Regularization Techniques for Statistical Methods Utilizing Matrix/Tensor Decompositions

by

Joseph Carlton Poythress III

(Under the Direction of Jeongyoun Ahn and Cheolwoo Park)

Abstract

Many multivariate statistical methods rely on matrix decompositions. An archetypal example is canonical correlation analysis (CCA). In the traditional large sample setting, CCA admits an analytical solution: the best rank-k approximation of the sample version of a matrix. In high dimension, low sample size settings, it is not possible to calculate the analytical solution because it requires the inverse of a matrix for which the sample version is singular. In that situation, additional assumptions or regularization are necessary. Matrix decompositions can also be utilized in statistical methods that do not inherently rely on them. For a generalized linear model (GLM) with an image as the covariate, the corresponding parameter is a matrix or higher-order array called a tensor. In that case, we may exploit low rank matrix or tensor decompositions as a means to reduce the massive number of parameters to estimate to a feasible level. In this dissertation, we study two regularization techniques for statistical methods utilizing matrix/tensor decompositions, with emphasis on their applications in high dimensional CCA and GLMs with matrix- or tensor-valued parameters. One technique is sparse regularization that results in variable selection. In high dimensional problems, variable selection can substantially improve the interpretability of the solution.

The other technique is a penalty on the nuclear norm, which amounts to soft thresholding (i.e., shrinking) the singular values. For low rank approximation problems, shrinking the singular values can be an effective alternative to finding a fixed rank-k approximation. For GLMs with matrix- or tensor-valued parameters, we develop both fixed-rank and shrinkage versions of an orthogonal tensor regression model, which we intend for analyzing medical imaging data such as fMRI. For high dimensional CCA, we develop a sparse CCA method that achieves variable selection by penalizing the elements of the canonical vectors. We study the variable selection accuracy under different choices of the penalty function. We also develop a more general method of finding a sparse, low rank matrix approximation based on shrinkage, which we show aims to select the same variables as certain sparse CCA methods under some assumptions.

INDEX WORDS: Low rank approximation, Nuclear norm, Sparsity, High dimension low sample size, Imaging data, Multimodal data.

Regularization Techniques for Statistical Methods Utilizing Matrix/Tensor Decompositions

by

Joseph Carlton Poythress III

B.S., The University of North Carolina at Chapel Hill, 2009

M.S., University of Georgia, 2015

M.S., University of Georgia, 2018

A Dissertation Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2020

Regularization Techniques for Statistical Methods Utilizing Matrix/Tensor Decompositions

by

Joseph Carlton Poythress III

Approved:

Major Professors:	Jeongyoun Ahn Cheolwoo Park
Committee:	Pengsheng Ji Nicole Lazar Liang Liu

Electronic Version Approved:

Ron Walcott
Interim Dean of the Graduate School
The University of Georgia
May 2020

Acknowledgments

I owe every achievement since entering the Statistics Ph.D. program, including being admitted to the program, to my friends and advisors Cheolwoo and Jeongyoun. Cheolwoo taught the first statistics class I took at UGA (and arguably the first I took ever, considering that I had forgotten essentially all of what I learned in the one introductory course I took as an undergrad). When I first mentioned to Cheolwoo my interest in applying to the Ph.D. program, his advice was "Maybe you should take one of our theory courses first." Jeongyoun taught that course and, as I was warned it would, it changed my impression of what getting a graduate degree in statistics would be like.

Although the Statistics program turned out to be a different kind of challenge than I expected, I am grateful to my advisors for supporting my admission into the program and giving me a chance to succeed. I realize it is unusual for someone to directly enter a doctoral program in statistics with little prior background in statistics, math, or computing, so I am indebted to them for recognizing my potential and advocating for me. I am also grateful for the support they've given me through the rest of the program, including small things like offering words of positive encouragement before an exam, and major things like becoming my advisors for my dissertation research. They are the type of advisors that take a deep personal interest in their students' achievements and success, and I am glad I have had the opportunity to benefit from the nurturing relationships they foster with their students.

I am also grateful to the rest of my committee, the other faculty in the Statistics Depart-

ment, and my fellow students for their support throughout the program. The first two years were especially tough, partly because I was still transitioning into a new field, but mostly because of the rigorous course requirements of the graduate program. Without encouragement from the faculty and mutual support among the students, it would have been easy to become overwhelmed.

I am thankful to my friends Rickamer Hoover and Sarah Sapp for being there to listen and provide support during the toughest times, and for tolerating me when I was under exceptional stress.

I received funding from various sources to support me throughout the graduate program. For that I am grateful, because I would not have been able to fund my own way through graduate school. The Graduate School awarded me an assistantship during the first two years of the program that undoubtedly eased my transition into the program. I was later supported through teaching assistantships and various grants, including National Science Foundation Grant No. NSF IIS-1607919. Not only did those funding opportunities support me through graduate school, but they also helped diversify my professional development.

Contents

Acknowledgments **iv**

1 Overview **1**

1.1 Low Dimensional Structure in Large-Scale Multivariate Problems 1

1.2 Statistical Methods Based on Low Rank Approximations 3

1.3 Regularization Techniques 9

1.4 Scope of Dissertation 13

2 A Low Rank, Orthogonal Tensor Regression Model **18**

2.1 Introduction 18

2.2 Proposed Methodology 35

2.3 Simulation Experiment 42

2.4 Real Data Analysis 76

2.5 Discussion 86

3 A Sparse, Low Rank Matrix Approximation with Application to Variable Selection in High-Dimensional Canonical Correlation Analysis **91**

3.1 Introduction 91

3.2 Proposed Methodology 96

3.3 Simulation Experiment 110

3.4 Real Data Analysis 121
3.5 Discussion 125

4 Sparse Canonical Correlation Analysis as a Regularized Least Squares Problem with Quadratic Equality Constraints 132

4.1 Introduction 132
4.2 Proposed Methodology 135
4.3 Simulation Experiments 149
4.4 Limitations and Future Directions 160

5 Conclusions and Future Directions 171

Bibliography 180

Appendix 189

Chapter 1

Overview

1.1 Low Dimensional Structure in Large-Scale Multivariate Problems

Modern research often involves collecting information from large numbers of variables, sometimes many more variables than observational units. An archetypal example is "omics" data in biological research. In genomics, researchers may collect information about gene expression for thousands of genes. In epigenomics, one may collect information about the methylation of DNA at thousands of CpG sites. In transcriptomics, one may measure the levels of one or several of the various types of RNAs produced by a cell. In proteomics, one may measure the production of many types of proteins. In every example, the genes, RNAs, proteins, etc. are the variables measured, which typically number in the thousands, while the number of tissue samples, subjects, or other examples of observational units typically number on the order of tens or hundreds. Moreover, many experiments involve collecting information from multiple omics sources at once. Thus, much of modern research is characterized by obtaining high dimensional, multivariate data from a relatively small number of samples.

Often the goal of research involving large numbers of variables is to describe the multivariate relationships among the variables, or between the variables and some response. For example, if one collects gene expressions from a set of tumor samples representing several different types of cancers, one may ask: Is a large proportion of the total variability in the gene expressions related to the fact that the samples come from different cancer tissues? Answering such a question involves describing the relationships among gene expressions across all of the genes. If a large proportion of the variability in the gene expressions is not due to there being different cancer types, it could still be the case that some combination of gene expressions could be useful for discriminating among tissues from different cancers. In another example, suppose one collects information about gene expression and DNA methylation from the same set of samples. One could ask: Is the level of DNA methylation related to the level of expression of certain genes? That question involves describing the relationships among variables between two different sets, rather than either set alone.

Describing the multivariate relationships among many variables can be framed as an exploratory analysis that tries to find and characterize lower dimensional structure in the relationships. To determine whether a large proportion of the total variability in gene expressions is related to the type of cancer from which the tissue was collected, we find the dominant modes of variation in the gene expression (e.g., the goal of principal components analysis), then examine whether the dominant modes of variation are associated with cancer type. To determine whether some combination of gene expressions could be useful for discriminating among tissues from different cancers, we try to find a low dimensional representation of the variables that best separate the cancer types (e.g., the goal of linear discriminant analysis). To determine whether DNA methylation levels are related to gene expression, we can attempt to characterize the dominant modes of co-variation between the two sets of variables; that is, we find a low dimensional representation of the DNA methylations and a low dimensional representation of the gene expressions that are strongly associated (e.g., the goal of canonical correlation analysis).

Problems that involve finding and characterizing low dimensional structure in multivariate data often have as their solutions a low rank approximation of some matrix. Principal components analysis, linear discriminant analysis, and canonical correlation analysis are just three examples of multivariate analyses that fall into the class of low rank matrix approximation problems. In Section 1.2, we describe why those analyses involve low rank matrix approximations, and also describe some other statistical methods that fall into the same class of problems.

1.2 Statistical Methods Based on Low Rank Approximations

Many multivariate statistical methods rely on matrix decompositions. Some methods explicitly assume a model in which a matrix decomposes as a product of two or more lower rank matrices, plus a residual. For example, in factor analysis, we assume the random variables $X_1, \ldots, X_p$ are actually composed of linear combinations of $k < p$ latent factors. If we obtain n observations of the variables and collect them in the matrix $X_{n\times p}$, then the factor analysis model assumes

$$
\begin{aligned}
&X = FL + E, \text{ where}\\
&F_{n\times k}: \text{ factor matrix with columns representing the latent factors}\\
&L_{k\times p}: \text{ loading matrix with rows giving the coefficients of linear combinations of the factors}\\
&E_{n\times p}: \text{ residual matrix.}
\end{aligned}
$$

Because the number of latent factors k is usually assumed to be small relative to the number of original variables p, we typically have $k = \text{rank}(FL) < \text{rank}(X)$. Thus, the decomposition FL is a low rank approximation of the data matrix X.

Similar models to the factor analysis model have been proposed when the data have certain properties, or we wish to restrict the decomposition to have certain properties, or both. For example, if all of the elements of X are non-negative, then we may wish all of the elements of F and L to be non-negative as well. The non-negative matrix factorization model assumes $X = FL + E$ such that $F \geq 0$ and $L \geq 0$, where E is again a residual matrix. As in factor analysis, the number of columns of F and rows of L are typically assumed to be small relative to p, so that the matrix FL is a low rank approximation of X.

If we think of factor analysis and non-negative matrix factorization as explicitly assuming models based on matrix decompositions, then we may think of other methods as implicitly relying on matrix decompositions. In principal components analysis (PCA), the objective is to construct linear combinations of the original variables $X_1, \ldots, X_p$ such that the new variables

$$\begin{gathered} u_{1,1}X_1 + \cdots + u_{1,p}X_p \\ \vdots \\ u_{p,1}X_1 + \cdots + u_{p,p}X_p \end{gathered}$$

have the largest variances possible while being uncorrelated. The coefficients of the linear combinations $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_p$ are called principal components (PCs) and the new variables are called PC variables. Let Σ denote the variance of $\boldsymbol{X} := (X_1, \ldots, X_p)^T$. To find the first PC, we solve

$$\boldsymbol{u}_1 = \arg\max_{\boldsymbol{u}} \ \boldsymbol{u}^T \Sigma \boldsymbol{u} \ \text{ s.t. } \ \boldsymbol{u}^T\boldsymbol{u} = 1,$$

since $Var(u_{1,1}X_1 + \cdots + u_{1,p}X_p) = Var(\boldsymbol{u}_1^T\boldsymbol{X}) = \boldsymbol{u}_1^T\Sigma\boldsymbol{u}_1$.

At first glance, PCA does not seem to involve a matrix decomposition or a low rank approximation. However, it can be shown that the solution to PCA (not only the first PC, but all others as well) is the spectral decomposition of the matrix Σ. That is, the PCs $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_p$ are the eigenvectors of Σ and the variances of the PC variables $Var(\boldsymbol{u}_1^T\boldsymbol{X}), \ldots, Var(\boldsymbol{u}_p^T\boldsymbol{X})$ are the eigenvalues $\lambda_1, \ldots, \lambda_p$. Because the solution to PCA is obtained from the spectral decomposition, it is equivalent to find the first PC by solving

$$\boldsymbol{u}_1 = \arg\min_{\boldsymbol{u}} \ \|\Sigma - \lambda \boldsymbol{u}\boldsymbol{u}^T\|_F \ \text{ s.t. } \ \lambda > 0, \ \boldsymbol{u}^T\boldsymbol{u} = 1.$$

As formulated above, it is clear that PCA involves finding a low rank approximation of the matrix Σ. In fact, the matrix $\lambda_1\boldsymbol{u}_1\boldsymbol{u}_1^T$ is the *best* rank-1 approximation of Σ in the sense of the Frobenius norm $\|M\|_F = \sqrt{\sum_{j,k} m_{jk}^2}$. More generally, if we collect the first k eigenvectors (i.e., the first k PCs) into the matrix U_k and the corresponding eigenvalues into the diagonal matrix Λ_k, then the matrix $U_k\Lambda_k U_k^T$ is the best rank-k approximation of Σ.

Fisher's linear discriminant analysis (LDA) can also be viewed as a statistical method implicitly relying on matrix decompositions. Fisher's LDA maximizes the between-to-within variance for data from k classes. Let $\{\boldsymbol{x}_i, Y_i\}$, $i = 1, \ldots, n$, denote pairs of observed covariates and class labels, where $\boldsymbol{x}_i \in \mathbb{R}^p$ and $Y_i \in \{1, \ldots, k\}$. Let $\hat{\boldsymbol{\mu}}_j$, $j = 1, \ldots, k$, denote the mean covariate vector for each class. Define the within and between covariance matrices as

$$\begin{aligned}
\Sigma_w &= \frac{1}{n-k}\sum_{j=1}^{k}\sum_{i=1}^{n_j}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_j)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_j)^T \\
\Sigma_b &= \sum_{j=1}^{k}\frac{n_j}{n}\hat{\boldsymbol{\mu}}_j\hat{\boldsymbol{\mu}}_j^T.
\end{aligned}$$

Then Fisher's LDA solves

$$\begin{aligned}
&\max_{\boldsymbol{u}_j} \ \frac{\boldsymbol{u}_j^T\Sigma_b\boldsymbol{u}_j}{\boldsymbol{u}_j^T\Sigma_w\boldsymbol{u}_j} \quad j = 1, \ldots, k-1 \\
\Longleftrightarrow \ &\max_{\boldsymbol{u}_j} \ \boldsymbol{u}_j^T\Sigma_b\boldsymbol{u}_j \ \text{ s.t. } \ \boldsymbol{u}_j^T\Sigma_w\boldsymbol{u}_{j'} = I(j = j'), \quad j = 1, \ldots, k-1,
\end{aligned}$$

where $I(\cdot)$ denotes the indicator function.

Using the change of variables $\tilde{\boldsymbol{u}} = \Sigma_w^{1/2}\boldsymbol{u}$, we can write the problem as

$$\max_{\tilde{\boldsymbol{u}}_j} \; \tilde{\boldsymbol{u}}_j^T \Sigma_w^{-1/2}\Sigma_b\Sigma_w^{-1/2}\tilde{\boldsymbol{u}}_j \;\; \text{s.t.} \;\; \tilde{\boldsymbol{u}}_j^T\tilde{\boldsymbol{u}}_{j'} = I(j = j'), \;\; j = 1, \dots, k-1.$$

Then the problem of LDA appears nearly identical to the problem of PCA. That is, instead of finding the spectral decomposition of Σ, we can obtain the solution to LDA by finding the spectral decomposition of $\Sigma_w^{-1/2}\Sigma_b\Sigma_w^{-1/2}$. As in PCA, the elements of the eigenvectors are the coefficients of linear combinations of the covariates, but in the case of LDA we call the eigenvectors the discriminant vectors or directions. Because the solution to LDA is based on the spectral decomposition, we could equivalently solve

$$\min_{U_{k-1},\Lambda_{k-1}} \; \|\Sigma_w^{-1/2}\Sigma_b\Sigma_w^{-1/2} - U_{k-1}\Lambda_{k-1}U_{k-1}^T\|_F \;\; \text{s.t.} \;\; \lambda_j > 0 \;\; \forall j \;\; \text{and} \;\; U_{k-1}^T U_{k-1} = I_{k-1}.$$

In that form, it is clear that the objective is to find a low rank approximation of $\Sigma_w^{-1/2}\Sigma_b\Sigma_w^{-1/2}$.

Another method closely related to PCA is canonical correlation analysis (CCA). Whereas PCA finds linear combinations of variables in one dataset that maximize the variance, CCA finds linear combinations of variables in two datasets that maximize the correlation between them. Suppose a dataset X consists of p variables and a dataset Y consists of q variables. Then, formally, CCA finds a linear combination $\boldsymbol{u} := (u_1, \dots, u_p)^T$ and a linear combination $\boldsymbol{v} := (v_1, \dots, v_q)^T$ such that the correlation $\rho := Cor(X\boldsymbol{u}, \; Y\boldsymbol{v})$ is maximized. The vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are called the canonical vectors and ρ is called the canonical correlation.

Let Σ_{XX} denote the variance of X, Σ_{YY} the variance of Y, and Σ_{XY} the covariance between X and Y. Then using the definition of correlation, CCA solves

$$\begin{aligned}
\boldsymbol{u}_1, \boldsymbol{v}_1 &= \underset{\boldsymbol{u},\boldsymbol{v}}{\arg\max}\ \frac{Cov(X\boldsymbol{u},\ Y\boldsymbol{v})}{\sqrt{Var(X\boldsymbol{u})}\sqrt{Var(Y\boldsymbol{v})}} \\
&= \underset{\boldsymbol{u},\boldsymbol{v}}{\arg\max}\ \frac{\boldsymbol{u}^T\Sigma_{XY}\boldsymbol{v}}{\sqrt{\boldsymbol{u}^T\Sigma_{XX}\boldsymbol{u}}\sqrt{\boldsymbol{v}^T\Sigma_{YY}\boldsymbol{v}}} \\
&= \underset{\boldsymbol{u},\boldsymbol{v}}{\arg\max}\ \boldsymbol{u}^T\Sigma_{XY}\boldsymbol{v} \ \text{ s.t. } \ \boldsymbol{u}^T\Sigma_{XX}\boldsymbol{u} = \boldsymbol{v}^T\Sigma_{YY}\boldsymbol{v} = 1,
\end{aligned}$$

where the constraint in the last line is imposed because the problem in the penultimate line is invariant to scaling. CCA can be used to find up to $\min(p, q)$ nonzero canonical correlations, but the focus of Chapters 3 and 4 is on finding the largest canonical correlation, so we limit the discussion to that problem.

As with PCA, it is not immediately apparent that CCA involves a low rank matrix approximation. Using the change of variables $\tilde{\boldsymbol{u}} = \Sigma_{XX}^{1/2}$ and $\tilde{\boldsymbol{v}} = \Sigma_{YY}^{1/2}$, the CCA problem can be rewriten as

$$\underset{\tilde{\boldsymbol{u}},\tilde{\boldsymbol{v}}}{\arg\max}\ \tilde{\boldsymbol{u}}^T\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1/2}\tilde{\boldsymbol{v}} \ \text{ s.t. } \ \tilde{\boldsymbol{u}}^T\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{v}}^T\tilde{\boldsymbol{v}} = 1.$$

It can be shown that the solution to the transformed problem can be found from the singular value decomposition (SVD) of the matrix $\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1/2}$. The canonical correlation ρ is the largest singular value and the canonical vectors can be found from the corresponding singular vectors. Because the solution to CCA is obtained from the SVD, the problem can be written equivalently as

$$\underset{\boldsymbol{u},\boldsymbol{v}}{\arg\min}\ \|\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1/2} - \rho\boldsymbol{u}\boldsymbol{v}^T\|_F \ \text{ s.t. } \ \rho > 0,\ \boldsymbol{u}^T\boldsymbol{u} = \boldsymbol{v}^T\boldsymbol{v} = 1.$$

Under its alternative formulation, it is clear that CCA involves finding a low rank approximation of the matrix $\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1/2}$. The matrix $\rho_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T$ is the best rank-1 approximation of $\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1/2}$ in the sense of the Frobenius norm.

The solutions to PCA, LDA, and CCA involve low rank approximations of matrices, so one may think of those methods as inherently relying on matrix decompositions. Matrix decompositions have also been used as tools in statistical methods that do not inherently rely on them. The focus of Chapter 2 is using images as covariates in generalized linear models (GLMs). The images we are concerned with are typically two- or three-dimensional arrays. We refer to 2D arrays as matrices and 3D or higher dimensional arrays as tensors. When we include a matrix- or tensor-valued covariate in a GLM, we must estimate a parameter of the same size. That is, we must estimate either a matrix- or tensor-valued parameter. Because the number of parameters increases multiplicatively with the number of dimensions, it is necessary to make some assumptions that reduce the number of parameters to estimate. We may assume the parameter matrix/tensor admits some particular matrix/tensor decomposition, then estimate the parameter under a low rank assumption. Then the estimate of the parameter matrix/tensor is a low rank approximation of the underlying population parameter matrix/tensor.

Let μ denote the mean of the response, $g(\cdot)$ the link function of a GLM, $\langle \cdot \rangle$ the inner product between vectorized versions of two matrices/tensors, and "$\circ$" the outer product of two or more vectors. Then the models we consider in Chapter 2 are of the form

$$g(\mu) = \alpha + \boldsymbol{\gamma}^T \boldsymbol{z}_i + \langle \mathcal{B},\ \mathcal{X}_i \rangle \ \text{ s.t. } \ \mathcal{B} = \sum_{r=1}^{k} \boldsymbol{\beta}_{1r} \circ \boldsymbol{\beta}_{2r} \circ \cdots \circ \boldsymbol{\beta}_{Dr},$$

where α is the scalar intercept, $\boldsymbol{z}_i$ is vector of regular covariates with corresponding parameters $\boldsymbol{\gamma}$, and $\mathcal{X}_i$ is a D-dimensional matrix- or tensor-valued covariate with corresponding parameter $\mathcal{B}$. The model assumes the parameter $\mathcal{B}$ can be represented as the sum of an outer product of vectors that is at most rank-k. The rank is usually chosen to be small, so that the estimated model parameter is a low rank approximation of $\mathcal{B}$. The number of parameters is additive, rather than multiplicative, in the number of dimensions, so substantially fewer observations are needed to fit the model.

A low rank assumption for the parameter matrix/tensor can be a strong assumption, and it may not be useful in many contexts. In Section 2.1, we describe the intuition behind why such a model may be reasonable for the types of images and applications with which we are concerned.

It is also not obvious which decomposition we may employ so that the parameter may be represented as the sum of an outer product of vectors that is at most rank-k. In the case of matrices, we may use the SVD. But for the model to generalize to higher order tensors, we must first define a notion of tensor rank, then define a decomposition that captures information about the rank. Neither the definition of matrix rank nor the matrix SVD generalizes directly to tensors. In both cases, there are analogs for tensors that share some, but not all, of the properties of the matrix rank and SVD. In Section 2.1.1, we give some background on multilinear algebra that describes different notions of tensor rank and some of the various tensor decompositions that have been proposed. We focus on the decompositions that allow us to construct a model such as above.

1.3 Regularization Techniques

The statistical methods discussed in Section 1.2 involve finding a low rank approximation to a matrix. In PCA, to find the first k PCs, one finds the best rank-k approximation of Σ. In CCA, to find the largest canonical correlation and corresponding canonical vectors, one finds the best rank-1 approximation of $\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1/2}$. In regression with a matrix- or tensor-valued parameter, one first assumes some rank k, then finds a rank-k estimate of the population parameter.

A common thread in all of the examples is that one must choose a fixed value k for the rank before finding a low rank approximation of the matrix or tensor. As an alternative, one may find a low rank approximation through regularization. For methods based on the SVD, such as CCA and regression with matrix-valued parameters, one can penalize the objective function with a sparsity-inducing function of the singular values. Because the rank of a matrix equals the number of nonzero singular values, that approach results in a low rank approximation by shrinking some of the matrix's singular values to exactly zero. Chapters 2 and 3 discuss that approach in the context of regression and CCA, respectively. Chapter 2 also discusses an extension of the approach for tensor-valued parameters. To extend the approach to tensors, it is first necessary to define a tensor decomposition for which there is analog of the singular value as in the matrix SVD.

A natural question is: What advantage would penalizing the singular values have over assuming a fixed value of the rank? For regression with matrix- or tensor-valued parameters, one purpose of the low rank assumption is to reduce the number of parameters to a feasible level. One could find the largest value k such that the number of parameters is smaller than the sample size. However, that strategy is ad hoc and could result in overfitting, since a smaller rank model might also provide an adequate fit to the data. A method based on penalizing the singular values allows one to find a low rank approximation through a data-driven choice of the tuning parameter. It also makes it possible to obtain estimates that are intermediate between rank k and $k + 1$ (see Section 2.3.3 for some examples). For CCA, the issue of choosing the rank is not as relevant. If one is only concerned with the largest canonical correlation, then the problem is by definition to find a rank-1 approximation.

