sigma}}^2 \\
= \;& E(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0-\tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star})^T(\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0-\tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star}) \\
= \;& E\mathrm{tr}((\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0-\tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star})(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0-\tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star})^T) \\
= \;& \mathrm{tr}\left((\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}E(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0-\tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star})(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0-\tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star})^T\right) \\
= \;& \mathrm{tr}\Big((\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}E(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T\tilde{\mathbf{G}}_r^T+\tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star}\boldsymbol{\alpha}_{r\star}^T\tilde{\mathbf{F}}_r^T \\
& -2\tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star}\boldsymbol{\alpha}_0^T\tilde{\mathbf{G}}_r^T)\Big),
\end{aligned}
$$

where we have used the trick to interchange expectation and trace.

Since $\tilde{\mathbf{G}}_r=\mathbf{G}_r-E\mathbf{G}_r$,

$$
\begin{aligned}
& E(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T\tilde{\mathbf{G}}_r^T) \\
= \;& \mathrm{cov}(\mathbf{G}_r\boldsymbol{\alpha}_0) \\
= \;& \mathrm{cov}(n^{-1}\sum_{i=1}^n \mathbf{X}_i\boldsymbol{\beta}_{r\star}\boldsymbol{\beta}_0^T\mathbf{X}_i^T\boldsymbol{\alpha}_0) \\
= \;& n^{-1}\mathrm{cov}(\mathbf{X}\boldsymbol{\beta}_{r\star}\boldsymbol{\beta}_0^T\mathbf{X}^T\boldsymbol{\alpha}_0) \\
= \;& n^{-1}\mathrm{tr}(\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star}\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_0\boldsymbol{\beta}_{r\star}^T)\mathrm{tr}(\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma})\boldsymbol{\Sigma} \\
& +n^{-1}\mathrm{tr}(\boldsymbol{\beta}_{r\star}\boldsymbol{\beta}_0^T\boldsymbol{\Psi})\mathrm{tr}(\boldsymbol{\beta}_0\boldsymbol{\beta}_{r\star}^T\boldsymbol{\Psi})\boldsymbol{\Sigma}\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma} \\
& +n^{-1}\mathrm{tr}(\boldsymbol{\beta}_{r\star}\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star}\boldsymbol{\beta}_0^T\boldsymbol{\Psi})\boldsymbol{\Sigma}\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma} \\
& -n^{-1}(\boldsymbol{\beta}_{r\star}^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)\boldsymbol{\Sigma}\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T((\boldsymbol{\beta}_{r\star}^T\boldsymbol{\Psi}\boldsymbol{\beta}_0)\boldsymbol{\Sigma})^T \\
= \;& n^{-1}\|\boldsymbol{\alpha}_0\|_{\boldsymbol{\Sigma}}^2\|\boldsymbol{\beta}_0\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2\boldsymbol{\Sigma}+n^{-1}(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star})^2\boldsymbol{\Sigma}\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma},
\end{aligned}
$$

where the second last step replies upon the matrix normal distribution property again and the rest steps are standard.

A similar derivation will produce

$$E(\tilde{\mathbf{F}}_r \boldsymbol{\alpha}_{r\star}\boldsymbol{\alpha}_{r\star}^T\tilde{\mathbf{F}}_r^T) = n^{-1}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^4\|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}}^2\boldsymbol{\Sigma} + n^{-1}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^4\boldsymbol{\Sigma}\boldsymbol{\alpha}_{r\star}\boldsymbol{\alpha}_{r\star}^T\boldsymbol{\Sigma}.$$

For the cross-product term, since $\tilde{\mathbf{G}}_r = \mathbf{G}_r - E\mathbf{G}_r$ and $\tilde{\mathbf{F}}_r = \mathbf{F}_r - E\mathbf{F}_r$,