Another advantage of a shrinkage-based approach is computational tractability. PCA, CCA, and regression with matrix- or tensor-valued covariates under a low (fixed) rank assumption are all non-convex problems. For PCA and CCA in the traditional $n > p$ setting, the non-convexity of the problem doesn't create any additional challenges because an analytical solution exists (namely, the SVD). However, in high dimensional settings, additional regularization is required for CCA (because the inverses of the sample versions of the matrices involved don't exist) and useful for PCA (e.g., to improve interpretation). Many popular strategies for adding regularization to the problem, such as sparsity-inducing constraints or penalties, require numerical solutions, in which case the non-convexity poses a challenge. The low rank matrix/tensor regression problem always requires a numerical solution, so its non-convexity is always a challenge. In contrast to the fixed-rank versions of the problems, versions based on penalizing the singular values can be constructed to be convex, depending on the choice of objective function and penalty function. Chapter 2 discusses the approach in the context of regression, and Chapter 3 discusses the approach in the context of CCA.

It is also possible that versions of the problems based on shrinkage reduce the effective number of parameters even further than the fixed-rank versions. For a normal linear model with a matrix-valued parameter, the effective number of parameters of a model that penalizes the ℓ_1 norm of the singular values is always dominated by the naive count of the number of parameters (Zhou and Li, 2014). That is, the effective number of parameters for the fixed-rank version of the problem is always larger than that of the shrinkage-based version (assuming the tuning parameter is chosen such that the shrinkage-based estimate has the same rank as the fixed-rank estimate). Whether that fact extends to other problems, such as CCA and regression with tensor-valued parameters, is unclear.

Another type of regularization that is useful for some of the statistical methods discussed is sparse regularization that results in variable selection. Variable selection is especially useful for PCA and CCA in high dimensional settings. For PCA, we may wish to find a linear combination of just a few variables that explain most of the variation in the data. For CCA, we may wish to select a few variables from each dataset such that the correlation between linear combinations of the subsets of variables is large. Variable selection is not only useful from a computational perspective, but also for improving the interpretation of the analysis. It is much easier to interpret the multivariate relationships among 10 variables than among 1,000 variables.

When we employ sparse regularization for variable selection, we make the implicit assumption that the underlying model is sparse; that is, the relationship among a small subset of the variables dominates the overall relationship among all of the variables. In the context of CCA, it means that the canonical correlation between a subset of variables from X and a subset from Y is nearly as large as the canonical correlation between all of the variables in X and Y. When we incorporate sparsity into a problem such as CCA, we hope to select the correct subset of variables that has the highest correlation possible among all subsets of that size. An exhaustive search is feasible only when the dimension is small, so we add a sparsity-inducing penalty function to the objective function instead.

In CCA, the canonical vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are the coefficients of the linear combinations of X and Y, respectively. If we wish to select a subset of variables from X and Y, then we need to estimate some of the coefficients of the linear combinations as exactly zero. Thus, one popular approach for sparse CCA is to penalize the objective function with a sparsity-inducing function of $\boldsymbol{u}$ and $\boldsymbol{v}$. With sparsity, we no longer have an analytical solution for CCA, so we must use a numerical approach. As mentioned previously, CCA is a non-convex problem, so sparse CCA is a challenging problem to solve even with the most sophisticated numerical algorithms. In addition, the non-convexity of the problem creates other difficulties. Unlike sparse regression problems based on the ℓ_1 norm [e.g., LASSO regression (Tibshirani, 1996)], the penalized and constrained versions of sparse CCA based on the ℓ_1 norms of $\boldsymbol{u}$ and $\boldsymbol{v}$ are not equivalent (Gaynanova et al., 2017). Importantly, the version of sparse CCA that constrains the ℓ_1 norms of $\boldsymbol{u}$ and $\boldsymbol{v}$ can achieve any level of sparsity, while the version that penalizes the ℓ_1 norms of $\boldsymbol{u}$ and $\boldsymbol{v}$ cannot.

1.4 Scope of Dissertation

The main goal of Chapter 2 is to develop a tensor regression model based on shrinking the singular values of the tensor. The model can be viewed as an extension of Zhou and Li's (2014) regularized matrix model, which penalizes the singular values of a matrix-valued parameter. The main challenge in extending Zhou and Li's model to tensors is how to define a "singular value" for tensors. Multiple definitions for higher-order analogs of the matrix SVD have been proposed. None of the proposals share every property of the matrix SVD, but all share some properties. We focus on a decomposition that enforces the higher-order analogs of the singular vectors to be orthogonal and the rank of the tensor to equal the number of nonzero singular values. Chen and Saad (2009) characterized some of the properties of the decomposition, which they called a low rank, orthogonal approximation to tensors (LROAT), and proposed an algorithm to compute the decomposition. We focus on the LROAT decomposition because the rank of the tensor is guaranteed to equal the number of nonzero singular values; if we penalize the singular values so that some shrink to zero, then we can find a low rank approximation of the parameter tensor just as we can for matrices.

The guarantee that the rank equals the number of nonzero singular values is not without cost. By employing the LROAT decomposition in our tensor regression model, we make the additional assumption that the parameter tensor is orthogonally-decomposable. Not all tensors admit an orthogonal decomposition, so Chen and Saad's algorithm finds the closest approximation to the tensor when it is not orthogonally-decomposable. Because orthogonally-decomposable is a strong assumption, in almost all cases the estimate obtained from our model will be an approximation of the parameter tensor in more than one sense. It will be a low rank approximation of the parameter tensor whenever the assumed rank is smaller than the true rank, which is our main goal. However, it will also be an approximation whenever the true population parameter tensor is not orthogonally-decomposable. Importantly, that implies the estimator of the parameter tensor cannot be consistent (unless we make the unrealistic assumption that the population parameter is also orthogonally-decomposable). That is, no matter how much data we collect, we can never make the estimation error arbitrarily small. Section 2.1.1 shows some examples of low rank tensors and their closest orthogonal approximation to give some idea of how much accuracy we sacrifice by using the LROAT decomposition. Section 2.3 explores the issue in the context of the proposed tensor regression model through a simulation experiment. Section 2.4 applies the model to a real dataset. Both the simulation experiment and real data analysis suggest the benefits of the proposed model outweigh its limitations, given the limitations of the best available alternatives. Although our model makes stricter assumptions, the algorithm we propose to fit the model appears to perform better than the algorithms proposed to fit alternative models.

Chapter 3 considers the problem of describing the dominant modes of co-variation between two datasets while simultaneously performing variable selection. The dominant modes of co-variation between two datasets can be found through a low rank approximation of the matrix containing all of the pairwise relationships between variables. We focus on the estimated covariance or correlation matrix. We develop a method that penalizes both the singular values of the matrix and the row or column norms. The penalty on the singular values encourages a low rank representation of the matrix, while the penalty on the row or column norms results in variable selection from one of the datasets. The problem is especially challenging to solve numerically because both the number of nonzero singular values and the number of nonzero rows or columns influence the rank.

One of the main contributions of Chapter 3 is to show that the proposed approach aims to select the same subsets of variables as certain state-of-the-art approaches for sparse CCA. Section 3.2.2 discusses the theoretical assumptions necessary to make such a claim. Section 3.2.4 introduces an alternative to tuning parameter selection that ranks variables according to their importance. In simulation, the average rank of the non-sparse variables can serve as a proxy for the variable selection accuracy. In practice, ranking the variables can be viewed as a kind of continuous variable selection. It can also alleviate some of the instability associated with trying to select a single model. Section 3.3 demonstrates through simulation that the proposed approach and sparse CCA not only rank the non-sparse variables highly, but also that the proposed approach performs better in that respect in many instances. Section 3.4 provides some empirical justification for our claims through a real data analysis. That the proposed approach and sparse CCA aim to select the same variables requires assumptions that are not likely to hold for real data. Despite that, we demonstrate that the proposed approach and sparse CCA share a high degree of overlap with respect to which variables they rank highly. In addition, if one performs non-sparse CCA with a few of the most highly ranked variables, the estimated canonical correlation is as high or higher using the variables selected by the proposed approach than the variables selected by sparse CCA. That fact is rather surprising given that the proposed approach does not attempt to maximize the canonical correlation.

Chapter 4 further explores regularization techniques for CCA. Whereas the methods described in Chapters 2 and 3 utilize a penalty on the singular values to achieve a low rank approximation, the methods introduced in Chapter 4 utilize the more traditional fixed-rank approximation for CCA. The methods in Chapter 4 explore some alternative approaches to achieve variable selection through sparse estimation. One of the main goals of Chapter 4 is to relax some of the assumptions that have been imposed in other sparse CCA proposals. For example, the sparse CCA methods of Witten et al. (2009) and Safo et al. (2018) standardize the data and assume the within-set covariance matrices Σ_{XX} and Σ_{YY} are identity, thus avoiding the need to estimate their inverses and allowing the method to depend on the estimate of Σ_{XY} alone. Assuming Σ_{XX} and Σ_{YY} are identity amounts to assuming the variables within each dataset are uncorrelated, which will almost never be realistic in practice.

To avoid making the assumption that Σ_{XX} and Σ_{YY} are identity, the methods in Chapter

4 make use of an alternative formulation of CCA as a multivariate regression problem (Izenman, 1975). Formulated as a multivariate regression, the canonical vectors corresponding to the largest canonical correlation can be obtained by solving a least squares criterion. The formulation is especially useful in the high dimensional setting because we can make use of all of the tools that have been developed for high dimensional regression. In particular, our interest is in applying the tools developed for sparse estimation. The formulation is not only amenable to adding a LASSO penalty (Tibshirani, 1996), as has been done in other sparse CCA proposals, but also other sparsity-inducing penalties, such as SCAD (Fan and Li, 2001) and MC (Zhang, 2010) penalty, which are known for their improved variable selection properties.

Although CCA can be formulated as a penalized least squares regression problem, it is much more challenging to solve than regression. Unlike a standard regression problem, CCA remains a non-convex optimization problem even when it is reformulated to look like a regression problem. Whereas a standard regression problem is convex in the parameters, CCA is bi-convex in the parameters. In addition, CCA involves quadratic equality constraints, which are non-convex. As a consequence, one of the algorithms we propose to solve the sparse CCA problem (the proximal gradient method described in Section 4.2.2) tends to be sensitive to the initial value. The other algorithm we propose (the ADMM algorithm described in Section 4.2.2) makes use of a solver designed to handle quadratic equality constraints, and so is less sensitive to the initial value, but only performs well when the dimension is small. The solver does not scale well with the dimension, so the computation time is prohibitive for high dimensional problems. In addition, neither algorithm performs well relative to sparse CCA methods that assume Σ_{XX} and Σ_{YY} are identity, suggesting that the cost of relaxing that assumption exceeds the benefit. Section 4.4 discusses the major limitations of the approach, as well as some ideas for future directions. Although it is unlikely that the approach could outperform the best sparse CCA methods in high dimensional settings, it is competitive in low dimensional settings. The approach also has other advantages, such as a natural way to extend CCA to more than two datasets or to grouped data.

Chapter 5 provides some concluding remarks and ideas for future research that incorporate regularization into statistical methods that are based on or utilize low rank matrix/tensor approximations.

Chapter 2

A Low Rank, Orthogonal Tensor Regression Model

2.1 Introduction

Modern technology allows researchers to collect data that can be represented as images. For example, images such as those obtained by structural magnetic resonance imaging (MRI), diffusion tensor imaging (DTI), electroencephalography (EEG), and functional MRI (fMRI) are frequently used by psychologists, neuroscientists, and others within health and medical disciplines to diagnosis or study disease. The need to analyze such imaging data poses new challenges for statisticians because many traditional statistical methods do not readily accommodate imaging data. Different approaches have been proposed depending on the type of question the researcher would like to answer using the imaging data. Our work considers research questions in which the image may be considered a covariate, and one would like to know how the image covariate relates to or affects a response variable (which may be discrete or continuous). Some examples of questions of interest include whether an image obtained via fMRI can be used to predict whether a person would be diagnosed with a particular psychiatric disorder (such as schizophrenia), or whether a structural MRI can be used to predict disease status (e.g., Alzheimer's). In either example, the response may be binary (1=disorder/disease, 0=healthy) or continuous (e.g., symptom scale scores). Determining whether an image can be used to make such a prediction and what elements of the image are important to the prediction can help researchers identify the neural circuitry involved in the disorder/disease and potentially contribute to a better understanding of its etiology.

From the statistical perspective, images can be treated as matrix-valued (2D) or array-valued (3D or higher) data. However, images are challenging to model using traditional methods because of the implicit spatial structure and large number of parameters to estimate. For example, if one wanted to model the relationship between disease status and a fMRI covariate using a generalized linear model (GLM), one would need to estimate $40 \times 48 \times 38 = 72,960$ parameters – the dimension of a typical image obtained by fMRI (and this is ignoring that fMRI data are usually collected over many time points). One solution is to fit a regularized GLM with a sparsity-inducing penalty, which effectively reduces the number of parameters to estimate. However, the results from penalized regression are often unsatisfactory when the data are ultra-high dimensional. Moreover, the penalized regression approach does not account for the spatial structure inherent in imaging data.

Several researchers have proposed tensor regression models to handle array-valued imaging data. "Tensor" simply refers to a multidimensional array (e.g., a 1D tensor is a vector, a 2D tensor is a matrix, etc.). Tensor regression models exploit the expected spatial dependency of the array-valued image by assuming the corresponding tensor of model parameters can be well-approximated by a low rank structure, thereby accounting for the spatial dependency while simultaneously reducing the number of parameters to estimate. To illustrate the intuition behind this idea, we consider a 2D image as an example. Notions of rank and decomposability of matrices do not extend directly to 3D and higher dimensional arrays, so we defer a general discussion of the low rank tensor model until after a brief introduction to multilinear algebra in Section 2.1.1.

Consider a $p_1 \times p_2$ image consisting of intensities for $p_1 p_2$ pixels. For example, the values might be the blood oxygen-level dependent (BOLD) signal for a slice of a brain image acquired by fMRI. If we obtain such images from two groups of subjects, say healthy controls and subjects diagnosed with a psychological disorder, then we might expect the images within each group to be more similar on average than images between groups. Moreover, if there are features that distinguish images belonging to healthy subjects from images belonging to subjects diagnosed with a disorder, we might expect the features to be few and to occur in spatially compact locations. Thus, an image consisting just of the features that distinguish one group of images from the other might be well-approximated by a low rank matrix. To place the example in the context of a GLM, the group status would represent the response, the fMRI images would represent the covariate, and the image of features that represent areas of the brain where the BOLD signal differs between the groups on average would represent the matrix of model parameters. Then, the tensor regression model assumes that the matrix of model parameters can be well-approximated by low rank matrix. Note that the model *does not* assume the images themselves are low rank. Also note that the model does not explicitly account for spatial dependency. It implicitly accounts for spatial dependency through the posited mechanism by which a low rank structure for the model parameters might arise. The extent to which a low rank assumption might hold in practice, and the usefulness of such a model in general, depends on the context and scientific question of interest. As the example demonstrates, a low rank tensor regression model may be reasonable for some experiments that collect fMRI data.

Two main approaches have been proposed to impose a low rank structure on the model's parameter tensor. One approach assumes a fixed rank *a priori*. However, the true rank of the parameter tensor will generally not be known, and it may not be clear how to choose the rank suitably. As the rank increases, the fit of the model should improve (assuming that the sample size is sufficient to fit higher rank models). Thus, the question is: What is the lowest rank such that the model fit is adequate (or does not improve appreciably for higher ranks)? Answering such a question is non-trivial and usually will require the subjective judgment of the analyst or an ad hoc approach.

To avoid fixing the rank *a priori*, another approach penalizes the ℓ_1 norm of the singular values of a matrix. That approach can be applied directly when the images are 2D because the parameter tensor is itself a matrix. When the images are 3D or higher, the parameter tensor must first be converted into a matrix (see "matricization" in Section 2.1.1), and various ways to do this have been proposed. Since the number of nonzero singular values equals the rank, such a model effectively reduces the rank of the matrix by shrinking some of its singular values to zero. Then the rank of the matrix is determined in a data-driven way through selection of the tuning parameter (e.g., by cross-validation, AIC, BIC, etc.).

Methods that penalize the singular values rely on the singular value decomposition (SVD) of a matrix. As a consequence, these methods cannot be applied directly to 3D or higher dimensional images because the matrix SVD does not have a direct analog for higher dimensional tensors. Although previous authors have overcome this limitation by converting the tensor into a matrix, an alternative is to use a tensor decomposition that has some key similarities with the matrix SVD. Several tensor decompositions have been proposed that share some, but not all, of the properties of the matrix SVD. Here, we extend methods that penalize singular values to higher dimensional tensors through a low rank, orthogonal approximation of tensors (LROAT) (Chen and Saad, 2009). In Section 2.1.1, we give some background on multilinear algebra, describe LROAT and several other tensor decompositions, and interpret the decompositions as higher-order analogs of the matrix SVD. In Section 2.1.2, we describe the some of the matrix/tensor regression models that have previously been proposed. In Section 2.2, we define the proposed LROAT regression model. We first present an algorithm to fit the model for fixed rank. We then extend the method to work by shrinking the "singular values," which can be viewed as an extension of methods for matrices. In Section 2.3, we conduct a simulation experiment to compare the proposed methods to state-of-the-art methods. In Section 2.4, we apply the proposed methods for visual stimulus decoding using a real fMRI dataset. We provide some concluding remarks in Section 2.5.

2.1.1 Background: Multilinear Algebra

[Note: The background information in this section follows Kolda and Bader (2009); see their review for a more extensive background on multilinear algebra.] Multilinear algebra extends concepts in linear algebra for vectors and matrices to multidimensional arrays called tensors. The number of dimensions of a tensor is called the *order* or the number of *modes*. Thus, a matrix is a second-order tensor or a two-mode tensor. If we fix the indices of all modes but one, we get vectors along that mode, which we call *fibers*. For example, the columns of a matrix are its mode-1 fibers and the rows are its mode-2 fibers. If we fix the indices of all modes but two, we get matrices, which we call *slices*.

A D-mode tensor may be written as the sum of an outer product of D vectors. For example, the SVD of a $m \times n$ matrix A can be written as

$$A = U\Sigma V^T = \sum_{r=1}^{R} \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^T = \sum_{r=1}^{R} \sigma_r \boldsymbol{u}_r \circ \boldsymbol{v}_r \quad \text{s.t.} \quad \begin{aligned}[t] & \boldsymbol{u} \in \mathbb{R}^m, \boldsymbol{v} \in \mathbb{R}^n, \\ & \operatorname{rank}(A) = R, \\ & \sigma_r > 0, \ \text{and} \\ & U^T U = V^T V = I_R, \end{aligned}$$

where "$\circ$" denotes the vector outer product. Note that the outer product of two vectors is a rank-1 matrix, and the outer product of D vectors is a rank-1 tensor.

A general $p_1 \times p_2 \times \cdots \times p_D$ tensor can be written as

$$\mathcal{X} = \sum_{r=1}^{R} \boldsymbol{\beta}_{1r} \circ \boldsymbol{\beta}_{2r} \circ \cdots \circ \boldsymbol{\beta}_{Dr}, \quad \boldsymbol{\beta}_{1r} \in \mathbb{R}^{p_1}, \boldsymbol{\beta}_{2r} \in \mathbb{R}^{p_2}, \ldots, \boldsymbol{\beta}_{Dr} \in \mathbb{R}^{p_D}, \quad r = 1, \ldots, R.$$

Here, R denotes the *tensor rank*, defined as the minimal number of rank-1 tensors needed such that their sum reconstructs the original tensor exactly. Although this definition of rank applies to both matrices and higher-order tensors, other properties of the matrix rank do not extend to higher-order tensors. For example, $\text{rank}(A) = dim(\mathcal{C}(A)) = dim(\mathcal{C}(A^T))$; that is, the rank is equal to the dimension of the column space is equal to the dimension of the row space. However, if one were to arrange the mode-n fibers of a tensor $\mathcal{X}$ into a matrix, the dimension of the column space of the resulting matrix need not be the same as the tensor rank. Arrangement of the mode-n fibers of a tensor into a matrix is called the mode-n *matricization* of $\mathcal{X}$, written $X_{(n)}$, and the rank of the mode-n matricization is called the mode-n rank. The mode-n ranks of a tensor need not all be the same, nor do any need to equal the tensor rank, although they are all bounded above by the tensor rank. Since the mode-n ranks are not necessarily the same, they are written as the tuple $(R_1, R_2, \ldots, R_D)$, called the *multilinear rank*.

Tensors can be multiplied with matrices or vectors of appropriate size, called the n-mode product. The multiplication of a $p_1 \times p_2 \times \cdots \times p_D$ tensor $\mathcal{X}$ with a $J \times p_d$ matrix U results in a tensor of size $p_1 \times p_2 \times \cdots \times p_{d-1} \times J \times p_{d+1} \times \cdots \times p_D$. The n-mode product can be expressed as

$$\mathcal{X} \times_n U \iff UX_{(n)}.$$

We may also take the inner product of two tensors of the same size. The inner product of two tensors $\mathcal{X}$ and $\mathcal{Y}$ simply multiplies the corresponding elements and sums everything up.

This idea is equivalently expressed as

$$\langle \mathcal{X}, \mathcal{Y} \rangle \iff \boldsymbol{x}^T \boldsymbol{y}, \text{ where } \boldsymbol{x} := vec(\mathcal{X}),\ \boldsymbol{y} := vec(\mathcal{Y}).$$

The squared Frobenius norm of a tensor $\mathcal{X}$, denoted $||\mathcal{X}||_F^2$, is defined as $\langle \mathcal{X}, \mathcal{X} \rangle$.

Using these concepts and notations, we can discuss tensor decompositions. For notational convenience, we restrict the discussion to tensors of order three. The results extend in a straightforward manner to general tensors of higher order. The *canonical polyadic decomposition* (CPD) (Hitchcock, 1927) of a tensor expresses the tensor as a sum of rank-1 tensors, which we have already seen. That is, the CPD of a rank-R tensor can be written as

$$\mathcal{X} = \sum_{r=1}^{R} \boldsymbol{\beta}_{1r} \circ \boldsymbol{\beta}_{2r} \circ \boldsymbol{\beta}_{3r}, \quad \boldsymbol{\beta}_{1r} \in \mathbb{R}^{p_1}, \boldsymbol{\beta}_{2r} \in \mathbb{R}^{p_2}, \boldsymbol{\beta}_{3r} \in \mathbb{R}^{p_3}, \quad r = 1, \ldots, R.$$

We may also collect the vectors $\boldsymbol{\beta}_{1r}, \boldsymbol{\beta}_{2r}, \boldsymbol{\beta}_{3r},\ r = 1, \ldots, R$ into factor matrices $B_1 : p_1 \times R$, $B_2 : p_2 \times R$, and $B_3 : p_3 \times R$ and write $\mathcal{X} = [[B_1, B_2, B_3]]$. Note that unlike the matrix SVD, the CPD imposes no constraints on the factor matrices, such as orthogonality. Even without additional constraints, the CPD is often unique up to scaling and permutation. Kruskal (1977, 1989) derived sufficient conditions for the uniqueness of the CPD for general $p_1 \times p_2 \times p_3$ tensors. First, the k-rank of a matrix B_1, denoted k_{B_1}, is defined as the maximum value k_{B_1} such that any k_{B_1} columns are linearly independent. Then Kruskal's results state that the CPD is unique if $k_{B_1} + k_{B_2} + k_{B_3} \geq 2R + 2$. Sidiropoulos and Bro (2000) derived an analogous result for tensors of higher order.

In general, the rank of a higher-order tensor will not be known. Then for fixed $k \leq R$, the CPD can be used to find a rank-k approximation of the tensor. The rank-k approximation of a tensor solves

$$\min_{\hat{\mathcal{X}}} ||\mathcal{X} - \hat{\mathcal{X}}||_F \text{ s.t. } \hat{\mathcal{X}} = \sum_{r=1}^{k} \boldsymbol{\beta}_{1r} \circ \boldsymbol{\beta}_{2r} \circ \boldsymbol{\beta}_{3r} = [[B_1, B_2, B_3]]. \tag{2.1}$$

The problem (2.1) is typically solved by an alternating least squares (ALS) algorithm. ALS computes the CPD by fixing B_2, B_3 and solving (2.1) for B_1, then fixing B_1, B_3 and solving for B_2, then fixing B_1, B_2 and solving for B_3. The procedure is repeated until convergence (the change in $||\mathcal{X} - \hat{\mathcal{X}}||$ between successive iterations falls below some threshold). The matrices B_1, B_2, B_3 need to be initialized, and the initial values can be arbitrary (e.g., random initialization). Unlike matrices, higher-order tensors do not necessarily admit a best rank-k approximation (i.e., there is no version of the Eckart and Young (1936) theorem for higher-order tensors). Consequently, one may not apply the ALS algorithm sequentially to obtain low rank approximations for an increasing sequence of ranks $k_1 < k_2 < \cdots \leq R$ by using the factors found for lower ranks when solving for the additional factors of higher ranks.

Whereas the CPD is most strongly associated with the notion of tensor rank, the *Tucker decomposition* (TKD) is more strongly associated with the notion of multilinear rank. Tucker (1966) proposed to decompose a tensor as the multilinear product of a core tensor (same shape as the original tensor, but smaller size) and factor matrices corresponding to each mode. In fact, the CPD is a special case of the TKD, in which the core tensor is constrained to be diagonal. The TKD of a tensor $\mathcal{X}$ is

$$\mathcal{X} = \mathcal{G} \times_1 B_1 \times_2 B_2 \times_3 B_3 = [[\mathcal{G};\ B_1, B_2, B_3]].$$

Note that the factor matrices B_1, B_2, B_3 depend on the core tensor $\mathcal{G}$, and so are not the same as the factor matrices of the CPD in general. The core tensor in the TKD is of minimal size if its dimensions are the same as the multilinear rank of $\mathcal{X}$.

There are several variants of TKD, differing primarily in the constraints imposed on the factor matrices. The most common constraint is that the factor matrices be orthonormal. For orthonormal B_1, B_2, B_3, De Lathauwer et al. (2000a) argued that the TKD shares many properties analogous to the matrix SVD, so they called this version of TKD the higher-order singular value decomposition (HOSVD). For a rank-(R_1, R_2, R_3) tensor, the HOSVD can be computed using Algorithm 1. By construction, the HOSVD is an exact decomposition and can be applied to any tensor (see Proposition 1 in the Appendix).

Algorithm 1: Higher-Order Singular Value Decomposition (HOSVD)

```
Input: Tensor $\mathcal{X}$, multilinear rank $(R_1, R_2, R_3)$.
Output: $\mathcal{G}, B_1, B_2, B_3$ s.t. $B_1^T B_1 = I_{R_1}, B_2^T B_2 = I_{R_2}, B_3^T B_3 = I_{R_3}$.

for $n = 1, 2, 3$ do
    $B_n \longleftarrow$ left singular vectors corresponding to $R_n$ largest singular values from
      $SVD(X_{(n)})$
end

$\mathcal{G} = \mathcal{X} \times_1 B_1^T \times_2 B_2^T \times_3 B_3^T$
```

The HOSVD algorithm can also be used to obtain an approximation of the original tensor by letting some or all of the R_n's to be less than the mode-n ranks. This version is known as the truncated HOSVD. Like for the tensor rank, there is no version of the Eckart and Young (1936) theorem for the multilinear rank. Consequently, the truncated HOSVD is not necessarily optimal in terms of the Frobenius norm of the difference between the true tensor and the truncated decomposition. In light of this fact, given (R_1, R_2, R_3) s.t. R_n is less than the actual mode-n rank for some n, one can instead solve

$$\min_{\mathcal{G}, B_1, B_2, B_3} ||\mathcal{X} - \mathcal{G} \times_1 B_1 \times_2 B_2 \times_3 B_3||_F \tag{2.2}$$

subject to $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ and $B_n \in \mathbb{R}^{p_n \times R_n}$ column-wise orthonormal, $n = 1, 2, 3$.

As with the CPD, an ALS algorithm can be used to solve the problem (2.2). One can use the HOSVD algorithm to supply the initial values. De Lathauwer et al. (2000b) gave an efficient method for solving problem (2.2), which they called the higher-order orthogonal iteration (HOOI); however, their algorithm is not guaranteed to converge to a global optimum.

In real data applications, tensor decompositions are often used to obtain a low rank, interpretable approximation of a data tensor. With respect to obtaining a decomposition with low tensor rank, the CPD has an advantage over the TKD because the TKD fixes the multilinear rank, which does not have a direct relationship to the tensor rank. However, with respect to obtaining a decomposition with interpretable factor matrices, the TKD has an advantage over the CPD because the factor matrices in the CPD are completely unconstrained. In the TKD, the columns of the factor matrices may be constrained to be orthonormal, non-negative, linearly independent, etc. to suite the context of the application.

Chen and Saad (2009) proposed a decomposition that yields a low tensor rank and constrains the factor matrices to be orthonormal, combining the strengths of the CPD and TKD. Their decomposition, called low rank orthogonal approximation to tensors (LROAT), takes the form

$$\mathcal{X} = \sum_{r=1}^{R} \sigma_r \boldsymbol{\beta}_{1r} \circ \boldsymbol{\beta}_{2r} \circ \boldsymbol{\beta}_{3r} = [[diag(\sigma_1, \dots, \sigma_R);\ B_1, B_2, B_3]] \text{ s.t. } B_1^T B_1 = B_2^T B_2 = B_3^T B_3 = I_R.$$

The LROAT decomposition can be considered a special case of the CPD, since it takes the same form as the CPD but imposes additional constraints. Chen and Saad (2009) argued that the LROAT decomposition also shares many properties with the matrix SVD, and so LROAT can be thought of as a higher-order generalization of the matrix SVD. Like the HOSVD, the factor matrices in the LROAT decomposition are column-wise orthonormal. However, the LROAT decomposition also shares additional properties with the matrix SVD that are not properties of the HOSVD. For a third-order orthogonally-decomposable tensor $\mathcal{X}$, the LROAT decomposition has the following additional properties:

1. The tensor rank is bounded by the smallest size among the modes:
   $R = \text{rank}(\mathcal{X}) \leq \min(p_1, p_2, p_3)$.

2. The "singular values" are non-negative and ordered: $0 \leq \sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_R$.

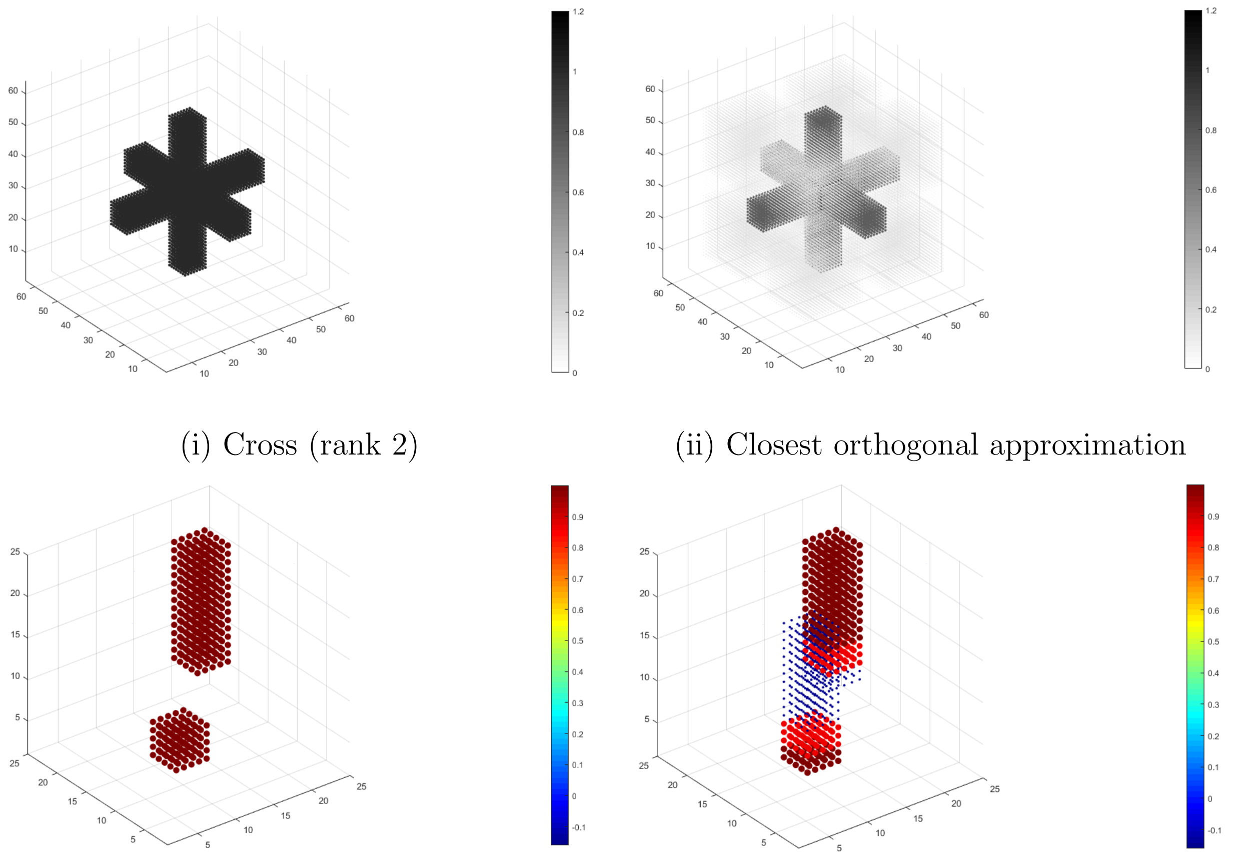
3. The tensor rank equals the number of nonzero singular values:
   $R = \#\{\sigma_r : \sigma_r > 0,\ r = 1, \ldots, \min(p_1, p_2, p_3)\}$.

4. The decomposition is unique up to signs (even when some singular values are repeated).

The key difference between LROAT for higher-order tensors and the matrix SVD is that not all tensors are orthogonally-decomposable. From property 1, a tensor with $\text{rank}(\mathcal{X}) > \min(p_1, p_2, p_3)$ does not admit an orthogonal decomposition. Moreover, even if $\text{rank}(\mathcal{X}) \leq \min(p_1, p_2, p_3)$, there is no guarantee that $\mathcal{X}$ is orthogonally-decomposable. Thus, in real data applications, the LROAT decomposition is nearly always an approximation of the original tensor (hence "approximation" is part of the acronym). In contrast, the CPD and TKD (theoretically) provide exact decompositions for R and (R_1, R_2, R_3) large enough, respectively. Despite this shortcoming of the LROAT decomposition, Chen and Saad (2009) proved that the approximation is optimal in the sense that the problem

$$\min_{\sigma_1, \ldots, \sigma_R, B_1, B_2, B_3} \left\| \mathcal{X} - \sum_{r=1}^{R} \sigma_r \boldsymbol{\beta}_{1r} \circ \boldsymbol{\beta}_{2r} \circ \boldsymbol{\beta}_{3r} \right\|_F \quad \text{s.t. } B_1^T B_1 = B_2^T B_2 = B_3^T B_3 = I_R \tag{2.3}$$

has a solution for any $\mathcal{X} \in \mathbb{R}^{p_1 \times p_2 \times p_3}$ and $R \leq \min(p_1, p_2, p_3)$. Some examples of 3D tensors that do not admit an orthogonal decomposition and their closest orthogonally-decomposable approximations are shown in Figure 2.1.



(i) Cross (rank 2)

(ii) Closest orthogonal approximation

(iii) Hyper-rectangles (rank 2)

(iv) Closest orthogonal approximation

Figure 2.1: Examples of non-orthogonally-decomposable 3D tensors and their closest orthogonal approximations.

Chen and Saad (2009) proposed an alternating algorithm to solve problem (2.3). Their algorithm is given in Algorithm 2 for a Dth-order tensor. Note that the polar decomposition step implies that their algorithm is a more general block coordinate descent algorithm rather than an ALS algorithm.

Algorithm 2: Low Rank Orthogonal Approximation of Tensors (LROAT)

```
Input: Tensor $\mathcal{X}$, tensor rank $R$, orthonormal initial guesses $B_1, \ldots, B_D$ (typically
       obtained by HOSVD with $R_1 = \cdots = R_D = R$).
Output: $\sigma_1, \ldots, \sigma_R$, final estimates $B_1, \ldots, B_D$ s.t. $B_1^T B_1 = \cdots = B_D^T B_D = I_R$.

while objective criterion not met do
    for $d = 1, \ldots, D$ do
        Compute $V_d = (\boldsymbol{v}_{d1}, \ldots, \boldsymbol{v}_{dR})$, where
        $\boldsymbol{v}_{dr} = \mathcal{X} \times_1 \boldsymbol{\beta}_{1,r}^T \times \cdots \times_{d-1} \boldsymbol{\beta}_{d-1,r}^T \times_{d+1} \boldsymbol{\beta}_{d+1,r}^T \times \cdots \times_D \boldsymbol{\beta}_{D,r}^T \in \mathbb{R}^{p_d}$

        Compute $\Sigma = diag(\sigma_1, \ldots, \sigma_R)$, where
        $\sigma_r = \mathcal{X} \times_1 \boldsymbol{\beta}_{1,r}^T \times \cdots \times_D \boldsymbol{\beta}_{D,r}^T = \langle \boldsymbol{\beta}_{dr}, \boldsymbol{v}_{dr} \rangle$

        $(Q_d, H_d) \leftarrow$ polar-decomp$(V_d \Sigma)$

        Update $B_d \leftarrow Q_d$
    end
end
```

2.1.2 Matrix and Tensor Regression Models

Tensor decompositions have a long history of applications in the fields of psychometrics, chemometrics, and signal processing (Kolda and Bader, 2009). In these applications, the objective is often to approximate an observed tensor by one of lower rank. The CPD and TKD described in Section 2.1.1 are commonly employed for this purpose. In contrast, for applications in statistics and machine learning, the tensor to approximate by one of lower rank is usually not observed. In a tensor regression model, the parameter tensor is assumed to admit a low rank structure. That is, the problem is to simultaneously estimate the parameter tensor and approximate it by a tensor of lower rank. Thus, the problem of estimating and decomposing an unobserved quantity is of a fundamentally different nature than finding an approximation to an observed quantity.

Various types of tensor regression models have been considered. For classification problems, Hung and Wang (2012) and Tan et al. (2013) studied logistic regression with a matrix and tensor covariate, respectively. Hung and Wang (2012) formulated the systematic part of the model as a bilinear form in the parameters, which is equivalent to a rank-1 CPD. Tan et al. (2013) considered the more general case of tensors of arbitrary order and a rank-R CPD. Signoretto et al. (2014) also studied classification problems using a logistic loss, but assumed the TKD for the parameter tensor. They penalized the spectral norm of the mode-n matricizations to achieve low (multilinear) rank structure.

Hoff (2015) and Lock (2018) considered models in which both the response and covariate are tensors (not necessarily of the same size). Hoff (2015) used the TKD for the parameter tensor, while Lock (2018) used the CPD with an additional ridge penalty for the elements of the factor matrices in the CPD. When both the response and covariate are tensors, the number of parameters to estimate grows quickly, so the ridge penalty on the elements of the factor matrices ensures that the parameters can be estimated even when the sample size is small relative to the number of parameters.

Several authors have considered the case of scalar or multivariate response and tensor covariate. Guo et al. (2012) studied linear regression for continuous responses and support vector machine for binary responses. In both cases, they used CPD with an additional ridge penalty or group sparsity penalty. Guhaniyogi et al. (2017) constructed a Gaussian linear regression model under a Bayesian framework by specifying priors for the factor matrices in the CPD.

Others have approached the case of scalar response and tensor covariate from the framework of GLMs. A GLM with tensor covariate can be written as

$$
\begin{aligned}
&g(\mu) = \alpha + \boldsymbol{\gamma}^T \boldsymbol{z}_i + \langle \mathcal{B},\ \mathcal{X}_i \rangle \\
\text{where } &g(\mu):\ \text{a link function for the mean of } Y_i \\
&\alpha:\ \text{a constant parameter} \\
&\boldsymbol{z}_i:\ \text{a vector of regular covariates} \\
&\boldsymbol{\gamma}:\ \text{a vector of parameters corresponding to the regular covariates} \\
&\mathcal{X}_i:\ \text{a tensor covariate (usually an image) of order } D \text{ and size } p_1 \times \cdots \times p_D \\
&\mathcal{B}:\ \text{a tensor of parameters corresponding to the tensor covariate}
\end{aligned}
$$

Zhou et al. (2013) assumed a rank-R CPD for the parameter tensor. The systematic component of their model can be expressed as

$$
\begin{aligned}
g(\mu) &= \alpha + \boldsymbol{\gamma}^T \boldsymbol{z}_i + \langle \sum_{r=1}^{R} \boldsymbol{\beta}_{1r} \circ \boldsymbol{\beta}_{2r} \circ \cdots \circ \boldsymbol{\beta}_{Dr},\ \mathcal{X}_i \rangle \qquad (2.4) \\
&= \alpha + \boldsymbol{\gamma}^T \boldsymbol{z}_i + \langle [[B_1, B_2, \ldots, B_D]],\ \mathcal{X}_i \rangle.
\end{aligned}
$$

Assuming a fixed rank for the parameter tensor can be interpreted in the strict sense – that the true parameter tensor is rank R. Alternatively, the assumption may be interpreted more loosely – that the true parameter tensor is well-approximated by a rank-R CPD. The rank-R assumption reduces the number of parameters associated with the tensor covariate from $\prod_{d=1}^{D} p_d$ to $R \sum_{d=1}^{D} p_d$. For the fMRI example from Section 2.1, a rank-1 assumption would reduce the number of parameters from $40 \times 48 \times 38 = 72,960$ to $40 + 48 + 38 = 126$.

Zhou et al. (2013) proposed to fit the tensor regression model (2.4) by a block coordinate descent algorithm, similar in principle to the ALS algorithm used for the CPD. The problem is naturally suited to an alternating algorithm because (2.4) is nonlinear problem in $\mathcal{B}$, but reduces to separate classical GLM problems in B_d for fixed $(B_1, \ldots, B_{d-1}, B_{d+1}, \ldots, B_D)$. However, the model inherits the nonuniqueness problems associated with the CPD and the algorithm is not guaranteed to converge to a global optimum. Zhou et al. (2013) imposed some additional constraints to deal with nonuniqueness and suggested initializing the algorithm from many random starts to obtain a solution that achieves a good objective value. Note that Zhou et al. (2013) also introduced a penalized version of model (2.4) by adding a penalty to the loss as a function of the elements of the $\boldsymbol{\beta}_{dr}$'s. Solving a penalized version of the problem may be desirable to improve the interpretability of the solution or further reduce the effective number of parameters to estimate (since $R\sum_{d=1}^{D} p_d$ can still be quite large relative to the sample size). Li et al. (2016) extended the penalized version of model (2.4) to the case of a multivariate (i.e., vector-valued) response.

Recently, Li et al. (2018) and Chen et al. (2019) proposed GLMs with tensor covariates that make use of the TKD. The systematic components of their models can be written as

$$g(\mu) = \alpha + \boldsymbol{\gamma}^T \boldsymbol{z}_i + \langle [[\mathcal{G};\ B_1, B_2, \ldots, B_D]],\ \mathcal{X}_i \rangle, \tag{2.5}$$

which is similar to model (2.4) except that the core tensor $\mathcal{G}$ is not constrained to be diagonal and it assumes a fixed multilinear rank rather than a fixed tensor rank. Although the TKD is more flexible than the CPD in some respects, the improved flexibility comes at the cost of additional parameters to estimate. Thus, the model (2.5) may be difficult to fit unless the dimensionality is reduced or the sample size is large. In addition, since the model (2.5) assumes a fixed multilinear rank $(R_1, R_2, \ldots, R_D)$, it compounds the problem of choosing an appropriate rank that was encountered for the model (2.4). Chen et al. (2019) overcame those difficulties to some extent through various assumptions about the multilinear rank (e.g., the maximum of the mode-n ranks is bounded).

Because *a priori* specification of the tensor rank or multilinear rank may be difficult in practice, Zhou and Li (2014) proposed a convex relaxation of the rank for matrices by penalizing the singular values, which they call regularized matrix regression. Their approach does not require fixing the rank in advance, but instead shrinks some of the singular values of the parameter matrix to zero, thereby encouraging a low rank representation of the parameter matrix. The objective function for their model can be written as

$$\min_{B} \; -\ell\ell(B) + P_\lambda(\sigma(B)), \tag{2.6}$$

where B is the parameter matrix, $-\ell\ell(\cdot)$ is the negative log-likelihood of a GLM, $\sigma(\cdot)$ extracts the singular values of a matrix, and $P_\lambda(\cdot)$ is a sparsity-inducing penalty function. For example, if $P_\lambda(\cdot)$ is a LASSO penalty (Tibshirani, 1996), then $P_\lambda(\sigma(B))$ is a function of the nuclear norm of B, $||B||_* = \sum_{r=1}^{R} \sigma_r$. The systematic component of the GLM is

$$g(\mu) = \alpha + \boldsymbol{\gamma}^T \boldsymbol{z}_i + \langle B,\ X_i \rangle,$$

which is the same as in models (2.4) and (2.5) except that there are no restrictions on the parameter matrix B.

Zhou and Li (2014) provide an algorithm based on the proximal gradient method for solving the problem (2.6). The algorithm splits the problem into two steps: one that minimizes the negative log-likelihood by a gradient descent step, and one that applies the proximal operator associated with $P_\lambda(\sigma(B))$. The penalty functions they consider all result in proximal operators that threshold the singular values, driving some to exactly zero. Note that the penalty effectively reduces the number of parameters to estimate by encouraging a low rank representation of B. However, unlike the models (2.4) and (2.5) that use a fixed rank, the effective degrees of freedom must be estimated. Zhou and Li (2014) showed that the effective degrees of freedom of the regularized matrix regression model is always dominated by the naive count of the number of parameters. As a consequence, their model can be fit even when $\min(p_1, p_2)$ exceeds the sample size.

2.2 Proposed Methodology

In this section, we develop a low rank, orthogonal tensor regression model that makes use of the LROAT decomposition, which we call the LROAT regression model. We first specify the model for fixed tensor rank and propose a projected gradient descent algorithm to fit the model. Such a model is similar to the model proposed by Zhou et al. (2013), albeit more restrictive. We then formulate a version of the problem that avoids fixing the tensor rank by penalizing the objective function instead. The penalty is applied to the "singular values" of the LROAT decomposition, which encourages a low rank representation of the parameter tensor by shrinking some of the singular values to exactly zero. We propose an algorithm based on the proximal gradient method to solve the problem. This version of the problem can be viewed as an extension of the model proposed by Zhou and Li (2014) to higher-order tensors. However, a key difference between the two problems is that matrices always admit an orthogonal decomposition (by the SVD), while higher-order tensors are not guaranteed to be orthogonally decomposable.

Under the GLM framework, the LROAT regression model for fixed rank can be specified as

$$\begin{aligned} g(\mu) &= \alpha + \boldsymbol{\gamma}^T \boldsymbol{z}_i + \langle \mathcal{B},\ \mathcal{X}_i \rangle \text{ s.t. } \operatorname{rank}(\mathcal{B}) = R \text{ and orthogonally decomposable} \\ &= \alpha + \boldsymbol{\gamma}^T \boldsymbol{z}_i + \langle \sum_{r=1}^{R} \sigma_r \boldsymbol{\beta}_{1r} \circ \boldsymbol{\beta}_{2r} \circ \cdots \circ \boldsymbol{\beta}_{Dr},\ \mathcal{X}_i \rangle \\ &\quad \text{s.t. } \boldsymbol{\beta}_{dr}^T \boldsymbol{\beta}_{dr'} = I(r = r'),\ d = 1, \ldots, D,\ r, r' \in \{1, \ldots, R\} \\ &= \alpha + \boldsymbol{\gamma}^T \boldsymbol{z}_i + \langle [[diag(\sigma_1, \ldots, \sigma_R);\ B_1, B_2, \ldots, B_D]],\ \mathcal{X}_i \rangle \qquad (2.7) \\ &\quad \text{s.t. } B_1^T B_1 = \cdots = B_D^T B_D = I_R, \end{aligned}$$

where $I(\cdot)$ denotes the indicator function. The model (2.7) is very similar to the model (2.4), with the only difference being that the factor matrices are restricted to be orthogonal. In fact, when $D = 2$ (i.e., the tensors are matrices), the two models are essentially equivalent in that the estimated parameters $\widehat{\mathcal{B}}$ should be equal. However, the estimated factor matrices $\widehat{B_1}, \widehat{B_2}$ will not be equal because of the orthogonality constraint. The interpretation is that the models (2.4) and (2.7) are estimating the same parameter $\mathcal{B}$, but the model (2.4) allows any basis for the column space of $\mathcal{B}$, while the model (2.7) requires an orthogonal basis for the column space of $\mathcal{B}$. When $D \geq 3$, the models are not equivalent because the parameter space of model (2.4) is all rank-R tensors of size $p_1 \times \cdots \times p_D$, while the parameter space of model (2.7) is all rank-R orthogonally-decomposable tensors of size $p_1 \times \cdots \times p_D$.

Because of the orthogonality constraints, the alternating minimization algorithm of Zhou et al. (2013) cannot be used to fit model (2.7), even when $D = 2$. We propose a projected gradient descent algorithm to fit model (2.7). For a general problem min $f(x)$ s.t. $x \in \mathcal{S}$, where $\mathcal{S}$ denotes a set of constraints, the updates of the projected gradient descent algorithm take the form

$$x^{(k+1)} = \prod\nolimits_{\mathcal{S}} \left(x^k - \delta^k \nabla f(x^k) \right),$$

where δ^k is a step size and $\prod_{\mathcal{S}}$ denotes the projection operator onto the feasible set $\mathcal{S}$. The projection operator can be defined as $\prod_{\mathcal{S}}(x) = \arg\min_{z\in\mathcal{S}} \ ||x - z||_F$. The projected gradient descent method can be interpreted as the usual gradient descent method with the additional modification that each update is required to lie in the feasible set. The projection step ensures that each update lies in the feasible set while minimizing the distance between the update and the update that would be obtained from an unconstrained version of the problem.