$$\begin{aligned}
& E(\tilde{\mathbf{F}}_r \boldsymbol{\alpha}_{r\star}\boldsymbol{\alpha}_0^T\tilde{\mathbf{G}}_r^T) \\
= \;& \mathrm{cov}(\mathbf{F}_r\boldsymbol{\alpha}_{r\star}, \mathbf{G}_r\boldsymbol{\alpha}_0) \\
= \;& \mathrm{cov}(n^{-1}\sum_{i=1}^n \mathbf{X}_i\boldsymbol{\beta}_{r\star}\boldsymbol{\beta}_{r\star}^T\mathbf{X}_i^T\boldsymbol{\alpha}_{r\star}, n^{-1}\sum_{i=1}^n \mathbf{X}_i\boldsymbol{\beta}_{r\star}\boldsymbol{\beta}_0^T\mathbf{X}_i^T\boldsymbol{\alpha}_0) \\
= \;& n^{-1}\mathrm{cov}(\mathbf{X}\boldsymbol{\beta}_{r\star}\boldsymbol{\beta}_{r\star}^T\mathbf{X}^T\boldsymbol{\alpha}_{r\star}, \mathbf{X}\boldsymbol{\beta}_{r\star}\boldsymbol{\beta}_0^T\mathbf{X}^T\boldsymbol{\alpha}_0) \\
= \;& n^{-1}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_{r\star})(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star})\boldsymbol{\Sigma} + n^{-1}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star})\boldsymbol{\Sigma}\boldsymbol{\alpha}_0\boldsymbol{\alpha}_{r\star}^T\boldsymbol{\Sigma}.
\end{aligned}$$

Taking the last three displays into consideration, we can group them into two parts

$$\begin{aligned}
& E(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T\tilde{\mathbf{G}}_r^T + \tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star}\boldsymbol{\alpha}_{r\star}^T\tilde{\mathbf{F}}_r^T - 2\tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star}\boldsymbol{\alpha}_0^T\tilde{\mathbf{G}}_r^T) \\
= \;& n^{-1}\left(\|\boldsymbol{\alpha}_0\|_{\boldsymbol{\Sigma}}^2\|\boldsymbol{\beta}_0\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2\boldsymbol{\Sigma} + \|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^4\|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}}^2\boldsymbol{\Sigma}\right. \\
& \left.-2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_{r\star})(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star})\boldsymbol{\Sigma}\right) \\
& +n^{-1}\left((\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star})^2\boldsymbol{\Sigma}\boldsymbol{\alpha}_0\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma} + \|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^4\boldsymbol{\Sigma}\boldsymbol{\alpha}_{r\star}\boldsymbol{\alpha}_{r\star}^T\boldsymbol{\Sigma}\right. \\
& \left.-2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star})\boldsymbol{\Sigma}\boldsymbol{\alpha}_0\boldsymbol{\alpha}_{r\star}^T\boldsymbol{\Sigma}\right).
\end{aligned}$$

Once this is plugged back

$$
\begin{aligned}
& E\|(\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0-\tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star})\|_{\boldsymbol{\Sigma}}^2 \\
= \; & n^{-1}\mathrm{tr}\left((\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}\boldsymbol{\Sigma}\right)\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2 \\
& \left(\|\boldsymbol{\alpha}_0\|_{\boldsymbol{\Sigma}}^2\|\boldsymbol{\beta}_0\|_{\boldsymbol{\Psi}}^2+\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}}^2-2(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_{r\star})(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star})\right) \\
& +n^{-1}\left((\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star})^2\|\boldsymbol{\alpha}_0\|_{\tilde{\boldsymbol{\Sigma}}}^2+\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^4\|\boldsymbol{\alpha}_{r\star}\|_{\tilde{\boldsymbol{\Sigma}}}^2\right. \\
& \left.-2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star})(\boldsymbol{\alpha}_0^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{\alpha}_{r\star})\right),
\end{aligned}
$$

where $\tilde{\boldsymbol{\Sigma}}=\boldsymbol{\Sigma}(\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}\boldsymbol{\Sigma}$

The first three terms on the right hand of the last equation can be further simplified to

$$
n^{-1}d_{\boldsymbol{\alpha}}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star}-\boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}^2,
$$

and the last three terms can be bounded by

$$
\begin{aligned}
& n^{-1}\left(\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\beta}_0\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\alpha}_0\|_{\tilde{\boldsymbol{\Sigma}}}^2+\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^4\|\boldsymbol{\alpha}_{r\star}\|_{\tilde{\boldsymbol{\Sigma}}}^2\right. \\
& \left.-2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star})(\boldsymbol{\alpha}_0^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{\alpha}_{r\star})\right) \\
= \; & n^{-1}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star}-\boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi}\otimes\tilde{\boldsymbol{\Sigma}}}^2 \\
\leq \; & n^{-1}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star}-\boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}^2,
\end{aligned}
$$

where the first line is a simple application of Cauchy-Schwarz inequality and the last line holds because $\boldsymbol{\Sigma}-\tilde{\boldsymbol{\Sigma}}$ is positive definite.

In all,

$$
\begin{aligned}
Es_{12} &= 2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}+\lambda_\beta\mathbf{I}}^{-4}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2E\|(\boldsymbol{\Sigma}+\lambda_\alpha \mathbf{I})^{-1}(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0-\tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star})\|_{\boldsymbol{\Sigma}}^2 \\
&\leq 2n^{-1}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}+\lambda_\beta\mathbf{I}}^{-4}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^4(d_{\boldsymbol{\alpha}}+1)\|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star}-\boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}^2 \\
&\leq 2n^{-1}d_{\boldsymbol{\alpha}}\|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star}-\boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}^2(1+o(1))
\end{aligned}
$$

Now we have

$$E\|s_1\|^2_{\mathbf{\Psi}\otimes\mathbf{\Sigma}} \leq 2n^{-1}d_{\boldsymbol{\alpha}}(\tau^2 + \|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star} - \boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|^2_{\mathbf{\Psi}\otimes\mathbf{\Sigma}})(1+o(1))$$

And switching the role of α and β will prove

$$E\|s_2\|^2_{\mathbf{\Psi}\otimes\mathbf{\Sigma}} \leq 2n^{-1}d_{\boldsymbol{\beta}}(\tau^2 + \|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star} - \boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|^2_{\mathbf{\Psi}\otimes\mathbf{\Sigma}})(1+o(1))$$

Proof of Lemma 3 The expectation of s_3 can be written as

$$\begin{aligned}
& E\|s_3\|^2_{\mathbf{\Psi}\otimes\mathbf{\Sigma}} \\
= \;& \|\boldsymbol{\beta}_{r\star}\|^{-4}_{\mathbf{\Psi}+\lambda_\beta\mathbf{I}}\|\boldsymbol{\alpha}_{r\star}\|^{-4}_{\mathbf{\Sigma}+\lambda_\alpha\mathbf{I}}\|\boldsymbol{\alpha}_{r\star}\|^2_{\mathbf{\Sigma}}\|\boldsymbol{\beta}_{r\star}\|^2_{\mathbf{\Psi}}E[\boldsymbol{\alpha}^T_{r\star}(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0 - \tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star} + \mathbf{c}_{r\beta})]^2 \\
\leq \;& 2\|\boldsymbol{\beta}_{r\star}\|^{-4}_{\mathbf{\Psi}+\lambda_\beta\mathbf{I}}\|\boldsymbol{\alpha}_{r\star}\|^{-4}_{\mathbf{\Sigma}+\lambda_\alpha\mathbf{I}}\|\boldsymbol{\alpha}_{r\star}\|^2_{\mathbf{\Sigma}}\|\boldsymbol{\beta}_{r\star}\|^2_{\mathbf{\Psi}} \\
& \Big(E[\boldsymbol{\alpha}^T_{r\star}\mathbf{c}_{r\beta}]^2 + E[\boldsymbol{\alpha}^T_{r\star}(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0 - \tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star})]^2\Big) \\
\stackrel{def}{=} \;& s_{31} + s_{32}.
\end{aligned}$$

Since

$$\begin{aligned}
& E[\boldsymbol{\alpha}^T_{r\star}\mathbf{c}_{r\beta}]^2 \\
= \;& E[\boldsymbol{\alpha}^T_{r\star}\frac{1}{n}\sum_{i=1}^n \mathbf{X}_i\boldsymbol{\beta}_{r\star}\epsilon_i]^2 \\
= \;& \frac{\tau^2}{n}E[\boldsymbol{\alpha}^T_{r\star}\mathbf{X}\boldsymbol{\beta}_{r\star}]^2 \\
= \;& \frac{\tau^2}{n}\|\boldsymbol{\alpha}_{r\star}\|^2_{\mathbf{\Sigma}}\|\boldsymbol{\beta}_{r\star}\|^2_{\mathbf{\Psi}},
\end{aligned}$$