The constraint set in model (2.7) is the space of rank-R orthogonally-decomposable tensors. To project onto this set, recall problem (2.3). Problem (2.3) is identical to the problem defined by projecting onto the space of rank-R orthogonally-decomposable tensors. This suggests that Algorithm 2 may be used for the projection step in a projected gradient descent algorithm for fitting model (2.7). The gradient descent step may be based on a suitable loss function, such as the negative log-likelihood. The complete details of the proposed algorithm for fitting model (2.7) are given in Algorithm 3.

Algorithm 3: Projected Gradient Descent for LROAT Regression Model

```
Input: Response $Y_{N\times 1}$, regular covariates $Z_{N\times p_0}$, tensor covariate $\mathcal{X}_{p_1\times\cdots\times p_D\times N}$,
       initial guesses for parameters $\alpha^{(0)}, \boldsymbol{\gamma}^{(0)}, \mathcal{B}^{(0)}$, assumed rank $R$, and
       distribution (normal or Bernoulli, though any distribution in the
       exponential family in theory).
Output: Final estimates $\hat{\alpha}, \hat{\boldsymbol{\gamma}}, \widehat{\mathcal{B}} = [[diag(\hat{\sigma}_1, \ldots, \hat{\sigma}_R);\ \widehat{B}_1, \ldots, \widehat{B}_D]]$.

// Force initial guess to be rank-R and orthogonally-decomposable.
$[B_1 \ldots, B_D] \leftarrow \text{hosvd}(\mathcal{B}^{(0)}, R_1 = \cdots = R_D = R)$ // Calls Algorithm 1
$\mathcal{B}^{(0)} \leftarrow \text{lroat}(\mathcal{B}^{(0)}, R, B_1 \ldots, B_D)$ // Calls Algorithm 2

$k \leftarrow 0$ // iteration counter
$\delta^0 \leftarrow 1$ // initialize step size

while objective criterion not met do

    // gradient descent step
    $vec(\alpha^{(temp)}, \boldsymbol{\gamma}^{(temp)}, \widetilde{\mathcal{B}}^{(temp)}) \leftarrow vec(\alpha^{(k)}, \boldsymbol{\gamma}^{(k)}, \mathcal{B}^{(k)}) - \delta^k \nabla f(\alpha^{(k)}, \boldsymbol{\gamma}^{(k)}, \mathcal{B}^{(k)})$
    // function $f$ is $-\ell\ell$

    // projection step
    $[B_1 \ldots, B_D] \leftarrow \text{hosvd}(\widetilde{\mathcal{B}}^{(temp)}, R_1 = \cdots = R_D = R)$ // Calls Algorithm 1
    $\mathcal{B}^{(temp)} \leftarrow \text{lroat}(\widetilde{\mathcal{B}}^{(temp)}, R, B_1 \ldots, B_D)$ // Calls Algorithm 2

    if $f(\alpha^{(temp)}, \boldsymbol{\gamma}^{(temp)}, \mathcal{B}^{(temp)}) < f(\alpha^{(k)}, \boldsymbol{\gamma}^{(k)}, \mathcal{B}^{(k)})$ then
        $[\alpha^{(k+1)}, \boldsymbol{\gamma}^{(k+1)}, \mathcal{B}^{(k+1)}] \leftarrow [\alpha^{(temp)}, \boldsymbol{\gamma}^{(temp)}, \mathcal{B}^{(temp)}]$ // accept update
        $\delta^{k+1} \leftarrow 1.2 * \delta^k$ // increase step size
    else
        // reject update; perform line search for step size
        $[\alpha^{(k+1)}, \boldsymbol{\gamma}^{(k+1)}, \mathcal{B}^{(k+1)}] \leftarrow [\alpha^{(k)}, \boldsymbol{\gamma}^{(k)}, \mathcal{B}^{(k)}]$
        $\delta^{k+1} \leftarrow 0.5 * \delta^k$ // shrink step size
    end

    $k \leftarrow k + 1$ // update iteration counter

end

$[\hat{\alpha}, \hat{\boldsymbol{\gamma}}, \widehat{\mathcal{B}}] \leftarrow [\alpha^{(final)}, \boldsymbol{\gamma}^{(final)}, \mathcal{B}^{(final)}]$
```

To modify model (2.7) to avoid *a priori* specification of the rank, we penalize the objective function associated with the model as in Zhou and Li (2014). The penalized version of the problem can be written as

$$\min_{\mathcal{B}} \quad -\ell\ell(\mathcal{B}) + P_\lambda(\sigma(\mathcal{B})), \tag{2.8}$$

where $\mathcal{B}$ is the parameter tensor, $-\ell\ell(\cdot)$ is the negative log-likelihood of a GLM, $\sigma(\cdot)$ extracts the singular values of an orthogonally-decomposable tensor, and $P_\lambda(\cdot)$ is a sparsity-inducing penalty function. For this study, we only consider the LASSO penalty. Note that for $D = 2$, the problem (2.8) is identical to the problem (2.6) considered by Zhou and Li (2014). Thus, the solution simply involves shrinking the singular values of $\mathcal{B}$. However, for $D \geq 3$, the solution involves first projecting onto the set of orthogonally-decomposable tensors, then shrinking the singular values. That is, problem (2.8) has different interpretations depending on D. For $D = 2$, the problem constrains the magnitudes of the singular values, but places no restriction on the parameter space of matrices. For $D \geq 3$, the problem constrains both the parameter space of tensors and the magnitudes of the singular values.

To solve problem (2.8), we propose an algorithm based on the proximal gradient method similar to the algorithm proposed by Zhou and Li (2014). Their algorithm improves on the standard proximal gradient method by incorporating an extrapolation step due to Nesterov (1983) that achieves an accelerated convergence rate. In addition, we modify Zhou and Li's algorithm to account for the fact that tensors of order $D \geq 3$ must be projected onto the set of orthogonally-decomposable tensors before shrinking the singular values. We propose to include a step that projects onto the set of rank-$\min(p_1, \ldots, p_D)$ orthogonally-decomposable tensors – the largest rank possible for the LROAT decomposition of a tensor of size $p_1 \times \cdots \times p_D$. When the actual tensor rank $R < \min(p_1, \ldots, p_D)$, this does not pose a problem because we will have $\sigma_{R+1} = \cdots = \sigma_{\min(p_1,\ldots,p_D)} = 0$.

Note that although we say the proposed algorithm is based on the proximal gradient method (and name it as such), it is more appropriate to say that the algorithm is inspired by or modeled after the proximal gradient method. The step that projects onto the set of orthogonally-decomposable tensors is not part of the standard proximal gradient method or any of its variants. Furthermore, even if the tensor resulting from the gradient descent step was already orthogonally decomposable (and hence no projection was necessary), the algorithm could still not accurately be called a proximal algorithm. To be called a proximal algorithm, it would require a concept or definition of a proximal operator for the sum of the singular values of an orthogonally-decomposable tensor. As far as we know, no such proximal operator has been defined. For matrices, the proximal operator of the sum of the singular values (i.e., the nuclear norm) is the soft thresholding operator, which shrinks each of the singular values as $\max(0, \sigma_r - \lambda)$ for some fixed constant λ, but does not alter the factor matrices (i.e., the singular vectors). Thus, for orthogonally-decomposable tensors, we apply the soft thresholding operator to the tensor's singular values in an analogous way.

The full details of the proposed algorithm are given in Algorithm 4. Note that although the algorithm initializes all of the parameters as zero, the parameters may be initialized as any value. In particular, it may be desirable to warm start the algorithm from previous estimates when fitting the model for a sequence of values of the tuning parameter.

Algorithm 4: Proximal Gradient Method for LROAT Regression Model

```
Input: Response $Y_{N \times 1}$, regular covariates $Z_{N \times p_0}$, tensor covariate $\mathcal{X}_{p_1 \times \cdots \times p_D \times N}$, tuning parameter $\lambda$, and
       distribution (normal, Bernoulli, or another distribution in the exponential family).
Output: Final estimates $\hat{\alpha}, \hat{\boldsymbol{\gamma}}, \widehat{\mathcal{B}} = [[diag(\hat{\sigma}_1(\lambda), \ldots, \hat{\sigma}_{\min(p_1,\ldots,p_D)}(\lambda));\ \widehat{B}_1, \ldots, \widehat{B}_D]]$.

// Note:  $\boldsymbol{\theta} := vec(\alpha, \boldsymbol{\gamma}, \mathcal{B})$
// initial guesses all zero
$\alpha^{(0)} = \alpha^{(1)} \leftarrow 0$; $\boldsymbol{\gamma}^{(0)} = \boldsymbol{\gamma}^{(1)} \leftarrow \mathbf{0}_{p_0}$; $\mathcal{B}^{(0)} = \mathcal{B}^{(1)} \leftarrow 0_{p_1 \times \cdots \times p_D}$

$k \leftarrow 0$ // iteration counter
$\delta^0 \leftarrow 1$ // initialize step size
$\eta^0 \leftarrow 0$;  $\eta^1 \leftarrow 1$ // initialize extrapolation parameters

while objective criterion not met do
    // extrapolation step
    $\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k)} + \frac{\eta^{k-1}-1}{\eta^k}(\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^{(k-1)})$

    while not descended do
        // gradient descent step
        $vec(\alpha^{(temp)}, \boldsymbol{\gamma}^{(temp)}, \widetilde{\mathcal{A}}) \leftarrow \boldsymbol{\theta}^{(k)} - \delta^k \nabla f(\boldsymbol{\theta}^{(k)})$
        // function $f$ is $-\ell\ell$
        // $\widetilde{\mathcal{A}}$ is part of an intermediate update for $\mathcal{B}^{(k)}$

        // projection step
        $[A_1 \ldots, A_D] \leftarrow \text{hosvd}(\widetilde{\mathcal{A}}, R_1 = \cdots = R_D = \min(p_1, \ldots, p_D))$ // Calls Algorithm 1
        $\mathcal{A} \leftarrow \text{lroat}(\widetilde{\mathcal{A}}, R = \min(p_1, \ldots, p_D), A_1 \ldots, A_D)$ // Calls Algorithm 2

        // thresholding step
        for $r = 1, \ldots, \min(p_1, \ldots, p_D)$ do
            $\sigma_r^{(temp)} \leftarrow \max(0,\ a_r - \delta^k \lambda)$  // $a_r$'s are singular values of $\mathcal{A}$
        end

        // update $\mathcal{B}^{(k)}$ with thresholded singular values and orthogonal factor matrices of $\mathcal{A}$
        $\mathcal{B}^{(temp)} \leftarrow [[diag(\sigma_1^{(temp)}, \ldots, \sigma_{\min(p_1,\ldots,p_D)}^{(temp)});\ A_1 \ldots, A_D]]$

        // check if descent
        if $f(\boldsymbol{\theta}^{(temp)}) < f(\boldsymbol{\theta}^{(k)}) + \nabla f(\boldsymbol{\theta}^{(k)})^T(\boldsymbol{\theta}^{(temp)} - \boldsymbol{\theta}^{(k)}) + \frac{1}{2\delta^k}||\boldsymbol{\theta}^{(temp)} - \boldsymbol{\theta}^{(k)}||_2^2$ then
            $[\alpha^{(k+1)}, \boldsymbol{\gamma}^{(k+1)}, \mathcal{B}^{(k+1)}] \leftarrow [\alpha^{(temp)}, \boldsymbol{\gamma}^{(temp)}, \mathcal{B}^{(temp)}]$ // accept update
        else
            // reject update; perform line search for step size
            $\delta^k \leftarrow 0.5 * \delta^k$ // shrink step size
        end
    end

    $\eta^{k+1} \leftarrow 0.5 * \left(1 + \sqrt{1 + (2\eta^k)^2}\right)$// update extrapolation parameter
    $k \leftarrow k + 1$ // update iteration counter
end

$[\hat{\alpha}, \hat{\boldsymbol{\gamma}}, \widehat{\mathcal{B}}] \leftarrow [\alpha^{(final)}, \boldsymbol{\gamma}^{(final)}, \mathcal{B}^{(final)}]$
```

2.3 Simulation Experiment

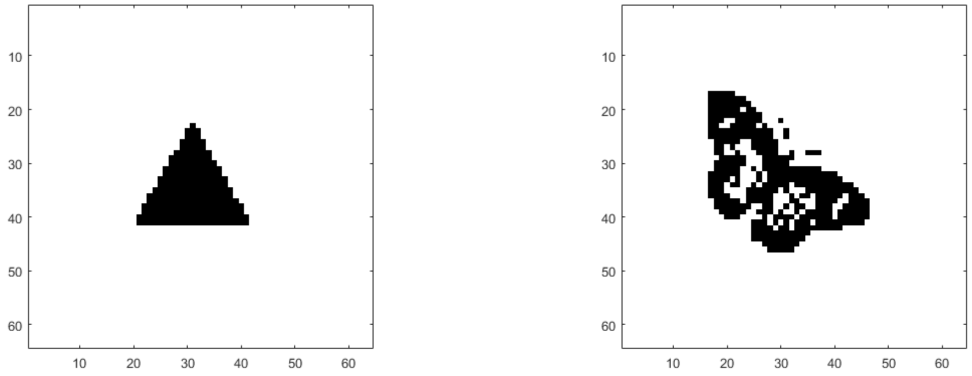
We compare the LROAT regression models and our methods for estimating them to the methods of Zhou et al. (2013) through three simulation experiments. The methods of Zhou et al. (2013) and Zhou and Li (2014) are implemented in the MATLAB TensorReg Toolbox (Zhou, 2017). We refer to model (2.7) for fixed rank simply as LROAT, the regularized version formulated in problem (2.8) as regLROAT, and Zhou et al.'s model (2.4) as CPD (since it is based on the canonical polyadic decomposition). In Experiment 1, we compare the fixed-rank LROAT model to the CPD model for a 2D covariate. For 2D covariates, models (2.7) and (2.4) are equivalent, differing only in their parametrizations and estimation algorithms. Thus, our main purpose is to compare the performance of our proposed Algorithm 3 to the alternating minimization algorithm of Zhou et al. (2013) with respect to number of iterations until convergence, computation time, and objective value achieved by the solution. In Experiment 2, we compare the fixed-rank LROAT model to the CPD model for a 3D covariate. For 3D covariates, the models (2.7) and (2.4) are not equivalent. We expect the LROAT model to perform poorly relative to the CPD model when the true parameter tensor does not admit an orthogonal decomposition, but there may be an advantage to the LROAT model when the true parameter tensor is orthogonally decomposable. In Experiment 3, we compare the fixed-rank LROAT and CPD models to the regLROAT model for a 3D covariate. Our purpose here is to determine whether soft-thresholding the singular values yields a better low rank solution than enforcing a low rank solution by fixing the rank (which may be interpreted as hard-thresholding the singular values). We expect the relative performances of the three methods to depend on the orthogonal-decomposability of the true parameter tensor. Additional details for each experiment follow.

2.3.1 Experiment 1

We generate data under model (2.7) for several scenarios. Note that, for 2D covariates, generating data under model (2.7) is equivalent to generating data under model (2.4). For the purposes of simulation, we omit a regular covariate from the model. The scenarios are:

1. $Y_i \sim N(\beta_0 + < \mathcal{X}_i, \mathcal{B} >,\ \sigma^2),\ (\mathcal{X}_i)_{jk} \sim N(0,\ 1),\ \mathcal{B}$: a 64×64 image of a triangle (rank 13), sample size: 2000
   (a) $\sigma^2 = 1$.
   (b) $\sigma^2 = 100$.
2. $Y_i \sim N(\beta_0 + < \mathcal{X}_i, \mathcal{B} >,\ 1),\ (\mathcal{X}_i)_{jk} \sim N(0,\ 1),\ \mathcal{B}$: a 64×64 image of a butterfly (rank 28), sample size: 2500
3. $Y_i \sim Bernoulli\left(p = \frac{\exp(\beta_0 + <\mathcal{X}_i, \mathcal{B}>)}{1+\exp(\beta_0 + <\mathcal{X}_i, \mathcal{B}>)}\right),\ (\mathcal{X}_i)_{jk} \sim N(0,\ 1),\ \mathcal{B}$: a 64×64 image of a triangle (rank 13), sample size: 3000
4. Real fMRI images.

For each scenario, we fit the LROAT and CPD models for a sequence of ranks. Note that the "images" in the scenarios are actually white noise, but the true parameter tensor is an image. Generating the "images" as white noise is simply for convenience, and using an image for the true parameter tensor makes it easy to control the rank of the true parameter tensor and to evaluate the performance of the methods visually (with respect to how well the estimated tensor matches the true tensor). Plots of the triangle and butterfly images used for the true parameter tensors are shown in Figure 2.2.



(i) Triangle: rank 13

(ii) Butterfly: rank 28

Figure 2.2: Images used for true parameter tensors in Scenarios 1–3.

Because Scenarios $1-3$ are not representative of what one might encounter with real data, we also wanted to simulate data for which the covariates are actually images. For that purpose, we used slices from a fMRI activation map dataset in Scenario 4. For a fixed slice number, we sampled (with replacement) images obtained from 35 healthy subjects 1000 times. So that none of the images were exactly the same, we added a small amount of noise to each sampled image. For half of the sampled images, we created a small "signal region" in the image by adding some value to a 6×6 region in the image. The 6×6 region was designed to appear (on average) as a 2×2 square nested inside a 6×6 square by generating the inner values as $N(0.60, 2(0.03)^2)$ and the outer values as $N(0.30, 0.03^2)$. The response was generated as $Y \sim N(0,1)$ for images without the signal and $Y \sim N(12,1)$ for images with the signal. The intuition behind this data-generating scheme is that the subjects come from two groups with activation maps that differ on average in a small region, and the differences in activation are associated with some response. Note that the signal is not uniform in magnitude over the region (the inner square has larger signal than the outer square, on average), and subjects with the signal express it to a different extent (the values making up the signal region are generated randomly for each subject). In addition, the relationship between the response and activation map is not defined in an exact way. Thus, the data cannot be viewed as coming a from a model in the usual sense. However, the data might be viewed as approximately coming from model (2.7), where the elements of the true parameter tensor take value zero except for a 6×6 region with the 2×2 inner part taking value two and the outer part taking value one (so that $\operatorname{rank}(\mathcal{B}) = 2$); a plot of the tensor is shown in Figure 2.3. Since the data are not generated from an exact model, this simulation scenario should pose a more challenging estimation problem with respect to fitting the LROAT and CPD models.

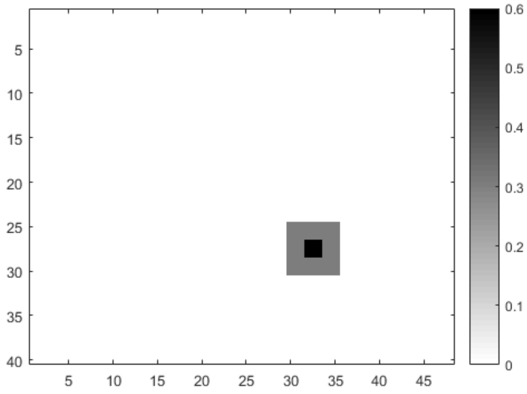


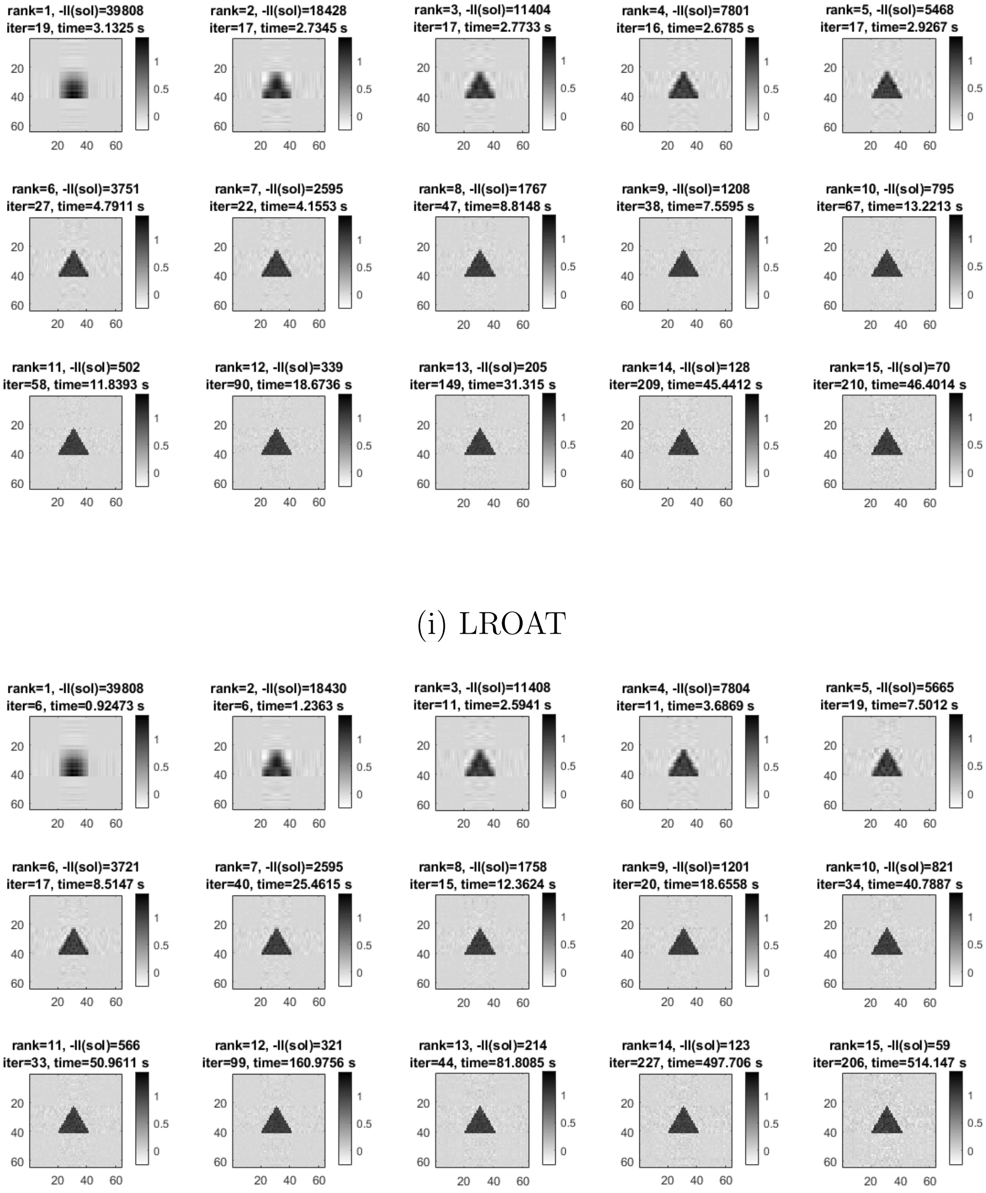
Figure 2.3: Average signal (rank 2) added to fMRI slices in Scenario 4.

It is desirable to evaluate the estimation accuracy of the proposed Algorithm 3 vs. the alternating minimization algorithm of Zhou et al. (2013). It is also of interest to compare the estimation accuracies of the fixed rank models (i.e., LROAT and CPD) to that of the regularized matrix model (Zhou and Li, 2014). However, fitting the models is computationally intensive even for one dataset, so it is impractical to repeat all of Scenarios 1-4 many times. Since Scenario 4 most closely resembles what one might encounter with real data, we repeat Scenario 4 100 times and measure estimation accuracy in terms of the mean squared error (MSE). We calculate MSE as MSE$= \frac{1}{100}\sum_{i=1}^{100}\langle \widehat{\mathcal{B}}_i - \mathcal{B},\ \widehat{\mathcal{B}}_i - \mathcal{B}\rangle$, where $\mathcal{B}$ represents the average difference in the activation map between the two groups of subjects and $\widehat{\mathcal{B}}$ is the estimate obtained under the LROAT, CPD, or regularized matrix model. Since the activation map slices were sampled only from healthy subjects, one would not expect differences in activation on average except for the region where signal was added for one group. That is, we take $\mathcal{B}$ to be the rank-2 tensor consisting of elements taking value zero except for the 6×6 region in which the 2×2 inner part takes value 0.6 and the outer part takes value 0.3.

Results - Experiment 1, Scenario 1

The LROAT and CPD models were fit for ranks 1–15. Plots of the estimates under each model for noise levels $\sigma^2 = 1$ and $\sigma^2 = 10$ are shown in Figures 2.4 and 2.5, respectively. The higher noise level poses a more challenging estimation problem for both methods, as expected. Neither method is able to estimate a clear image of the true signal regardless of the rank when $\sigma^2 = 10$. In contrast, both methods are able to estimate a clear image of the true signal when $\sigma^2 = 1$ as long as the rank ≥ 7 (roughly).