we have

$$s_{31} = 2\frac{\tau^2}{n}\|\boldsymbol{\beta}_{r\star}\|^{-4}_{\mathbf{\Psi}+\lambda_\beta\mathbf{I}}\|\boldsymbol{\alpha}_{r\star}\|^{-4}_{\mathbf{\Sigma}+\lambda_\alpha\mathbf{I}}\|\boldsymbol{\alpha}_{r\star}\|^4_{\mathbf{\Sigma}}\|\boldsymbol{\beta}_{r\star}\|^4_{\mathbf{\Psi}} \leq 2\frac{\tau^2}{n}.$$

As for s_{32}, we have

$$\begin{aligned}
& E[\boldsymbol{\alpha}_{r\star}^T(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0 - \tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star})]^2 \\
= \;& \mathrm{tr}(\boldsymbol{\alpha}_{r\star}\boldsymbol{\alpha}_{r\star}^T E((\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0 - \tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star})(\tilde{\mathbf{G}}_r\boldsymbol{\alpha}_0 - \tilde{\mathbf{F}}_r\boldsymbol{\alpha}_{r\star})^T)) \\
= \;& n^{-1}\|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}}^2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star} - \boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}^2 \\
& +n^{-1}\left((\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star})^2(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_{r\star})^2 + \|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^4\|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}}^4\right. \\
& \left.-2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}}^2(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star})(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_{r\star})\right) \\
\leq \;& n^{-1}\|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}}^2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star} - \boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}^2 \\
& +n^{-1}\left(\|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}}^2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\alpha}_0\|_{\boldsymbol{\Sigma}}^2\|\boldsymbol{\beta}_0\|_{\boldsymbol{\Psi}}^2 + \|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^4\|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}}^4\right. \\
& \left.-2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}}^2(\boldsymbol{\beta}_0^T\boldsymbol{\Psi}\boldsymbol{\beta}_{r\star})(\boldsymbol{\alpha}_0^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_{r\star})\right) \\
= \;& 2n^{-1}\|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}}^2\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^2\|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star} - \boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}^2,
\end{aligned}$$

which implies

$$\begin{aligned}
s_{32} \;\leq\; & 2\frac{1}{n}\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}+\lambda_\beta\mathbf{I}}^{-4}\|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}+\lambda_\alpha\mathbf{I}}^{-4}\|\boldsymbol{\alpha}_{r\star}\|_{\boldsymbol{\Sigma}}^4\|\boldsymbol{\beta}_{r\star}\|_{\boldsymbol{\Psi}}^4 \\
& \|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star} - \boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}^2 \\
\leq \;& 2\frac{1}{n}\|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star} - \boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}^2.
\end{aligned}$$

The analysis of s_{31} and s_{32} shows that

$$E\|s_3\|_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}^2 \leq 2n^{-1}(\tau^2 + \|\boldsymbol{\beta}_{r\star}\otimes\boldsymbol{\alpha}_{r\star} - \boldsymbol{\beta}_0\otimes\boldsymbol{\alpha}_0\|_{\boldsymbol{\Psi}\otimes\boldsymbol{\Sigma}}^2).$$

CHAPTER 5: FUTURE WORK

We intend to undertake the following two projects in the future. The first one is an extension of the bilinear regression and the second one is in the area of covariance matrix estimation.

The first project is an extension of the functional linear regression. The traditional one-way functional linear regression takes a one dimensional functional predictor. Recently, two-way functional data are becoming more and more often. We will extend the functional linear model to bilinear functional model and propose a bilinear functional estimator. We aim at studying the theoretical properties of the estimator under the framework of Reproducing Kernel Hilbert Space.

The second project is on covariance matrix estimation and testing, which is an important problem in multivariate analysis. There has been a large amount of literature on this subject for vector-valued data. As for matrix-valued data, usually both column-column and row-row correlations exist, which makes us model the observed data via separable covariance matrices with one characterizing correlation among columns and the other one among rows. There has been some recent work on the estimation of separable covariance matrices, but under the traditional fixed number of parameters and increasing sample size setup. We aim to study the high dimensional estimation problem with the number of parameters going to infinity under two possible additional assumptions: sparse or banded covariance matrix. Moreover, the hypothesis testing problem of some pre-specified structure for the covariance matrix for vector-valued data has emerged recently as well. We will extend the testing procedure proposed in Cai et al. (2013) to matrix-valued data. Lastly, we will analyze the theoretical and numerical properties of the estimation and testing procedures and apply them to real data analysis.

REFERENCES

G. I. Allen and R. Tibshirani. Inference with transposable data: modelling the effects of row and column correlations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(4):721–743, 2012.

S. Amari, A. Cichocki, and H. H. Yang. A New Learning Algorithm for Blind Signal Separation. In *Advances in Neural Information Processing Systems*, pages 757–763. MIT Press, 1996.

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *The Journal of Machine Learning Research*, 3:1–48, 2002.

C. F. Beckmann and S. M. Smith. Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE Transactions on Medical Imaging,*, 23(2):137–152, 2004.

C. F. Beckmann and S. M. Smith. Tensorial extensions of independent component analysis for multisubject FMRI analysis. *NeuroImage*, 25:294–311, 2005.

A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

D. R. Brillinger. *Time series: data analysis and theory*, volume 36. SIAM, 2001.

P. J. Brockwell and R. A. Davis. *Time series: theory and methods*. Springer, 2009.

T. Cai, W. Liu, and Y. Xia. Two-sample covariance matrix testing and support recovery in high-dimensional and sparse settings. *Journal of the American Statistical Association*, 108 (501):265–277, 2013.

V. D. Calhoun and T. Adali. Multisubject independent component analysis of fMRI: a decade of intrinsic networks, default mode, and neurodiagnostic discovery. *Biomedical Engineering, IEEE Reviews in*, 5:60–73, 2012.

V. D. Calhoun, T. Adali, G. D. Pearlson, and J. J. Pekar. A method for making group inferences from functional MRI data using independent component analysis. *Human Brain Mapping*, 14:140–151, 2001.

V. D. Calhoun, J. Liu, and T. Adali. A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data. *NeuroImage*, 45:S163–172, 2009.

C. M. Crainiceanu, B. S. Caffo, S. Luo, V. M. Zipunnikov, and N. M. Punjabi. Population Value Decomposition, a Framework for the Analysis of Image Populations. *Journal of the American Statistical Association*, 106(495):775–790, 2011.

C. Ding and J. Ye. 2-Dimensional Singular Value Decomposition for 2D Maps and Images. In *Proc. SIAM Int'l Conf. Data Mining (SDM'05)*, pages 32–43, 2005.

M. Dyrholm, C. Christoforou, and L. C. Parra. Bilinear discriminant component analysis. *The Journal of Machine Learning Research*, 8:1097–1111, 2007.

K. Dzhaparidze and S. Kotz. *Parameter Estimation and Hypothesis Testing in Spectral Analysis of Stationary Time Series* . New York: Springer, 1986.

F. Esposito, T. Scarabino, A. Hyvarinen, J. Himberg, E. Formisano, S. Comani, G. Tedeschi, R. Goebel, E. Seifritz, and F. Di Salle. Independent component analysis of fMRI group studies by self-organizing clustering. *NeuroImage*, 25:193–205, 2005.

G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

Y. Guo. A general probabilistic model for group independent component analysis and its estimation methods. *Biometrics*, 67:1532–1542, 2011.

Y. Guo and G. Pagnoni. A unified framework for group independent component analysis for multi-subject fMRI data. *NeuroImage*, 42(3):1078 – 1093, 2008.

Y. Guo and L. Tang. A Hierarchical Model for Probabilistic Independent Component Analysis of Multi-Subject fMRI Studies. *Biometrics*, 69(4):970–981, 2013.

A. K. Gupta and D. K. Nagar. *Matrix Variate Distributions*. Chapman & Hall/CRC, Boca Raton, FL, 2000.

T. Hastie and R. Tibshirani. Independent Components Analysis through Product Density Estimation. In *Advances in Neural Information Processing Systems*, pages 649–656, 2002.

D. Hsu, S. M. Kakade, and T. Zhang. Random design analysis of ridge regression. *Foundations of Computational Mathematics*, 14(3):569–600, 2014.