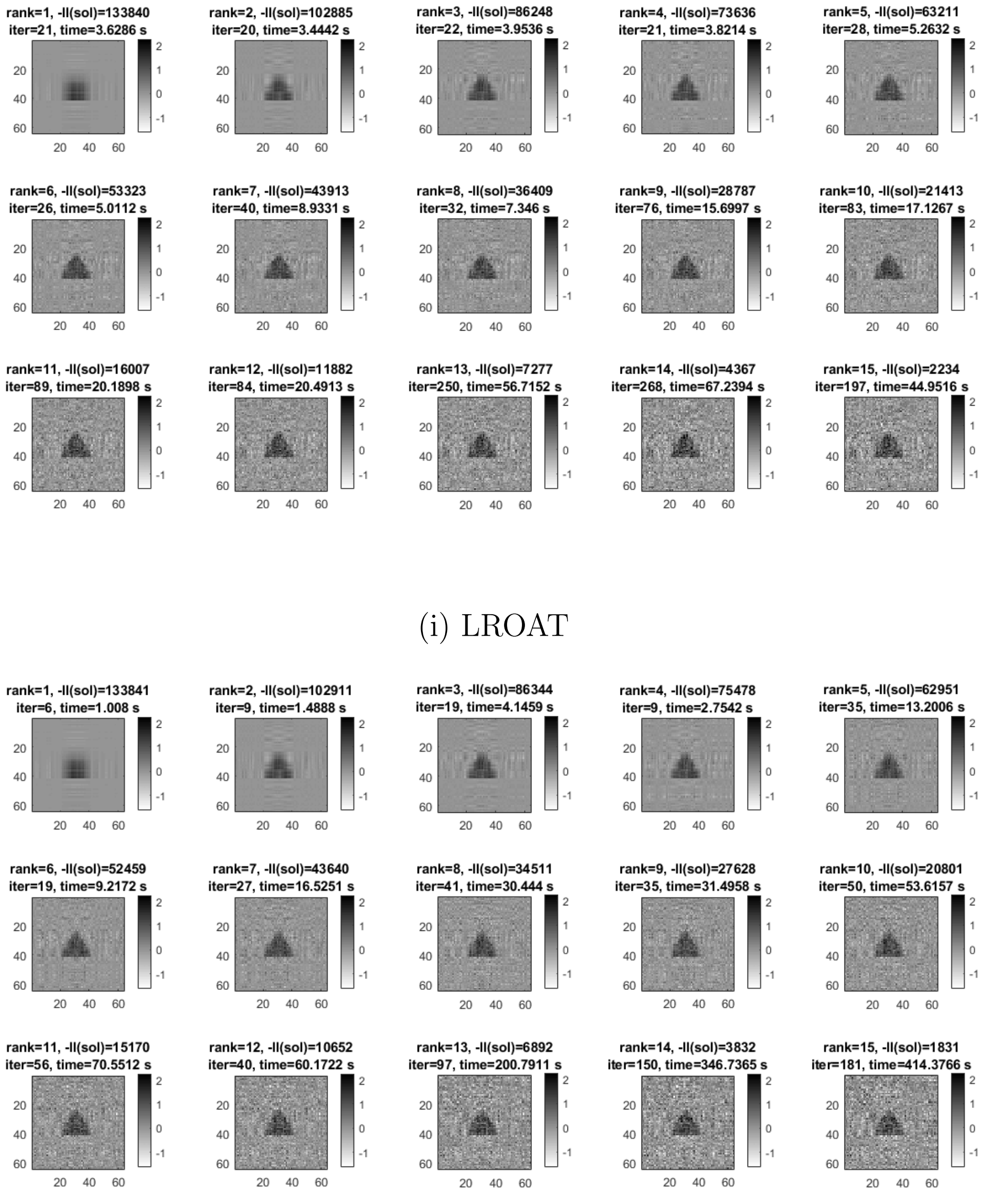
With respect to objective value achieved, neither method seems to have a consistent advantage – sometimes the CPD model achieves a lower objective value and sometimes the LROAT model achieves a lower objective value. That suggests that the projected gradient descent algorithm used to fit the LROAT model works at least as good as the alternating minimization algorithm used to fit the CPD in terms of minimizing the objective function. Both methods also appear to behave similarly with respect to the number of iterations until convergence, with the LROAT model taking slightly longer to converge on average. With respect to computation time, the LROAT model has a distinct advantage. The CPD model has slightly less computation time for ranks 1–3, but longer computation time for ranks ≥ 4 and much longer computation time for ranks ≥ 12. For ranks ≥ 12, the computation time for the CPD model is often on the order of 100's of seconds, whereas for the LROAT model it is on the order of 10's of seconds.



(i) LROAT

(ii) CPD

Figure 2.4: Estimates from the LROAT and CPD models, noise level $\sigma^2 = 1$.

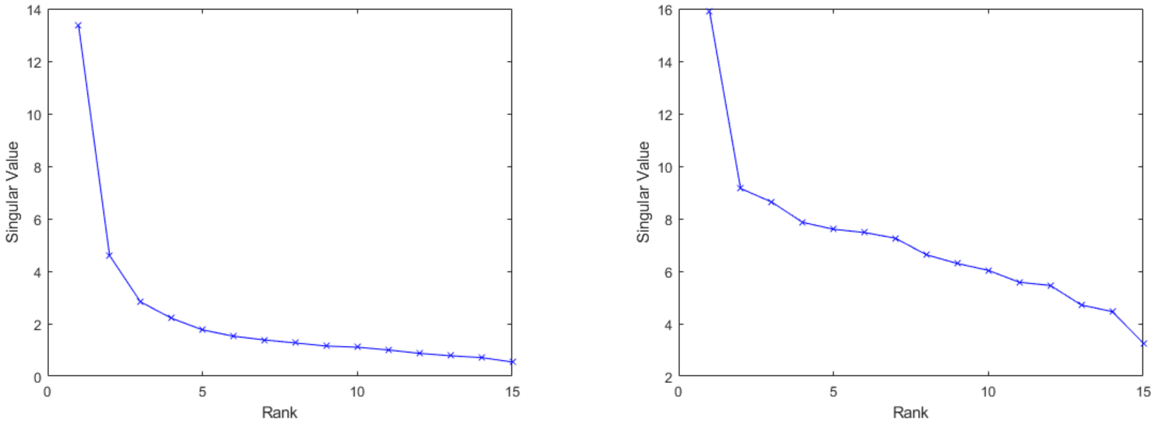


(i) LROAT

(ii) CPD

Figure 2.5: Estimates from the LROAT and CPD models, noise level $\sigma^2 = 10$.

The LROAT model also offers a way to choose an appropriate rank. Since the number of nonzero singular values equals the rank, one possibility is to look at a scree plot of the singular values obtained from the LROAT fit. Although none of the singular values of the estimated parameter tensor will be exactly zero, small singular values should indicate diminishing returns with respect to fitting higher rank models. Thus, one may choose the rank by finding the “elbow” in the scree plot of the singular values. Figure 2.6 shows scree plots of the singular values from the rank-15 fit for each noise level. The elbow is clearly visible for low noise level, but is less distinct for high noise level. However, both plots indicate that a rank-3 or rank-4 model should be sufficient for estimating the parameter tensor. Although the true parameter tensor is rank 13, the estimates shown in Figures 2.4 and 2.5 agree with the scree plots in suggesting that the true parameter tensor can be well-approximated by one of (much) lower rank. In both Figures 2.4 and 2.5, the estimate can be distinguished as a triangle for as low a rank as rank-3. For low noise level, the resolution improves up to about rank-7, so there may be an advantage to fitting higher rank models for signals that take more complex shapes than a triangle. The point here is that most of the major features that comprise a triangle can be discerned for ranks as low as rank-3, and the scree plot of the singular values is in agreement with the visual inspection of the estimates.



(i) $\sigma^2 = 1$

(ii) $\sigma^2 = 10$

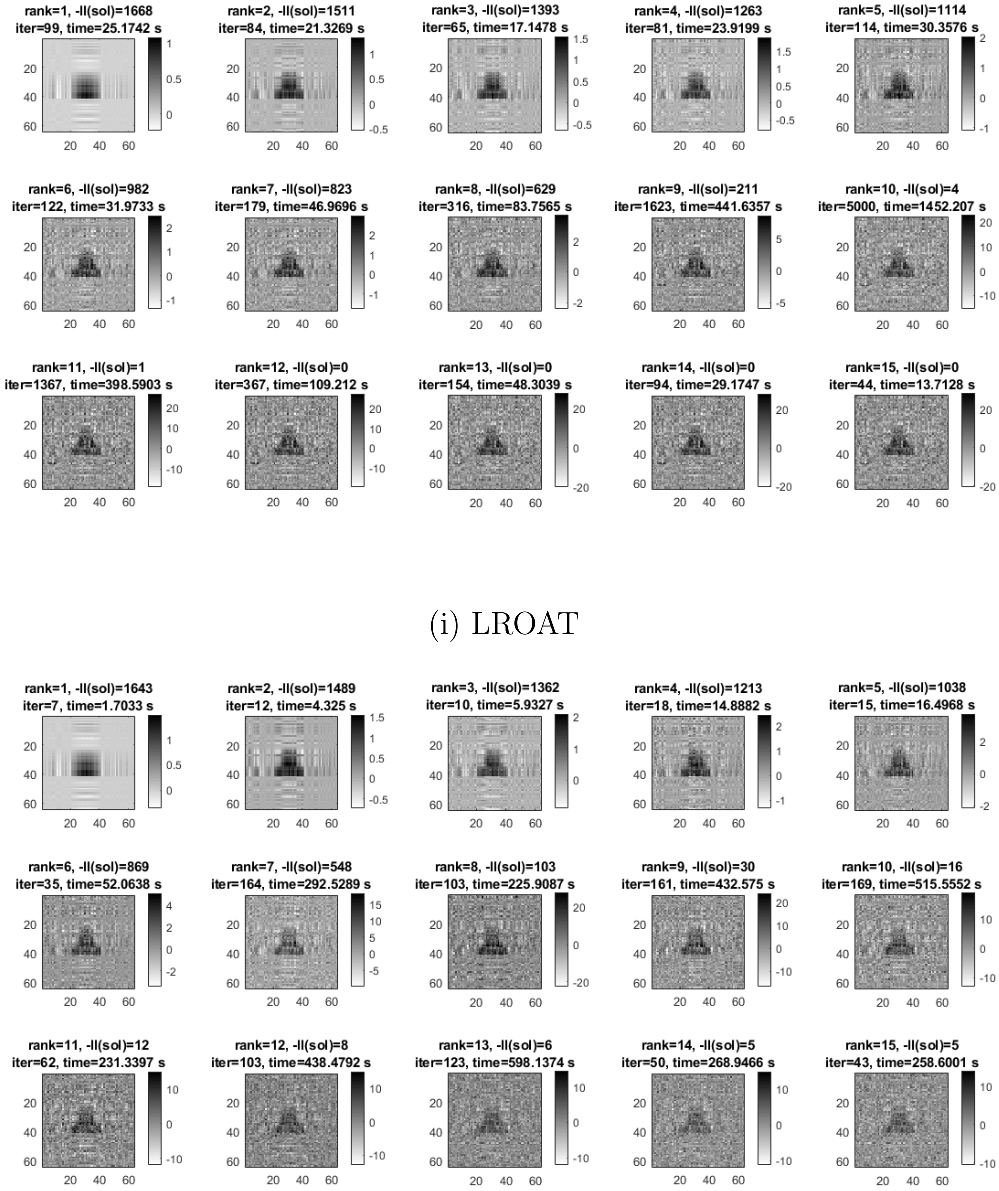
Figure 2.6: Scree plots of the singular values from the LROAT fit.

Results - Experiment 1, Scenario 2

The LROAT and CPD models were fit for ranks 1–20. The results were qualitatively similar to the results from Scenario 1(a). Plots of the estimates and a scree plot of the singular values are included in the Appendix.

Results - Experiment 1, Scenario 3

The LROAT and CPD models were fit for ranks 1–15. Plots of the estimates under each model are shown in Figure 2.7. Note that the color scales differ for each subplot, as indicated by the colorbars to the right of each subplot. Logistic regression is a more challenging estimation problem than the Gaussian linear model. That is reflected by the plots of the estimates in that the estimates do not improve after about rank-3 or rank-4. In addition, the estimates become unreliable for ranks ≥ 7. That can be seen by the large computation times, failure to converge in one instance (the maximum iterations was set to 5000), and the magnitudes of the values indicated on the colorbars. Based on the colorbars, many estimates for higher ranks have elements with values exceeding ± 15. One possible explanation for this phenomenon is quasi-complete separation – *viz.*, that so many parameters are being estimated in the higher rank models that a few elements of the estimated parameter tensor are able to separate the 0-1 response. When quasi-complete separation occurs, maximum likelihood estimates computed numerically tend toward $\pm\infty$. For ranks ≤ 6, the alternating minimization algorithm used to fit the CPD model seems to have a slight advantage over the projected gradient descent algorithm used to fit the LROAT model with respect to the number of iterations until convergence, computation time, and objective value achieved. However, the estimates for the LROAT model seem to remain stable for slightly longer as the rank of the model increases. Based magnitudes of the values of the estimates, the estimates from the CPD model become unreliable for ranks ≥ 7, while the estimates from the LROAT model become unreliable for ranks ≥ 9.

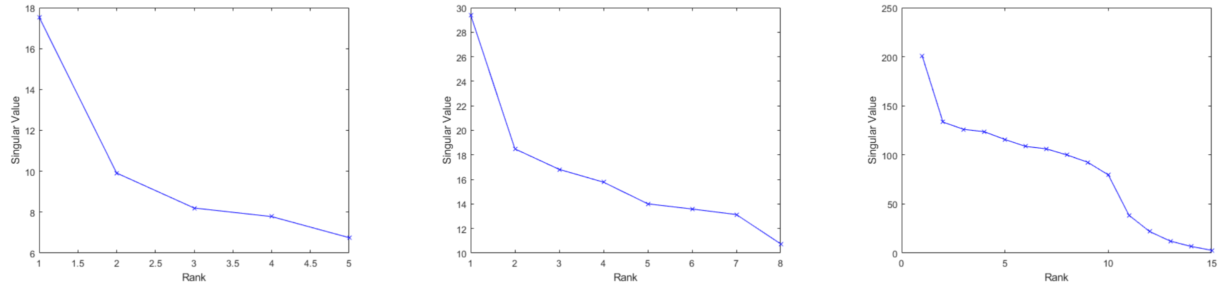


(i) LROAT

(ii) CPD

Figure 2.7: Estimates from the LROAT and CPD models for logistic regression.

Figure 2.8 shows scree plots of the singular values from the rank-5, rank-8, and rank-15 LROAT fits. From the rank-15 fit [Fig. 2.8(iii)], it is clear that a scree plot of the singular values is less useful for choosing the rank if the estimate itself is unreliable. However, the scree plots from the rank-5 and rank-8 fits [Fig. 2.8(i) and (ii)] both suggest that the rank-2 model is sufficient for estimating the parameter tensor. This conclusion agrees with a visual inspection of the estimates in Figure 2.7. That is, the rank-2 estimate captures the major features of the signal, and little is to be gained by fitting higher rank models (in this case, because of the challenging nature of the problem combined with the limited sample size).

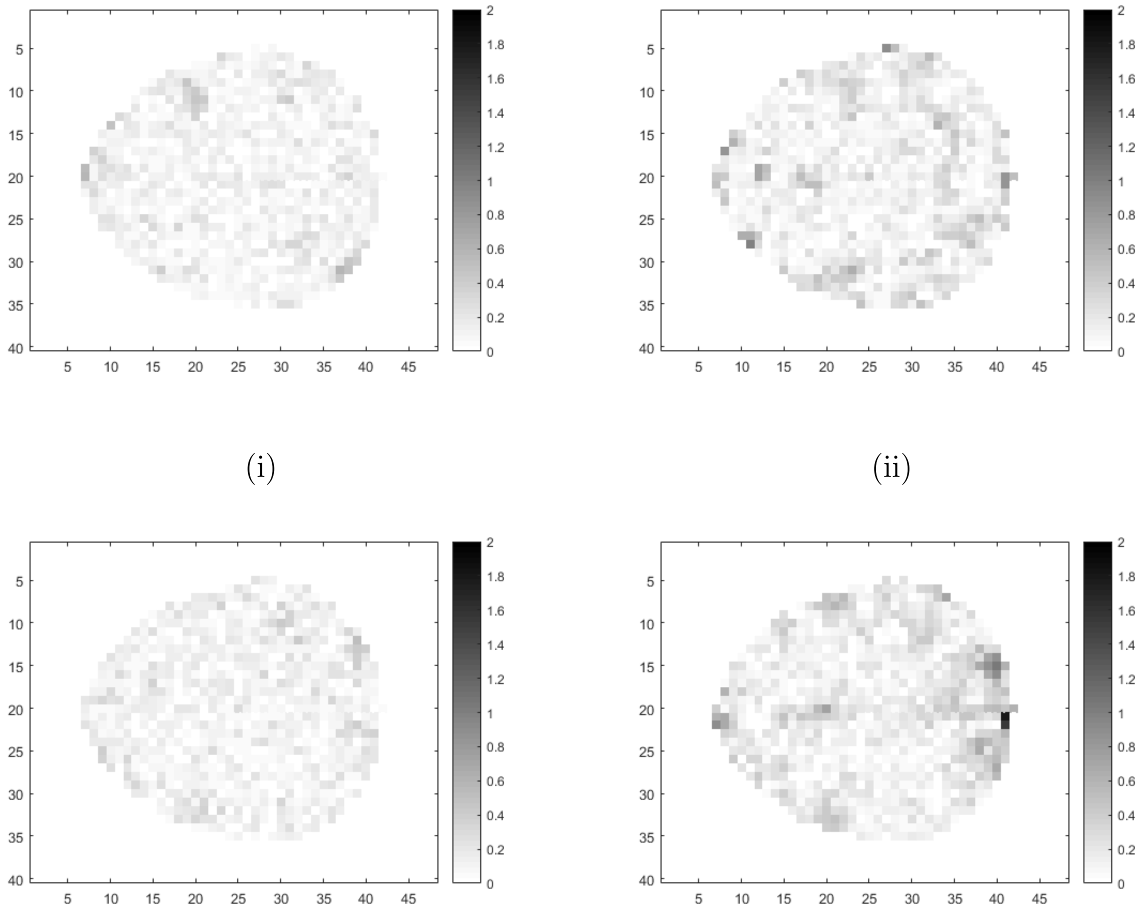


(i) rank-5 (ii) rank-8 (iii) rank-15

Figure 2.8: Scree plots of the singular values from three LROAT fits.

Results - Experiment 1, Scenario 4

We resampled fMRI images from slice 27 for Scenario 4. Figure 2.9 shows the images + small amount of added noise for four randomly selected subjects. Figure 2.10 shows the images + small amount of added noise + added signal for a different set of four subjects. The images in Figures 2.9 and 2.10 are masked for the purposes of visualization, but we used the unmasked images in the simulation. The absolute values of the images are shown so that zeros display as white. Comparing the images in Figures 2.9 and 2.10, one can mostly make out the region of added signal. However, in some examples, the signal region is not obvious [e.g., Fig. 2.10(ii)]. Moreover, the shape of the signal (see Fig. 2.3) is not obvious in any example. Thus, the altered images are realistic in the sense that the signal is expressed non-uniformly in a compact region and to a different extent across subjects. However, the altered images are unrealistic in the sense that the difference between subjects expressing the signal vs. those not expressing the signal would not be as marked.



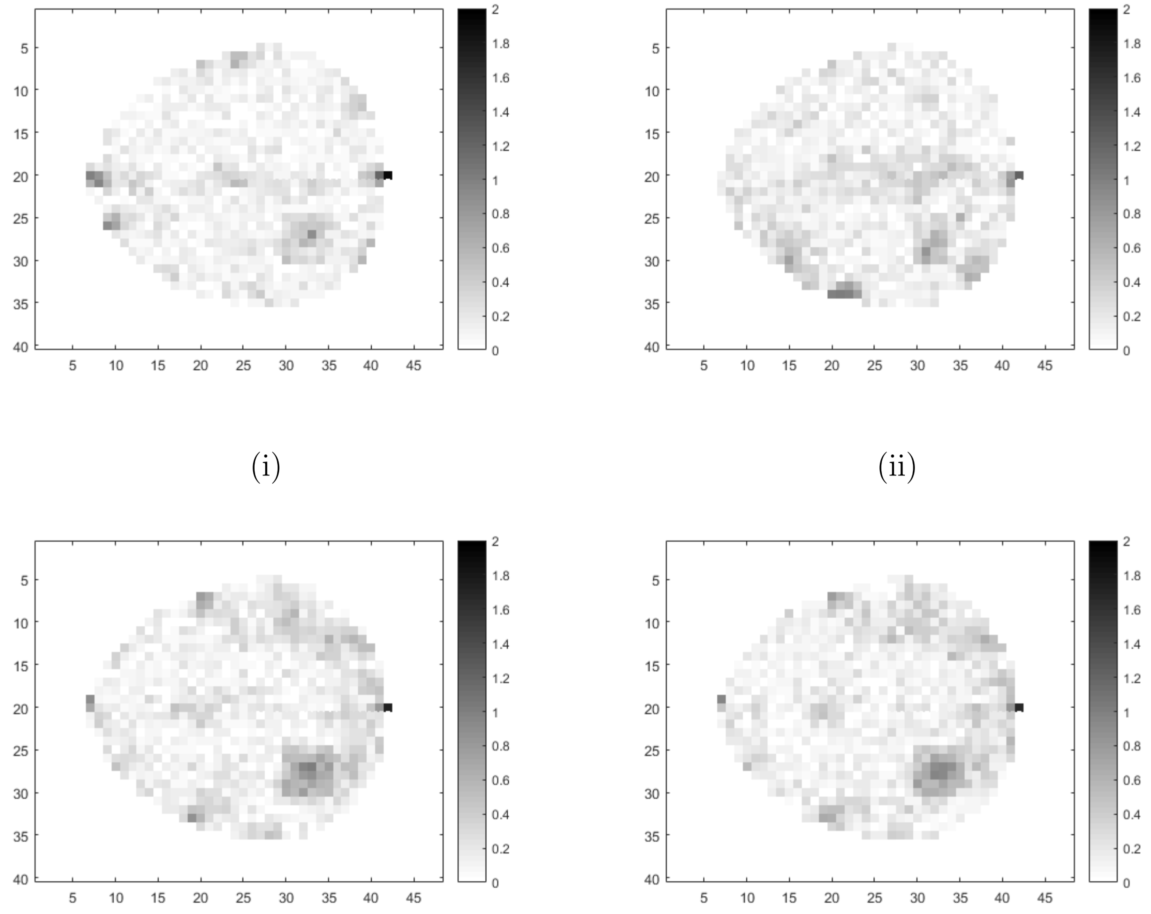
(i)

(ii)

(iii)

(iv)

Figure 2.9: Masked fMRI images from slice 27 with small amount of added noise for four randomly selected subjects.



(i)

(ii)

(iii)

(iv)

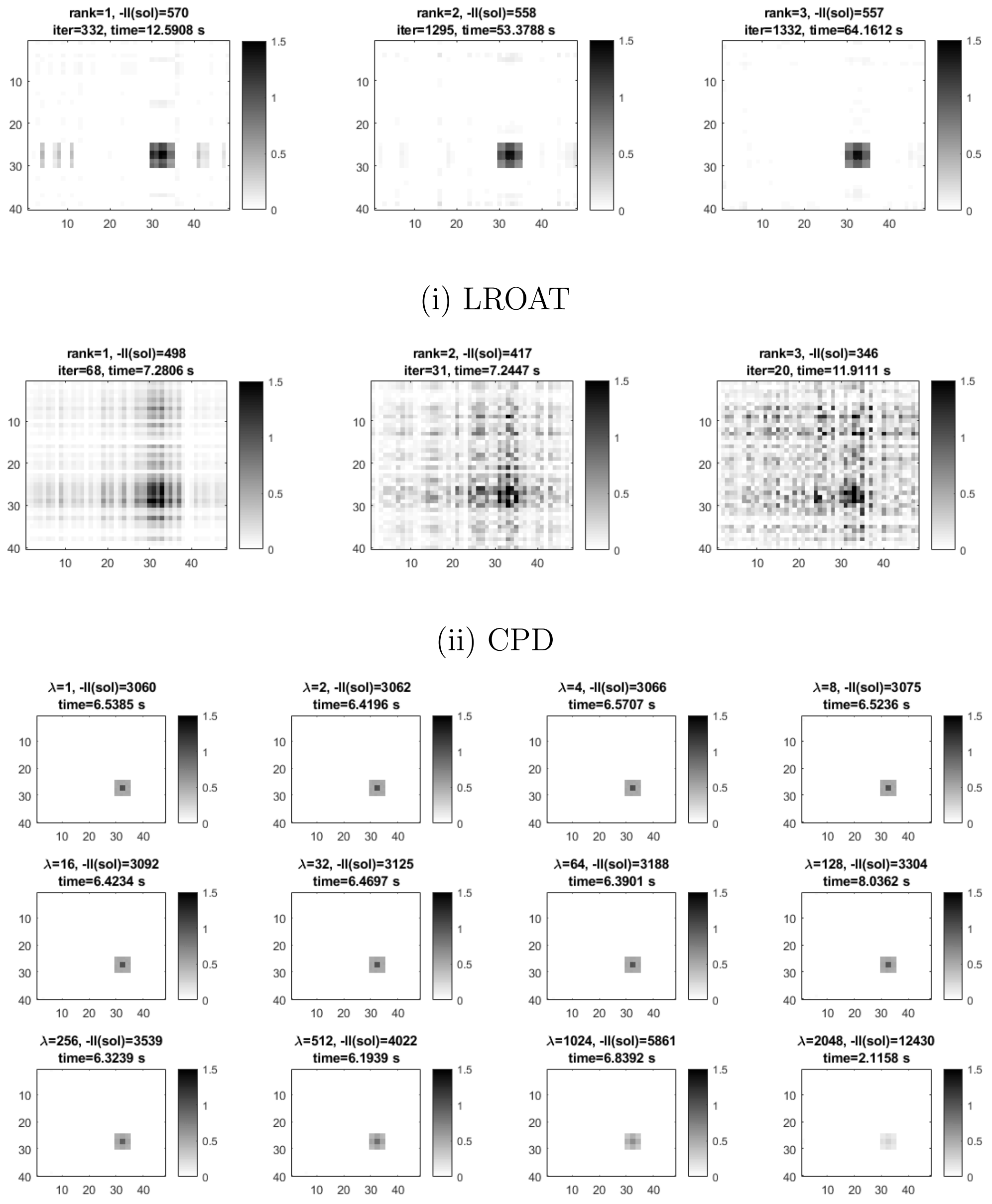
Figure 2.10: Masked fMRI images from slice 27 with small amount of added noise and added signal for four randomly selected subjects.

The LROAT and CPD models were fit for ranks 1–3 and the regularized matrix model was fit for 12 values of the tuning parameter. The MSEs based on 100 simulations are reported in Table 2.1. The regularized matrix model performed the best, with higher penalization (i.e., larger values of the tuning parameter) achieving better MSE. The LROAT model achieved substantially better MSE than the CPD model. For the LROAT model, the rank-2 and rank-3 estimates achieved better MSE than the rank-1 estimate. This was expected because the true signal was rank-2. Although the rank-3 estimate achieved better MSE than the rank-2 estimate, the difference was slight. For the CPD model, the rank-1 estimate achieved the best MSE, and the MSE worsened as the rank increased. This pattern is potentially attributable to the increasing number of parameters to estimate, although the LROAT model should have been similarly affected if that explanation alone sufficed.

Table 2.1: MSE for LROAT, CPD, and regularized matrix models. 100 simulations.

Method		MSE
LROAT	rank-1	27.15
	rank-2	15.02
	rank-3	12.85
CPD	rank-1	68.14
	rank-2	133.71
	rank-3	238.69
Regularized Matrix	$\lambda = 1$	2.23
	$\lambda = 2$	2.22
	$\lambda = 4$	2.22
	$\lambda = 8$	2.20
	$\lambda = 16$	2.18
	$\lambda = 32$	2.13
	$\lambda = 64$	2.06
	$\lambda = 128$	1.94
	$\lambda = 256$	1.73
	$\lambda = 512$	1.40
	$\lambda = 1024$	0.50
	$\lambda = 2048$	1.01

Plots of the estimates from one of the simulations are shown in Figure 2.11. The pattern in the estimated images match the pattern of MSE in Table 2.1. That is, the estimates from the regularized matrix model [Fig. 2.11(iii)] most closely match the true signal from Figure 2.3, indicating that the regularized matrix model performed the best. The LROAT model performed second best [Fig. 2.11(ii)], with the rank-2 and rank-3 estimates matching the true signal more closely than the rank-1 estimate. The estimates from the CPD model [Fig. 2.11(i)] help explain why it did not perform well and why the MSE became worse as the rank increased. Although the CPD model was able to correctly identify the region in which the signal occurred, it was not able to clearly identify the shape of the signal. Moreover, the estimate was much more noisy than the estimates from the LROAT and regularized matrix models. There were many nonzero estimates in the region where the signal was zero, which increased as the rank increased. Note that with respect to the objective values displayed in Figure 2.11, the CPD model performed the best, followed by the LROAT model then the regularized matrix model [the $P_\lambda(\sigma(B))$ part of the regularized matrix objective function was omitted to make the values comparable]. That the pattern in the objective value was opposite the pattern in MSE was surprising. However, it may indicate overfitting. Only $6 + 6 = 12$ nonzero parameters need to be estimated to correctly identify the signal region for a rank-1 fit, and only $6 + 6 + 2 + 2 = 16$ nonzero parameters need to be estimated to identify the signal perfectly for a rank-2 fit. The LROAT and regularized matrix models came closer to the correct level of parsimony than did the CPD model.



(i) LROAT

(ii) CPD

(iii) Regularized matrix

Figure 2.11: Estimates from the LROAT, CPD, and regularized matrix models for one simulation.

2.3.2 Experiment 2

For 3D covariates, generating data under model (2.4) is not necessarily equivalent to generating data under model (2.7).

For two scenarios, we generate data under model (2.4):

1. $Y_i \sim N(\beta_0 + <\mathcal{X}_i, \mathcal{B}>,\ 1),\ (\mathcal{X}_i)_{jk} \sim N(0,\ 1)$,
   $\mathcal{B}$: a $25 \times 25 \times 25$ image of two hyperrectangles
   (rank 2), sample size: 400

2. $Y_i \sim N(\beta_0 + <\mathcal{X}_i, \mathcal{B}>,\ 1),\ (\mathcal{X}_i)_{jk} \sim N(0,\ 1)$,
   $\mathcal{B}$: a $32 \times 32 \times 32$ image of a cross
   (rank 2), sample size: 600

Note that for Scenarios 1 and 2, the true parameter tensor does not admit an orthogonal decomposition.

For three scenarios, we generate data under model (2.7):

3. $Y_i \sim N(\beta_0 + <\mathcal{X}_i, \mathcal{B}>,\ 1),\ (\mathcal{X}_i)_{jk} \sim N(0,\ 1)$,
   $\mathcal{B}$: a $32 \times 32 \times 32$ image of an orthogonally-decomposable cross
   (rank 2), sample size: 500

4. $Y_i \sim N(\beta_0 + <\mathcal{X}_i, \mathcal{B}>,\ 1),\ (\mathcal{X}_i)_{jk} \sim N(0,\ 1)$,
   $\mathcal{B}$: a $32 \times 32 \times 32$ image of an orthogonally-decomposable circle
   (rank 3), sample size: 600

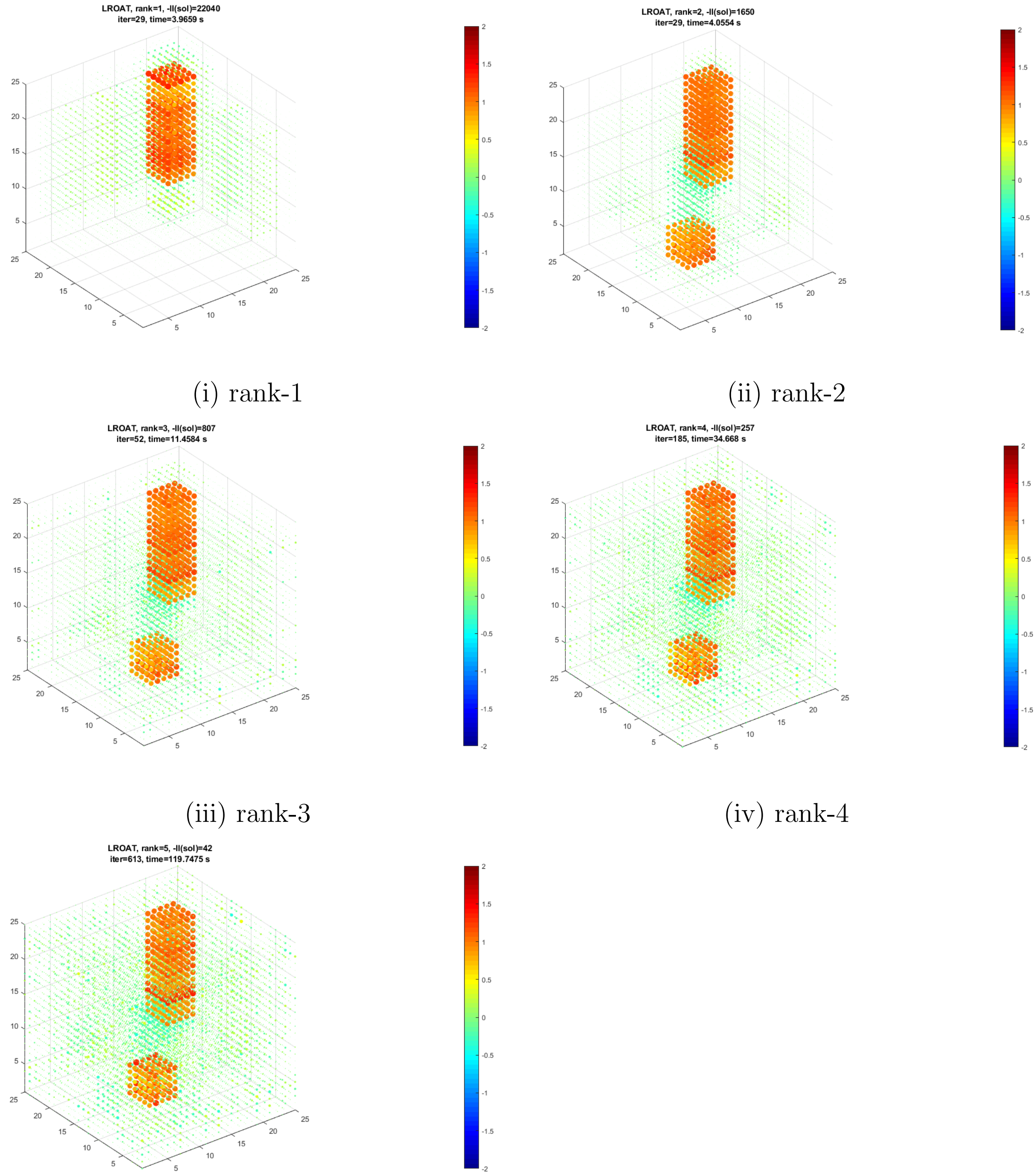
5. $Y_i \sim N(\beta_0 + <\mathcal{X}_i, \mathcal{B}>,\ 1),\ (\mathcal{X}_i)_{jk} \sim N(0,\ 1)$,
   $\mathcal{B}$: a $32 \times 32 \times 32$ image of an orthogonally-decomposable triangle
   (rank 9), sample size: 1200

As in Experiment 1, we fit the LROAT and CPD models for a sequence of ranks.

Results - Experiment 2, Scenario 1

The LROAT and CPD models were fit for ranks 1–5. Plots of the estimates under the LROAT and CPD models are shown in Figures 2.12 and 2.13, respectively. The color of a dot corresponds to the value of an element of the estimated tensor; a colorbar is included to the right of each subplot for reference. The size of a dot corresponds to the magnitude of an element of the estimated tensor. The rank-1 estimates for the LROAT and CPD models are very similar. The rank-2 estimate for the CPD model matches the true signal almost perfectly [see Fig. 2.1(iii)] , while the rank-2 estimate for the LROAT model matches the closest orthogonal approximation to the true signal [see Fig. 2.1(iv)]. As a consequence, the objective value is much lower for the rank-2 CPD estimate than for the rank-2 LROAT estimate. For ranks ≥ 3, the estimates from the CPD model are very noisy. The rank-3 estimate partially captures the true signal, while the rank-4 and rank-5 estimates miss the true signal entirely. In contrast, the rank-3, rank-4, and rank-5 estimates from the LROAT model all partially capture the true signal, though they all resemble the closest orthogonal approximation to the true signal rather than the true signal itself. The estimates also contain more noise as the rank increases, which partially occludes the signal region.

For the CPD model, the results suggest that the alternating minimization algorithm struggles to find the global optimum as the number of free parameters approaches the sample size (for the rank-5 model, there are 375 parameters to estimate with a sample size of $N = 400$). For the LROAT model, the results suggest that there is no advantage to fitting higher rank models than the rank of the true signal to ameliorate the effects caused by the orthogonality constraints.



(i) rank-1

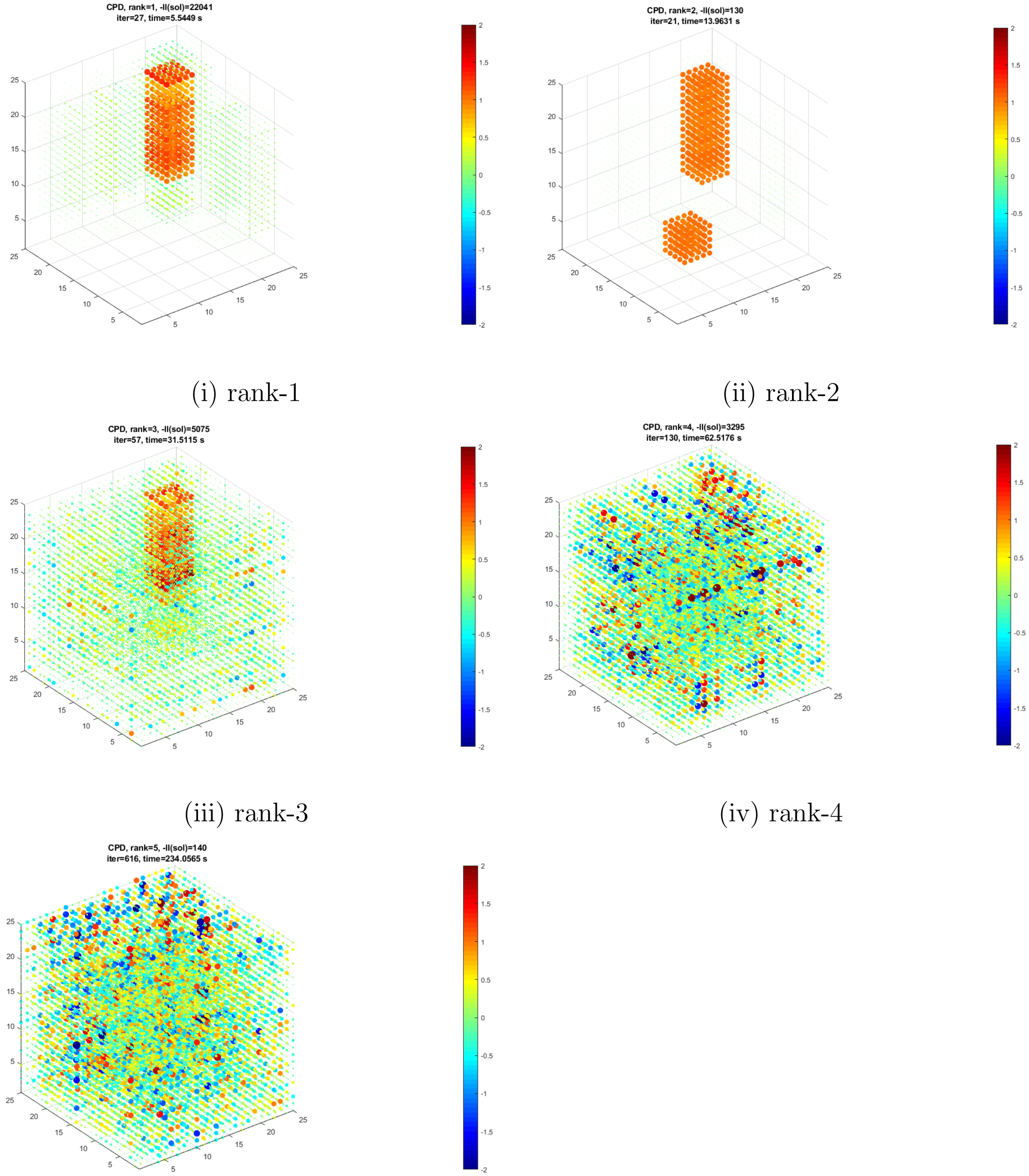
(ii) rank-2

(iii) rank-3

(iv) rank-4

(v) rank-5

Figure 2.12: Estimates from the LROAT model.



(i) rank-1

(ii) rank-2

(iii) rank-3

(iv) rank-4

(v) rank-5

Figure 2.13: Estimates from the CPD model.

As with 2D images, one can make a scree plot of the singular values to choose an appropriate rank. Figure 2.14 shows a scree plot of the singular values from the rank-5 fit. The plot appears to favor the rank-2 fit, which is the rank of the true signal.

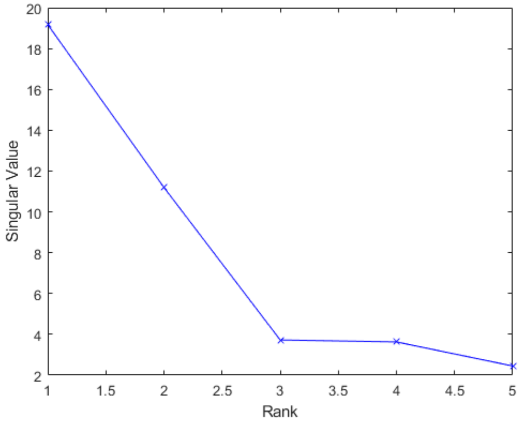


Figure 2.14: Scree plot of the singular values from the LROAT fit.

Results - Experiment 2, Scenarios 2 − 4

For Scenario 2, the LROAT and CPD models were fit for ranks 1–5. The results were qualitatively similar to the results from Scenario 1. Plots of the estimates and a scree plot of the singular values are included in the Appendix. For Scenarios 3 and 4, the LROAT and CPD models were fit for ranks 1–5 and ranks 1–6, respectively. The results for Scenarios 3–5 are all qualitatively similar, so we report the results from Scenario 5 in the next section and include plots of the estimates and scree plots of the singular values for Scenarios 3 and 4 in the Appendix.

Results - Experiment 2, Scenario 5

A plot of the true signal for Scenario 5 is shown in Figure 2.15. The signal was constructed by downsizing the image used for Experiment 1, Scenario 1 to 32×32, then finding the SVD of the resulting image. The singular values and corresponding singular vectors were then used construct the image via a 3-way outer product.

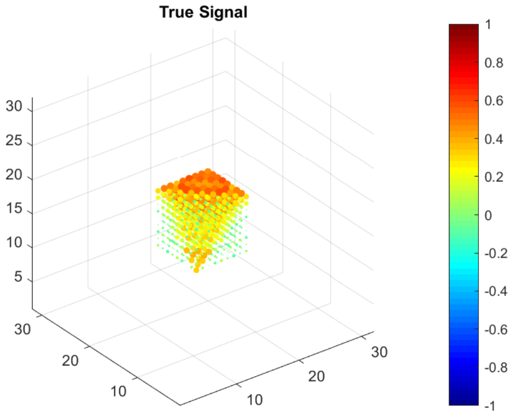
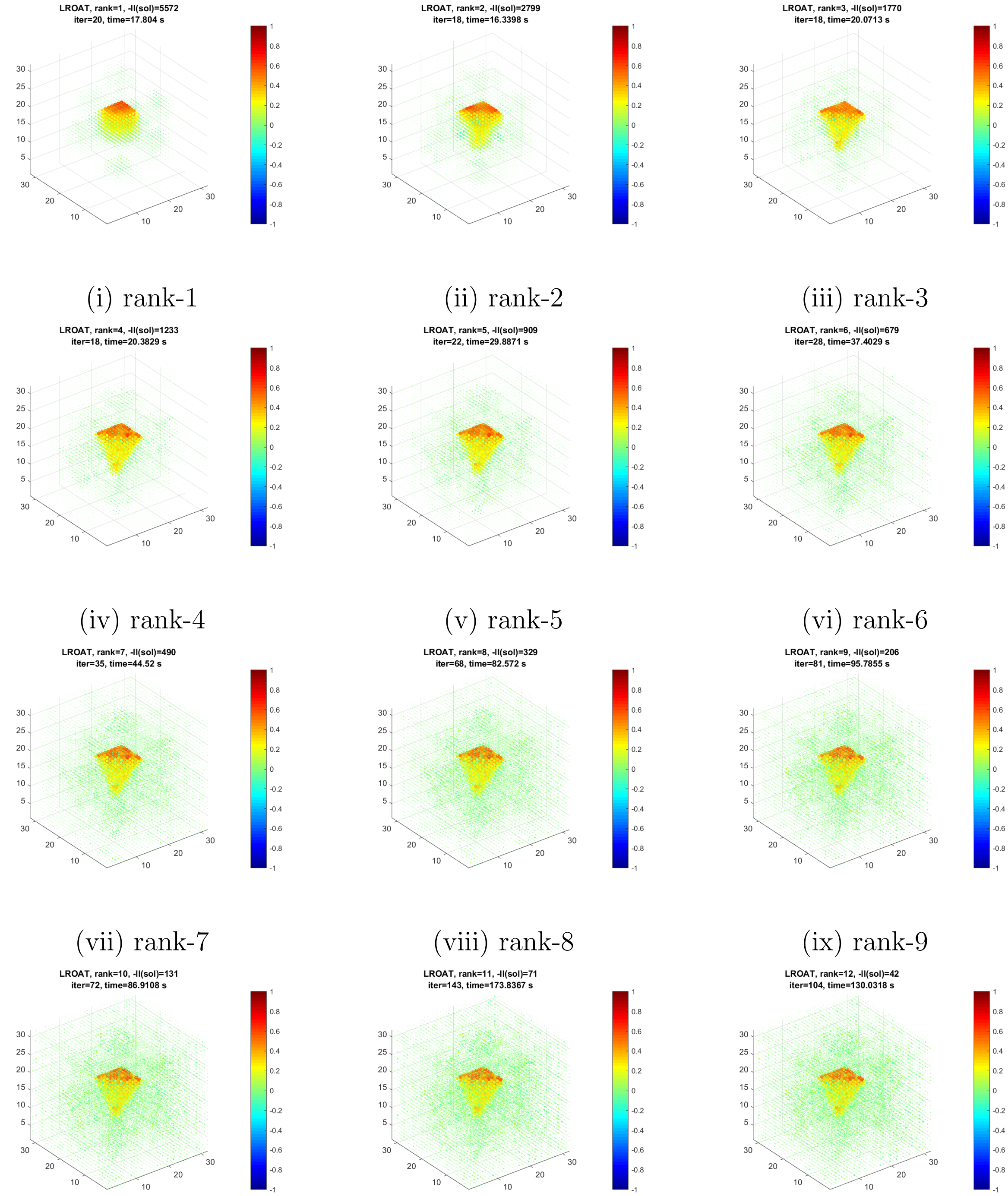


Figure 2.15: True signal: orthogonal triangle (rank 9).

The LROAT and CPD models were fit for ranks 1–12. Plots of the estimates under the LROAT and CPD models are shown in Figures 2.16 and 2.17, respectively. The plots of the estimates and objective values achieved are similar for the LROAT and CPD models up to about rank-6. For ranks ≥ 6, the estimates from the CPD model become more noisy than the estimates from the LROAT model. For ranks ≥ 10, the noise in the CPD estimates is so extreme that the signal region is almost completely occluded. Although the LROAT estimates also become more noisy for higher rank models, they do so to a lesser extent and the signal region remains clearly visible. These results suggest that there may be an advantage to assuming the parameter tensor is orthogonally decomposable when it is in fact orthogonally decomposable. However, from Scenarios 1–4, it is apparent that the alternating minimization algorithm used to fit the CPD model becomes less effective at minimizing the objective function as the number of parameters to estimate approaches the sample size. Thus, it is not possible to determine to what extent the advantages of the LROAT model over the CPD model are a consequence of differences in the model assumptions vs. the estimation algorithms.

The LROAT model also has an advantage with respect to computation time and number of iterations until convergence. When the computation time is on the order of 100's of seconds for the CPD model, it is on the order of 10's of seconds for the LROAT model. When the computation time is on the order of 1000's of seconds for the CPD model, it is on the order of 100's of seconds for the LROAT model.

A scree plot of the singular values from the rank-12 LROAT fit is shown in Figure 2.18. As in Experiment 1, Scenario 1, the scree plot suggests that a rank much lower than the rank of the true parameter tensor is adequate to fit the data. Figure 2.18 suggests that the rank-3 or rank-4 model is appropriate. Comparing the estimates in Figure 2.16(iii)–(iv) to the true signal in Figure 2.15, we can see that the rank-3 and rank-4 estimates capture most of the major features of the true signal, even though the true signal has rank 9. Thus, although higher rank models may offer some minor refinements for estimating the true signal, most of the main features are captured for ranks as low as rank-3.