H. Hung and C. Wang. Matrix variate logistic regression model with application to EEG data. *Biostatistics*, 14(1):189–202, 2013.

A. Hyvärinen. Independent component analysis: recent advances. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984), 2013.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, New York, 2001.

I. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002.

T. Kollo and D. von Rosen. *Advanced multivariate statistics with matrices*, volume 579. Springer, 2006.

S. Lee, H. Shen, Y. Truong, M. Lewis, and X. Huang. Independent Component Analysis Involving Autocorrelated Sources With an Application to Functional Magnetic Resonance Imaging. *Journal of the American Statistical Association*, 106(495):1009–1024, 2011.

C. Leng and C. Y. Tang. Sparse matrix graphical models. *Journal of the American Statistical Association*, 107(499):1187–1200, 2012.

B. Li, M. K. Kim, and N. Altman. On dimension folding of matrix-or array-valued statistical objects. *Annals of Statistics*, 38(2):1094–1121, 2010.

M. J. McKeown, S. Makeig, G. G. Brown, T. Jung, S. S. Kindermann, A. J. Bell, and T. J. Sejnowski. Analysis of fMRI data by blind separation into independent spatial components. *Human Brain Mapping*, 6(3):160–188, 1998.

M. P. Milham, D. Fair, M. Mennes, and S. H. Mostofsky. The adhd-200 consortium: a model to advance the translational potential of neuroimaging in clinical neuroscience. *Frontiers in Systems Neuroscience*, 6(62), 2012.

C. A. Rohde. Generalized inverses of partitioned matrices. *Journal of the Society for Industrial & Applied Mathematics*, 13(4):1033–1035, 1965.

F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 138–142, 1994.

A. Shapiro. Asymptotic theory of overparameterized structural models. *Journal of the American Statistical Association*, 81(393):142–149, 1986.

P. Stoica and T. L. Marzetta. Parameter estimation problems with singular information matrices. *IEEE Transactions on Signal Processing*, 49(1):87–90, 2001a.

P. Stoica and T. L. Marzetta. Parameter estimation problems with singular information matrices. *IEEE Transactions on Signal Processing*, 49(1):87–90, January 2001b.

J. V. Stone. *Independent Component Analysis: A Tutorial Introduction*. MIT Press, Cambridge, MA, 2004.

R. Vigário, J. Sarela, V. Jousmiki, M. Hamalainen, and E. Oja. Independent component approach to the analysis of EEG and MEG recordings. *Biomedical Engineering, IEEE Transactions on*, 47(5):589–593, 2000.

K. Werner, M. Jansson, and P. Stoica. On estimation of covariance matrices with Kronecker product structure. *IEEE Transactions on Signal Processing*, 56(2):478–491, 2008.

P. Whittle. Some Results in Time Series Analysis. *Skandivanisk Aktuarietidskrift*, 35:48–60, 1952.

K. J. Worsley, C. H. Liao, J. Aston, V. Petre, G. H. Duncan, F. Morales, and A. C. Evans. A General Statistical Analysis for fMRI Data. *NeuroImage*, 15(1):1–15, 2002.

J. Yang, D. Zhang, A. F. Frangi, and J. Yang. Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(1):131–137, 2004.

J. Ye. Generalized Low Rank Approximations of Matrices. *Machine Learning*, 61(1-3):167–191, 2005.

J. Yin and H. Li. Model selection and estimation in the matrix normal graphical model. *Journal of Multivariate Analysis*, 107(0):119 – 140, 2012.

D. Zhang and Z. Zhou. $(2D)^2$PCA: Two-directional two-dimensional PCA for efficient face representation and recognition. *Neurocomputing*, 69(1-3):224–231, 2005.

J. Zhao and C. Leng. Structured lasso for regression with matrix covariates. *Statistica Sinica*, 24:799–814, 2014.

H. Zhou and L. Li. Regularized matrix regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):463–483, 2014.

H. Zhou, L. Li, and H. Zhu. Tensor Regression with Applications in Neuroimaging Data Analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.

S. Zhou. Gemini: Graph estimation with matrix variate normal instances. *Annals of Statistics*, 42(2):532–562, 2014.