(i) rank-1

(ii) rank-2

(iii) rank-3

(iv) rank-4

(v) rank-5

(vi) rank-6

(vii) rank-7

(viii) rank-8

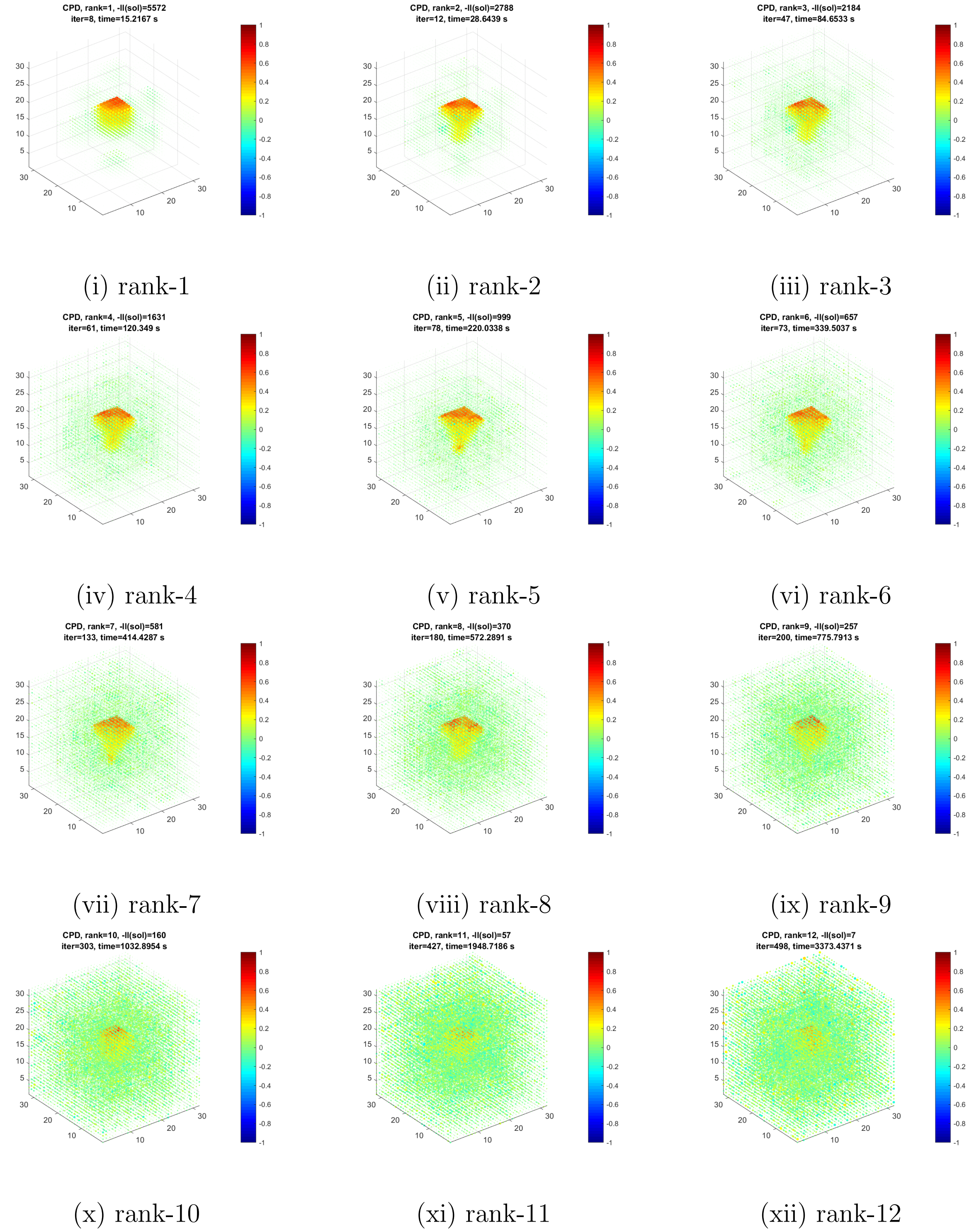
(ix) rank-9

(x) rank-10

(xi) rank-11

(xii) rank-12

Figure 2.16: Estimates from the LROAT model.



(i) rank-1

(ii) rank-2

(iii) rank-3

(iv) rank-4

(v) rank-5

(vi) rank-6

(vii) rank-7

(viii) rank-8

(ix) rank-9

(x) rank-10

(xi) rank-11

(xii) rank-12

Figure 2.17: Estimates from the CPD model.

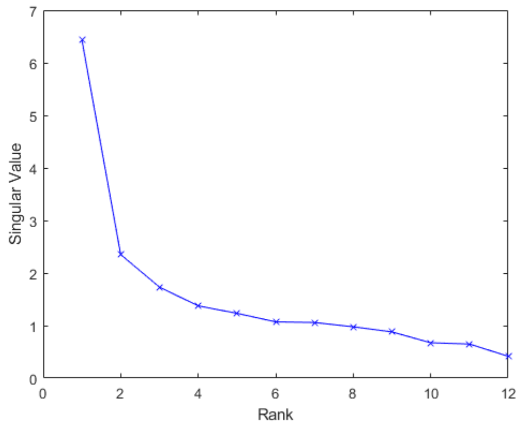


Figure 2.18: Scree plot of the singular values from the LROAT fit.

2.3.3 Experiment 3

We use three scenarios in Experiment 3:

1. $Y_i \sim N(\beta_0 + < \mathcal{X}_i, \mathcal{B} >,\ 1),\ (\mathcal{X}_i)_{jk} \sim N(0,\ 1),$

   $\mathcal{B}$: a $25 \times 25 \times 25$ image of two hyperrectangles

   (rank 2), sample size: 400

2. $Y_i \sim N(\beta_0 + < \mathcal{X}_i, \mathcal{B} >,\ 1),\ (\mathcal{X}_i)_{jk} \sim N(0,\ 1),$

   $\mathcal{B}$: a $32 \times 32 \times 32$ image of an orthogonally-decomposable triangle

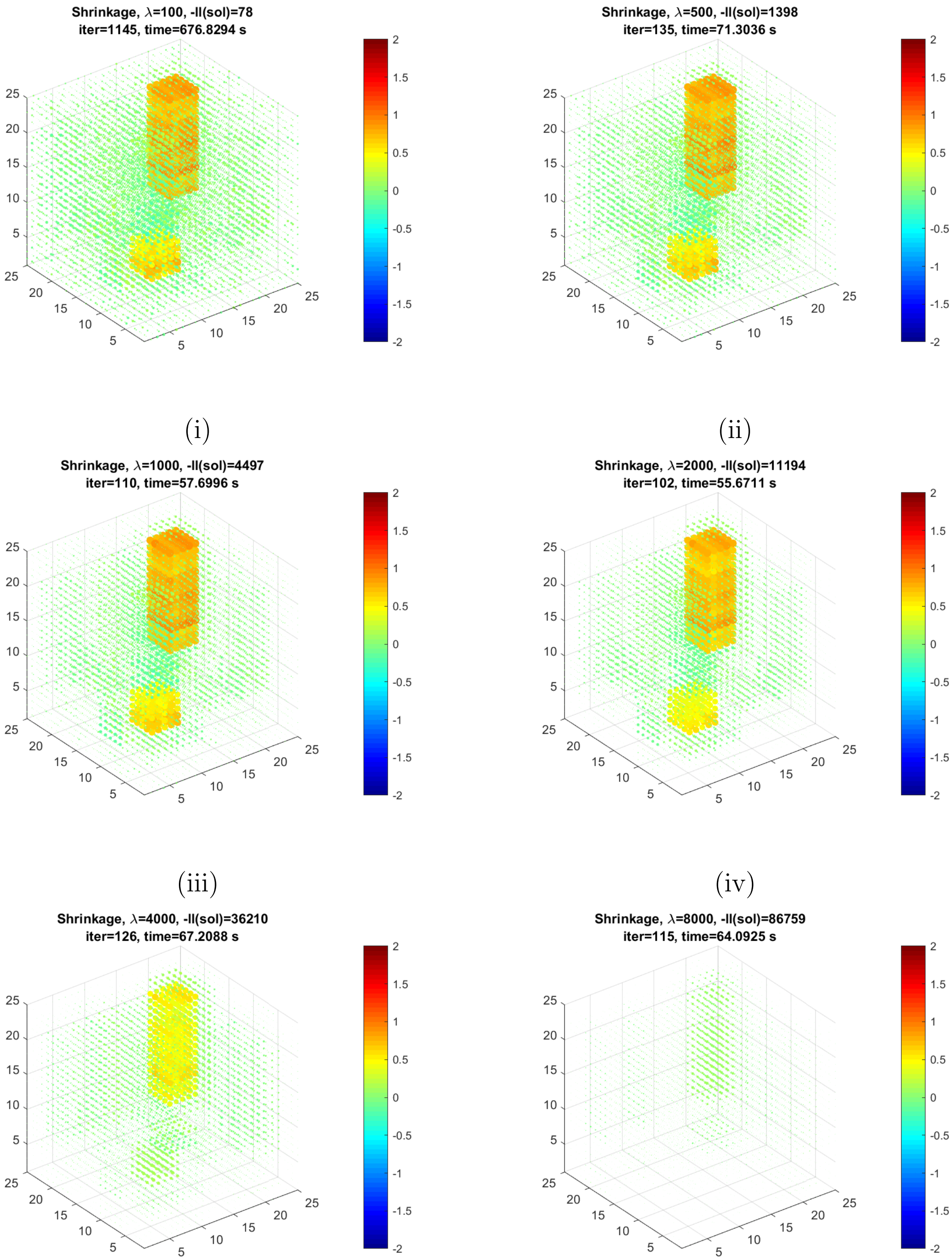
   (rank 9), sample size: 1200

3. A 3D version of Scenario 4 from Experiment 1: instead of slices of real fMRI images, we resample the whole-brain images 1000 times; instead of using a small square nested inside of a larger square for the signal, we use a small cube nested inside a larger cube.

For Scenarios 1 and 2, we fit the regLROAT model for a sequence of values for the tuning parameter. The LROAT and CPD models were already fit to data generated under these scenarios in Experiment 2, so we compare to those results. For Scenario 3, we fit the regLROAT model for a sequence of values for the tuning parameter and the LROAT and CPD models for a sequence of fixed ranks.

Results - Experiment 3, Scenario 1

Estimates from the regLROAT model are shown in Figure 2.19 for a sequence of values of the tuning parameter. For small values of the tuning parameter, the estimates most closely resemble the higher rank estimates from the LROAT model in Experiment 2, Scenario 1 [e.g., Fig. 2.12(v)]. That is, the estimates are noisy and do not capture the true signal exactly. The latter was expected because the true signal does not admit an orthogonal decomposition. For larger values of the tuning parameter, the noise begins to clear and the estimates more closely resemble the lower rank estimates from the fixed-rank LROAT model. The most striking difference between the estimates for the regLROAT model and the estimates for the fixed-rank LROAT model is that the regLROAT estimates appear intermediate in rank. For example, the estimate in Figure 2.19(v) appears to be intermediate between rank-1 and rank-2 estimates [i.e., between Fig. 2.12(i) and (ii)], and the estimate in Figure 2.19(vi) appears to be intermediate between rank-0 and rank-1 [i.e., between Fig. 2.12(i) and the zero tensor]. The plots illustrate that soft thresholding can result in continuous changes in the estimated parameter tensor, whereas hard thresholding (i.e., fixing the rank) can only result in a finite number of discrete changes in the estimated parameter tensor.



(i)

(ii)

(iii)

(iv)

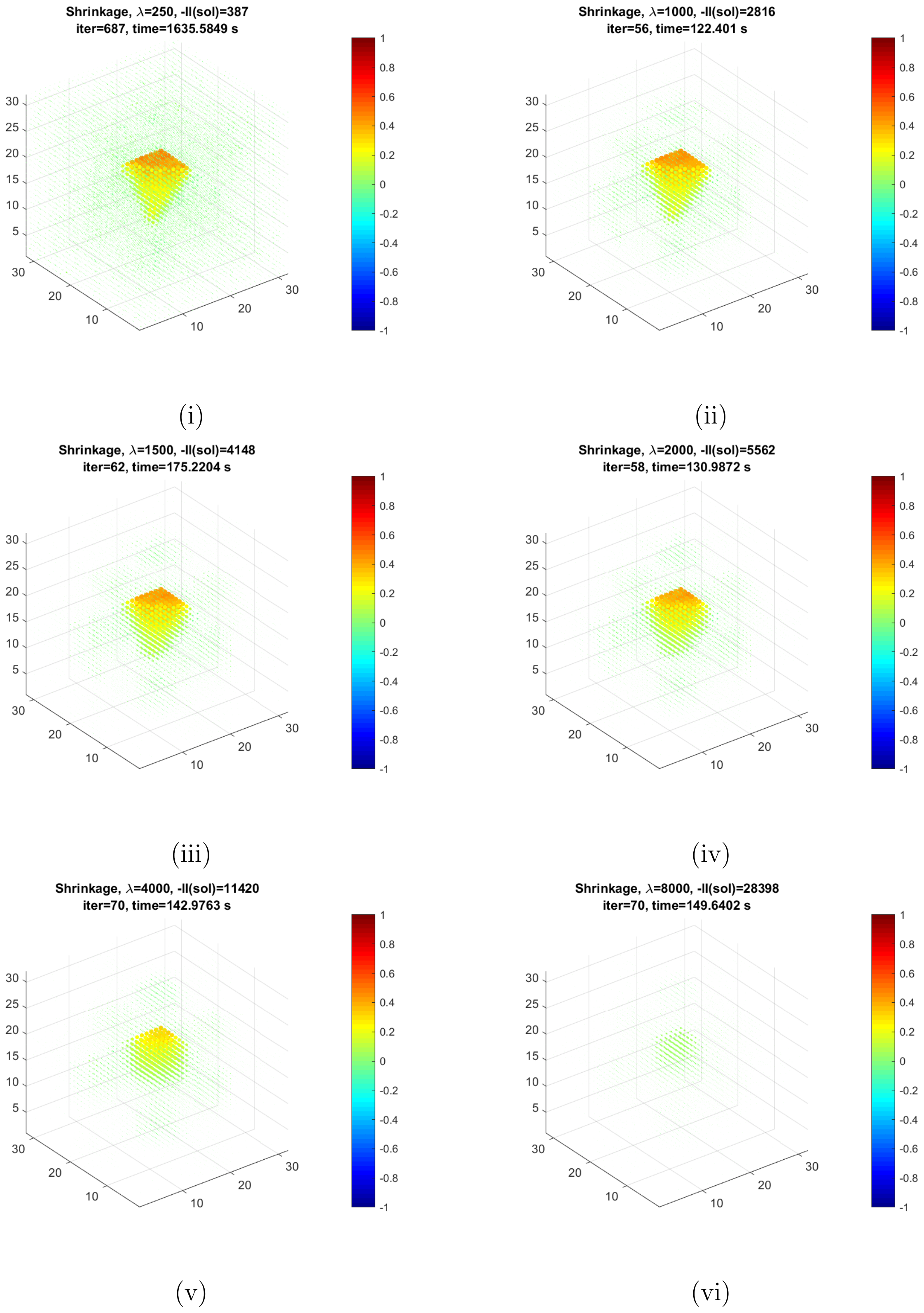
(v)

(vi)

Figure 2.19: Estimates from the regLROAT model.

Results - Experiment 3, Scenario 2

Estimates from the regLROAT model are shown in Figure 2.20 for a sequence of values of the tuning parameter. In general, the plots are similar to the plots shown in Figure 2.16 for the fixed-rank LROAT model. However, as in Experiment 3, Scenario 1, some of the estimates appear to be intermediate in rank. For example, the estimates in both Figure 2.20(v) and (vi) appear to be intermediate between rank-0 and rank-1 [i.e., between Fig. 2.16(i) and the zero tensor].



Shrinkage, λ=250, -ll(sol)=387
iter=687, time=1635.5849 s

(i)

Shrinkage, λ=1000, -ll(sol)=2816
iter=56, time=122.401 s

(ii)

Shrinkage, λ=1500, -ll(sol)=4148
iter=62, time=175.2204 s

(iii)

Shrinkage, λ=2000, -ll(sol)=5562
iter=58, time=130.9872 s

(iv)

Shrinkage, λ=4000, -ll(sol)=11420
iter=70, time=142.9763 s

(v)

Shrinkage, λ=8000, -ll(sol)=28398
iter=70, time=149.6402 s

(vi)

Figure 2.20: Estimates from the regLROAT model.

Results - Experiment 3, Scenario 3

A plot of the true signal for Scenario 3 is shown in Figure 2.21. Similar to Experiment 1, Scenario 4, the inner $2 \times 2 \times 2$ cube takes value 0.6 and the outer $6 \times 6 \times 6$ cube takes value 0.3. As in the 2D version of the experiment, the signal has rank 2; however, the signal is not orthogonally decomposable. Note that the 3D version of the parameter tensor represents a more difficult estimation problem not only with respect to the total number of parameters involved, but also with respect to the proportion of nonzero parameters. For the 2D version, the proportion of nonzero parameters is $\frac{6\times6}{40\times48} = 0.01875$. For the 3D version, the proportion of nonzero parameters is $\frac{6\times6\times6}{40\times48\times38} = 0.00296$. Also note that the signal shown in Figure 2.21 denotes the true signal on average. For a given image, the actual values added to the image are randomly drawn from $N(0.30, 0.03^2)$ and $N(0.60, 2(0.03)^2)$.

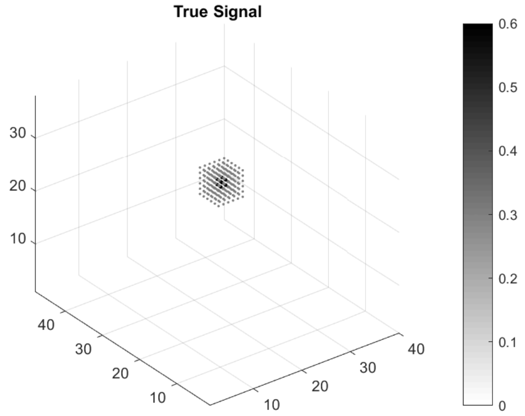
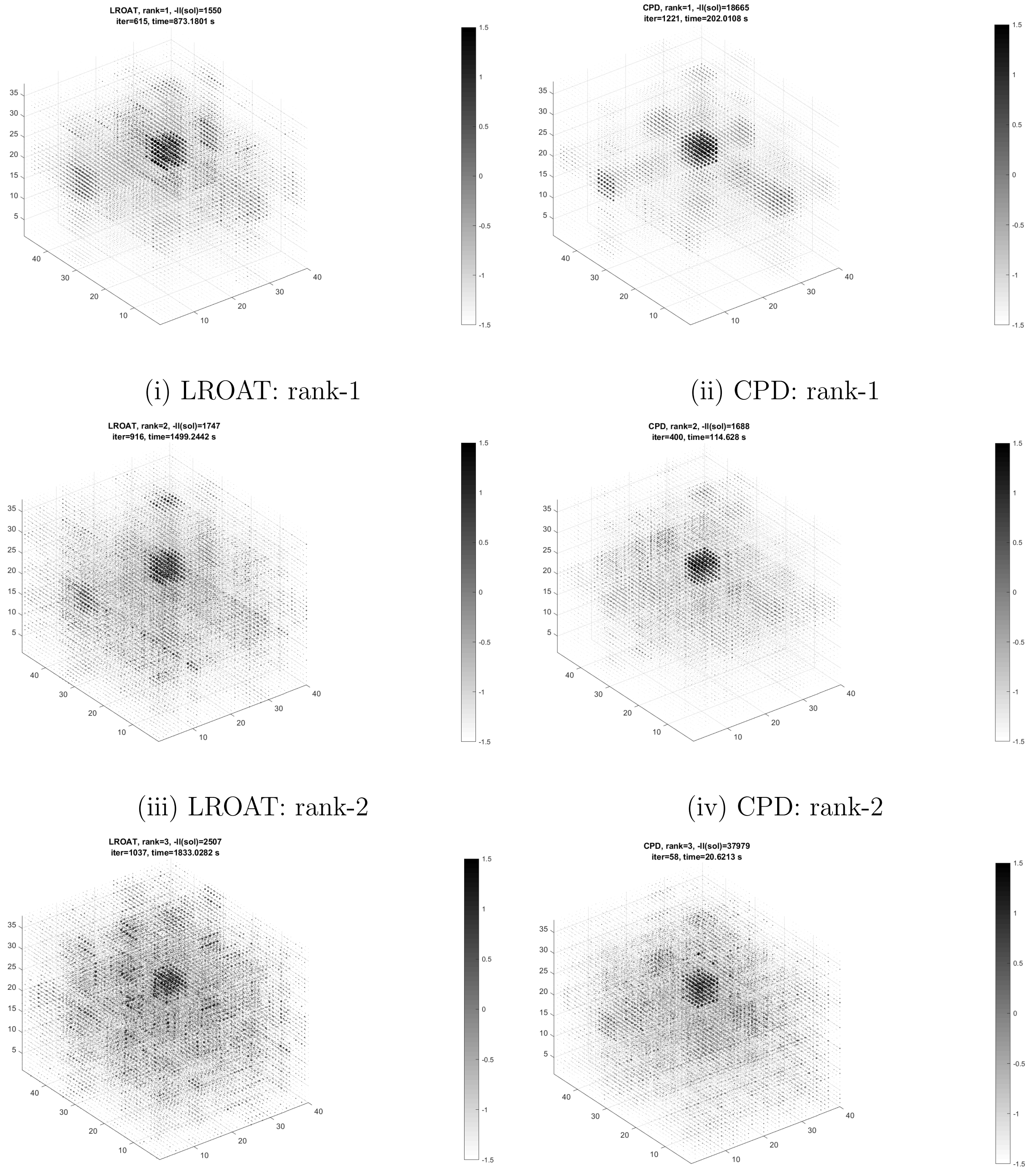


Figure 2.21: True signal: nested cubes (rank 2).

The LROAT and CPD models were fit for ranks 1–3. The estimates for both models are shown shown in Figure 2.22. The regLROAT model was fit for a sequence of values of the tuning parameter. The estimates for the regLROAT model are shown in Figure 2.23. As in the 2D version of the experiment (see Fig. 2.11), the estimates from the regLROAT model are less noisy and more closely resemble the true signal than the estimates from the LROAT or CPD models. For even the smallest values of the tuning parameter [e.g., Fig. 2.23(i)], the estimates from the regLROAT model are less noisy than the estimates from the LROAT or CPD models, regardless of the rank. For larger values of the tuning parameter [e.g., Fig. 2.23(v) and (vi)], the estimates from the regLROAT model correctly identify the signal region as nonzero and the rest of the parameter tensor as zero (or near zero). However, for the tuning parameter values shown, the estimates from regLROAT model do not distinguish between the inner and outer cubes comprising the true signal without also estimating many parameters in the non-signal region as nonzero. Thus, refining the estimate of the true signal seems to come at the cost of estimating additional noise.

Unlike the 2D version of the experiment, the CPD model seemed to perform slightly better than the LROAT model. In the 2D version, the estimates from the LROAT model were less noisy and more accurately captured the shape of the true signal than the estimates from the CPD model [see Fig. 2.11(i) vs. (ii)]. In the 3D version, neither model produced estimates that accurately captured the shape of the true signal (although they did correctly identify the region of the signal) and the estimates from the LROAT model contained more noise than the estimates from the CPD model. One possible explanation for this opposite behavior is that the true signal in the 2D version admits an orthogonal decomposition while the true signal in the 3D version does not. Thus, the LROAT model encounters additional challenges when moving from 2D to 3D than does the CPD model.



(i) LROAT: rank-1

(ii) CPD: rank-1

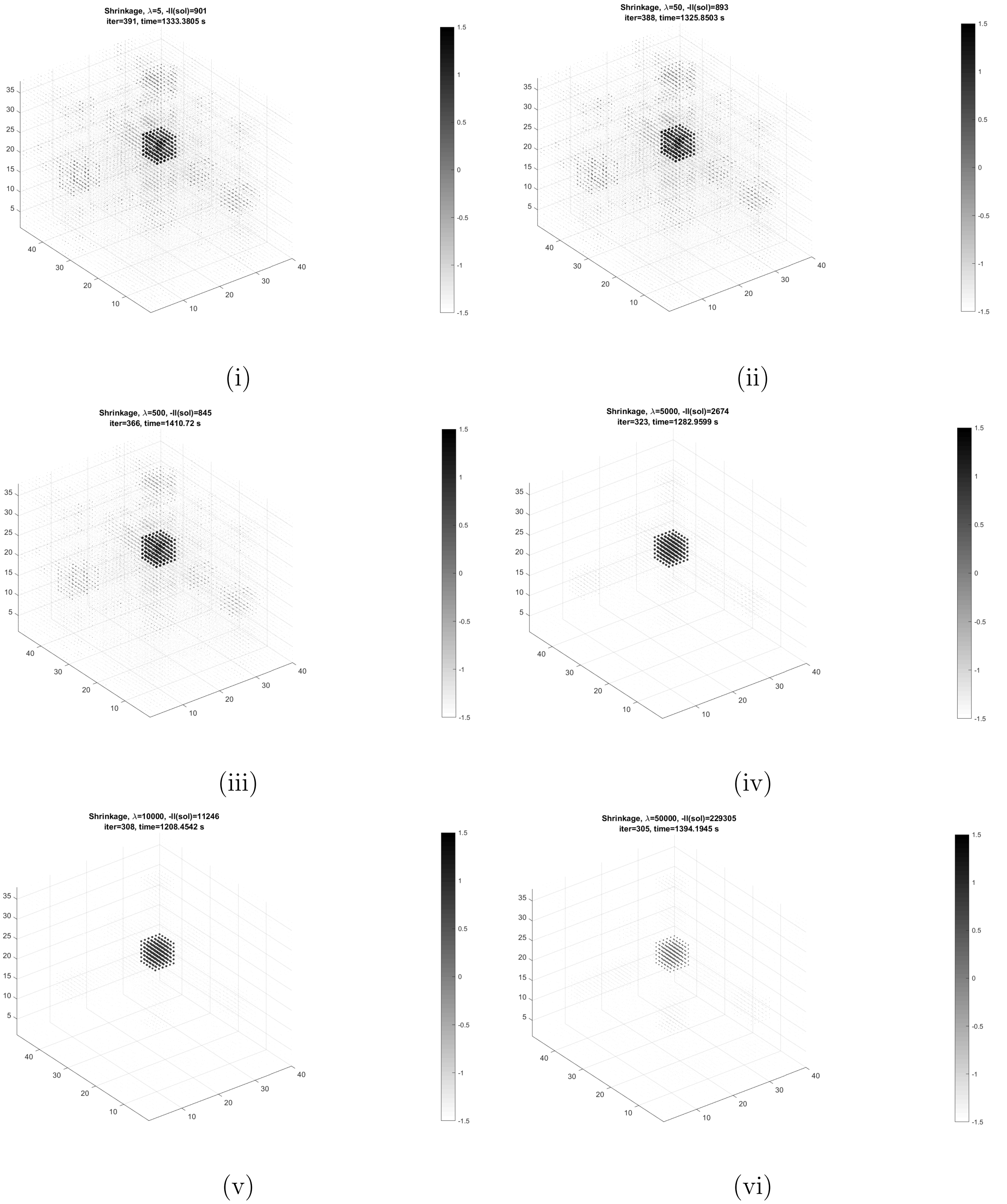
(iii) LROAT: rank-2

(iv) CPD: rank-2

(v) LROAT: rank-3

(vi) CPD: rank-3

Figure 2.22: Estimates from the LROAT and CPD models.



Shrinkage, λ=5, -ll(sol)=901
iter=391, time=1333.3805 s

(i)

Shrinkage, λ=50, -ll(sol)=893
iter=388, time=1325.8503 s

(ii)

Shrinkage, λ=500, -ll(sol)=845
iter=366, time=1410.72 s

(iii)

Shrinkage, λ=5000, -ll(sol)=2674
iter=323, time=1282.9599 s

(iv)

Shrinkage, λ=10000, -ll(sol)=11246
iter=308, time=1208.4542 s

(v)

Shrinkage, λ=50000, -ll(sol)=229305
iter=305, time=1394.1945 s

(vi)

Figure 2.23: Estimates from the regLROAT model.

2.4 Real Data Analysis

In Section 2.1, we described a scenario in which the proposed methods could be applied to fMRI data collected from two groups of subjects performing the same task, with the goal of identifying brain regions that help discriminate between healthy subjects and subjects diagnosed with a disease or disorder. Another application of the proposed methods is to fMRI data collected from the same subject performing different tasks, with the goal of identifying the brain regions involved for one task versus another. The Visual Object Recognition (VOR) data (Haxby et al., 2001; Hanson et al., 2004; O'Toole et al., 2005) is freely available through the OpenNeuro project (openneuro.org) and contains fMRI data for six subjects as they viewed different types of images (viewing each type of image is considered a task). There were eight different types of images: bottles, scissors, shoes, chairs, houses, cats, faces, and scrambled images. Data were recorded for each subject for 12 runs. For each run, the image types were shown sequentially in blocks of 24 s duration with 12 s of rest between blocks of different image type. Within a block, images of the same type were shown every 2 s (images of the same type includes different objects of the same type and different views of the same object).

We analyze the data from subject 1. The data for one subject for a single run consists of two parts: 1) an events file giving the time of onset of an image and the type of image and 2) a $40 \times 64 \times 64 \times 121$ fMRI tensor containing raw BOLD signal. The $40 \times 64 \times 64$ component of the fMRI tensor represents the spatial component with pixel dimensions $3.50\ mm \ \times\ 3.75\ mm \ \times\ 3.75\ mm$. The component of length 121 represents the time component with pixel dimension 2.5 s. For each block of image type, we extract the time slices corresponding to 6 s after the onset of the first image (to account for the delay in the hemodynamic response) up to the onset of the last image in the block. Because the resolution of the time component of the fMRI tensor does not match up exactly with the duration of a block, the number of time slices we obtain for a block varies slightly across image types (±1 slice). For each run and image type, we set aside one time slice as part of a test dataset.

Haxby et al. (2001) showed that the VOR data can be used for visual stimulus decoding. In encoding step of visual stimulus decoding, one uses fMRI images to train a classification model in which the fMRI data serve as the predictor and the type of image being viewed serves as the response. For the decoding step, one uses the model to predict the type of image being viewed for a separate test set of fMRI images. We apply the regLROAT, LROAT, and CPD models to the VOR data for binary visual stimulus decoding. We use a logistic regression model with the indicator of image type as the response and the subset of the fMRI tensor corresponding to the visual cortex as the predictor. The visual cortex is located in the posterior part of the brain. By visualizing the whole-brain fMRI image, we determined that indices 1–40 along the first spatial dimension, indices 9–30 along the second spatial dimension, and indices 23–48 along the third spatial dimension should conservatively capture the visual cortex. The size of the resulting image was $40 \times 22 \times 26$. For each pair of image types, we obtained approximately $N = 134$ time slices for the training data (approximately 67 time slices for each image type) and 24 time slices for the test data (exactly 12 time slices for each image type).

We focus on two kinds of classification tasks: scrambled images vs. all others and shoe vs. bottle. We compare scrambled images against all others because the signal should be strongest for these comparisons, and so they serve as good examples to illustrate the uses of the proposed methods. We compare shoes vs. bottles because it was one of the most challenging tasks among the many combinations we tried, and so it serves as a good example to compare the performances of the regLROAT, LROAT, and CPD models. For LROAT and CPD, we are restricted to fitting rank-1 models. The limited sample size relative to the dimension of the fMRI tensor precludes fitting higher rank models. For regLROAT, we fit the model for a sequence of values of the tuning parameter. For each type of image comparison, we report the classification accuracy achieved for the test dataset. We classified an image as 1 if $\hat{p} > 0.5$ and 0 if $\hat{p} < 0.5$, where $\hat{p}$ is defined as

$$\hat{p} = \frac{1}{1 + \exp\left[-\hat{\beta}_0 - \langle \mathcal{X}_i, \hat{\mathcal{B}} \rangle\right]}.$$

The test dataset classification accuracies achieved by each model are shown in Table 2.2. Using the fMRI tensor as input, all three models were able to predict which type of image was being viewed better than chance. The LROAT and regLROAT models performed uniformly better than the CPD model, and sometimes substantially better. The LROAT and regLROAT models performed equally well for many tasks, and when they differed, the regLROAT model outperformed the LROAT model. The regLROAT model was able to achieve 100% classification accuracy for at least one value of the tuning parameter for all classification tasks but one. The most challenging task was discriminating between shoe vs. bottle. Predictions based on the CPD model were barely better than chance, and the classification accuracy for the LROAT model was the lowest among all tasks tried. The regLROAT model performed better than the LROAT model, but it did not achieve 100% classification accuracy for any value of the tuning parameter. Note that the rank of the estimated parameter tensor under the regLROAT model was often larger than 1.

Table 2.2: Test dataset classification accuracy (%) for the CPD, LROAT, and regLROAT models applied to the VOR data.

	CPD	**LROAT**	**regLROAT**				
Task	(rank-1)	(rank-1)	$\lambda = 50$	$\lambda = 100$	$\lambda = 500$	$\lambda = 1000$	$\lambda = 2000$
scrambled vs. face	100	100	100 (rank-4)	100 (rank-2)	100 (rank-2)	100 (rank-2)	100 (rank-1)
scrambled vs. cat	87.5	100	100 (rank-4)	100 (rank-3)	100 (rank-2)	100 (rank-2)	95.83 (rank-1)
scrambled vs. house	91.67	100	100 (rank-4)	100 (rank-3)	100 (rank-3)	100 (rank-2)	100 (rank-1)
scrambled vs. house* (larger)	79.17	100	100 (rank-5)	100 (rank-4)	100 (rank-4)	100 (rank-3)	100 (rank-2)
scrambled vs. bottle	79.17	95.83	100 (rank-6)	95.83 (rank-4)	95.83 (rank-4)	95.83 (rank-2)	95.83 (rank-2)
scrambled vs. scissors	95.83	100	100 (rank-6)	100 (rank-4)	100 (rank-4)	100 (rank-3)	95.83 (rank-2)
scrambled vs. chair	95.83	95.53	100 (rank-4)	100 (rank-3)	100 (rank-3)	100 (rank-3)	100 (rank-2)
scrambled vs. shoe	87.5	100	100 (rank-6)	100 (rank-3)	100 (rank-3)	95.83 (rank-2)	95.83 (rank-2)
shoe vs. bottle	66.67	79.17	95.83 (rank-9)	91.67 (rank-5)	95.83 (rank-3)	95.83 (rank-3)	95.83 (rank-2)

*The size of the fMRI tensor was increased by including indices 9–40 along the second spatial dimension.

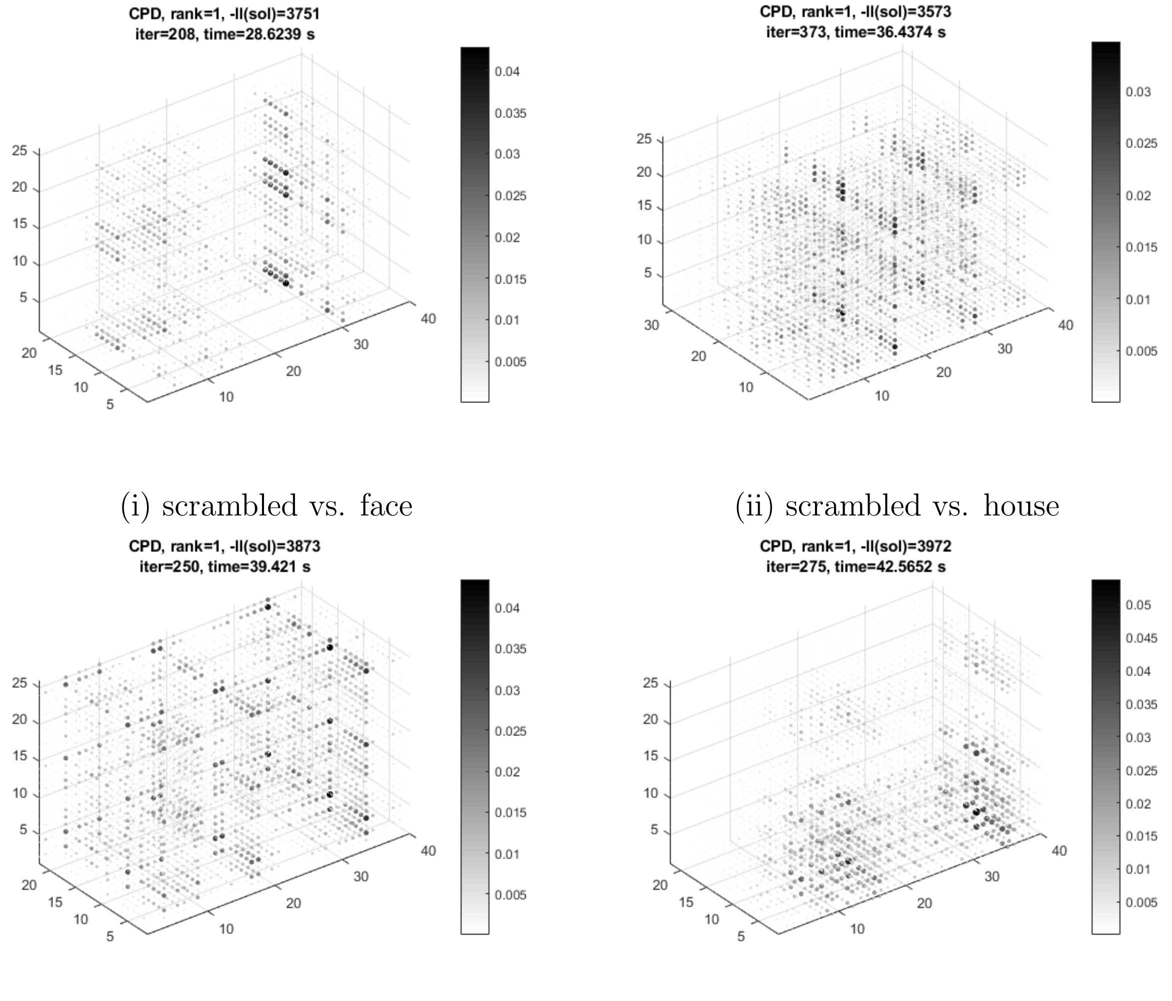
In addition to making accurate predictions, another goal of the proposed methods is to understand what components of the image are most related to the response. For visual stimulus decoding, this can be interpreted as understanding what areas of the brain are responsible for discriminating between one type of viewed image and another, as measured by the BOLD response. Figures 2.24 – 2.27 show the absolute values of the estimated parameter tensors (so that white corresponds to zero) from the CPD, LROAT, and regLROAT models for selected tasks. For scrambled images vs. house, we expanded the indices along the second spatial dimension from 9–30 to 9–40 to capture the signal region better. The regLROAT model estimates are shown for two values of the tuning parameter, representing different extremes with respect to the amount of shrinkage. Note that the colorbars for the regLROAT model estimates are on a much smaller scale than those for the rank-1 LROAT and CPD model estimates, reflecting the powerful effect of shrinkage. The size of the dots in the regLROAT estimates have been magnified to a comparable level as the LROAT and CPD estimates to enable easier identification of regions with (relatively) large values.

In all of the figures, regions of the estimated tensor with large magnitude values (relative to other regions within the same estimate) can be interpreted as signal regions – i.e., regions that are strongly associated with the type of image being viewed. The CPD estimates (Fig. 2.24) are noisy, with the largest values occurring in scattered voxels rather than in spatially compact locations. Such estimates are difficult to interpret because no clearly defined regions stand out. Although the CPD estimates are less interpretable, the classification accuracy for the test dataset was high for two tasks (scrambled vs. face and scrambled vs. scissors) and better than chance for the others.

The LROAT estimates (Fig. 2.25) are less noisy than the CPD estimates, especially for the scrambled vs. face [Fig. 2.25(i)] and scrambled vs. house [Fig. 2.25(ii)]. For those, distinct, spatially compact regions of the estimate stand out, suggesting that the corresponding region of the brain is relevant to the task. For scrambled vs. bottle [Fig. 2.25(iii)] and scrambled vs. scissors [Fig. 2.25(iv)], the estimates are easier to interpret than the CPD estimates, but not as easy to interpret as for the other two tasks. The regions with relatively large values are diffuse rather than spatially compact, making it more difficult to identify a well-defined brain region that might be relevant to the task.

The regLROAT estimates (Figs. 2.26 and 2.27) are easiest to interpret and become more interpretable with less shrinkage. The same regions that stood out in the rank-1 LROAT estimates for scrambled vs. face and scrambled vs. house stand out in the regLROAT estimates as well. The most improvement occurs for the scrambled vs. bottle and scrambled vs. scissors tasks under less shrinkage [Fig. 2.27(iii) and (iv)]. Although certain regions of the estimates stand out to some extent with greater shrinkage [Fig. 2.26(iii) and (iv)], they are much more well-defined with less shrinkage. From Table 2.2, the scrambled vs.

bottle estimate was rank-2 with greater shrinkage ($\lambda = 1000$) and rank-6 with less shrinkage ($\lambda = 50$). The estimate with greater shrinkage closely resembles the rank-1 LROAT estimate, suggesting that a higher rank estimate is needed to capture the main features associated with the scrambled vs. bottle task.



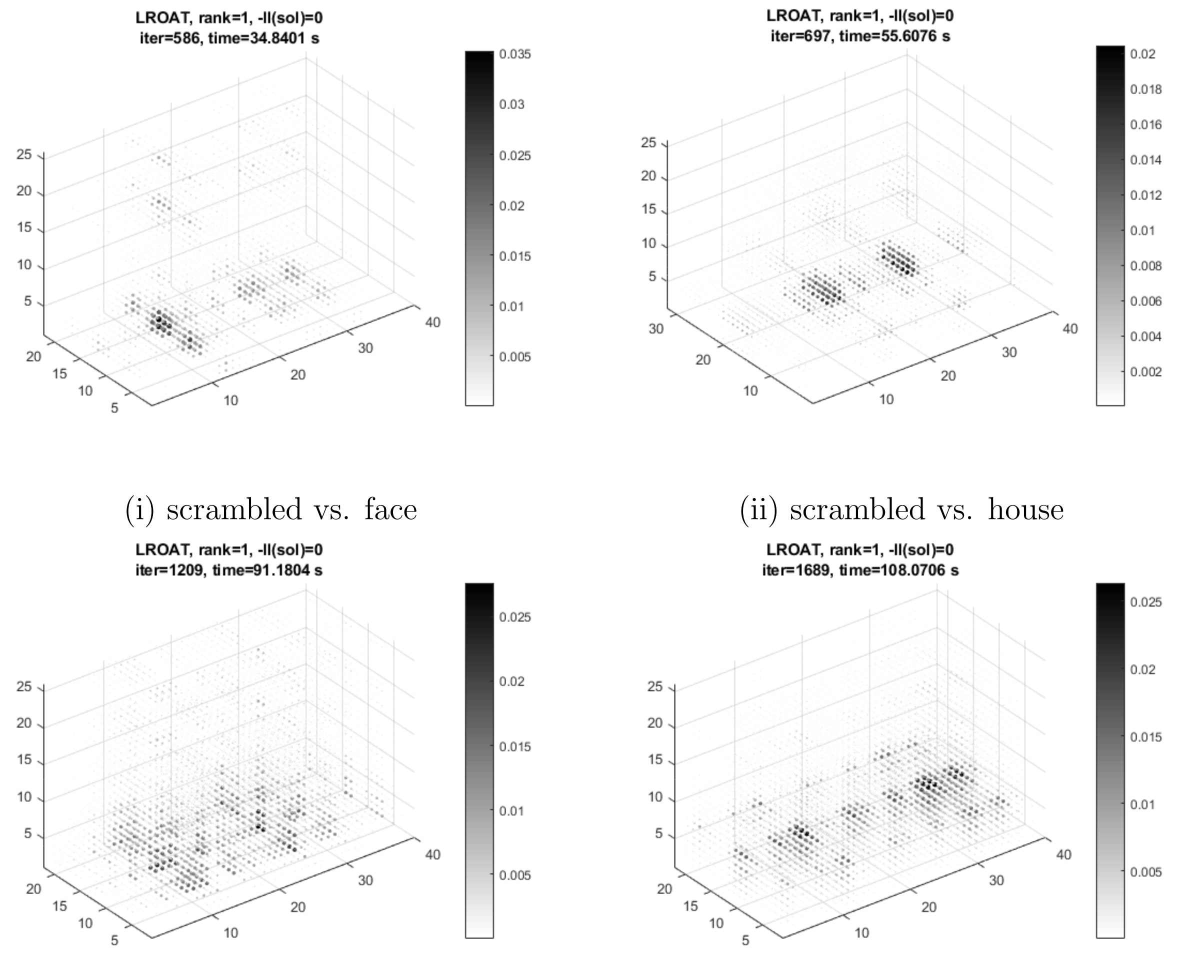
(i) scrambled vs. face

(ii) scrambled vs. house

(iii) scrambled vs. bottle

(iv) scrambled vs. scissors

Figure 2.24: Estimated parameter tensor from the rank-1 CPD model for selected tasks.


LROAT, rank=1, -ll(sol)=0
iter=586, time=34.8401 s

LROAT, rank=1, -ll(sol)=0
iter=697, time=55.6076 s

LROAT, rank=1, -ll(sol)=0
iter=1209, time=91.1804 s

LROAT, rank=1, -ll(sol)=0
iter=1689, time=108.0706 s

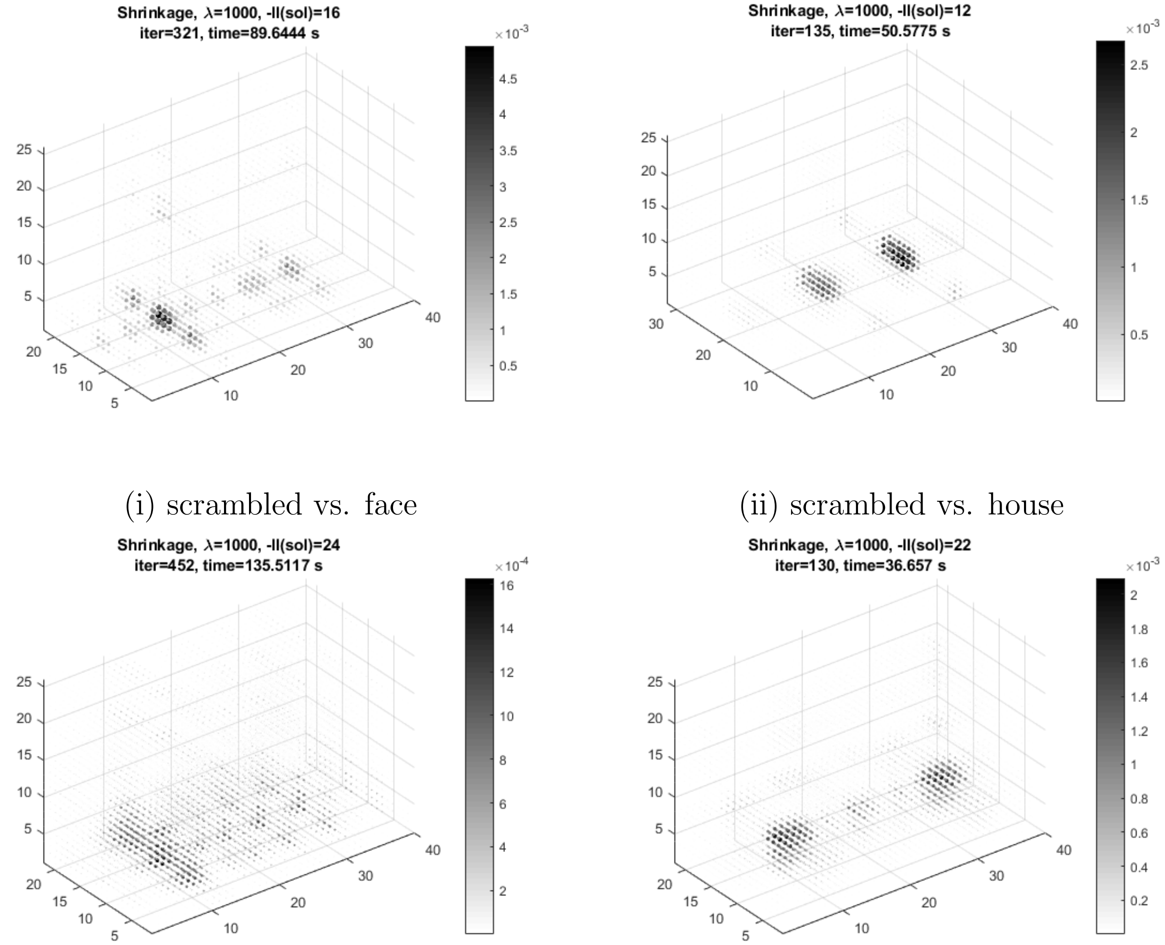
(i) scrambled vs. face

(ii) scrambled vs. house

(iii) scrambled vs. bottle

(iv) scrambled vs. scissors

Figure 2.25: Estimated parameter tensor from the rank-1 LROAT model for selected tasks.



Shrinkage, λ=1000, -ll(sol)=16
iter=321, time=89.6444 s

Shrinkage, λ=1000, -ll(sol)=12
iter=135, time=50.5775 s

Shrinkage, λ=1000, -ll(sol)=24
iter=452, time=135.5117 s

Shrinkage, λ=1000, -ll(sol)=22
iter=130, time=36.657 s

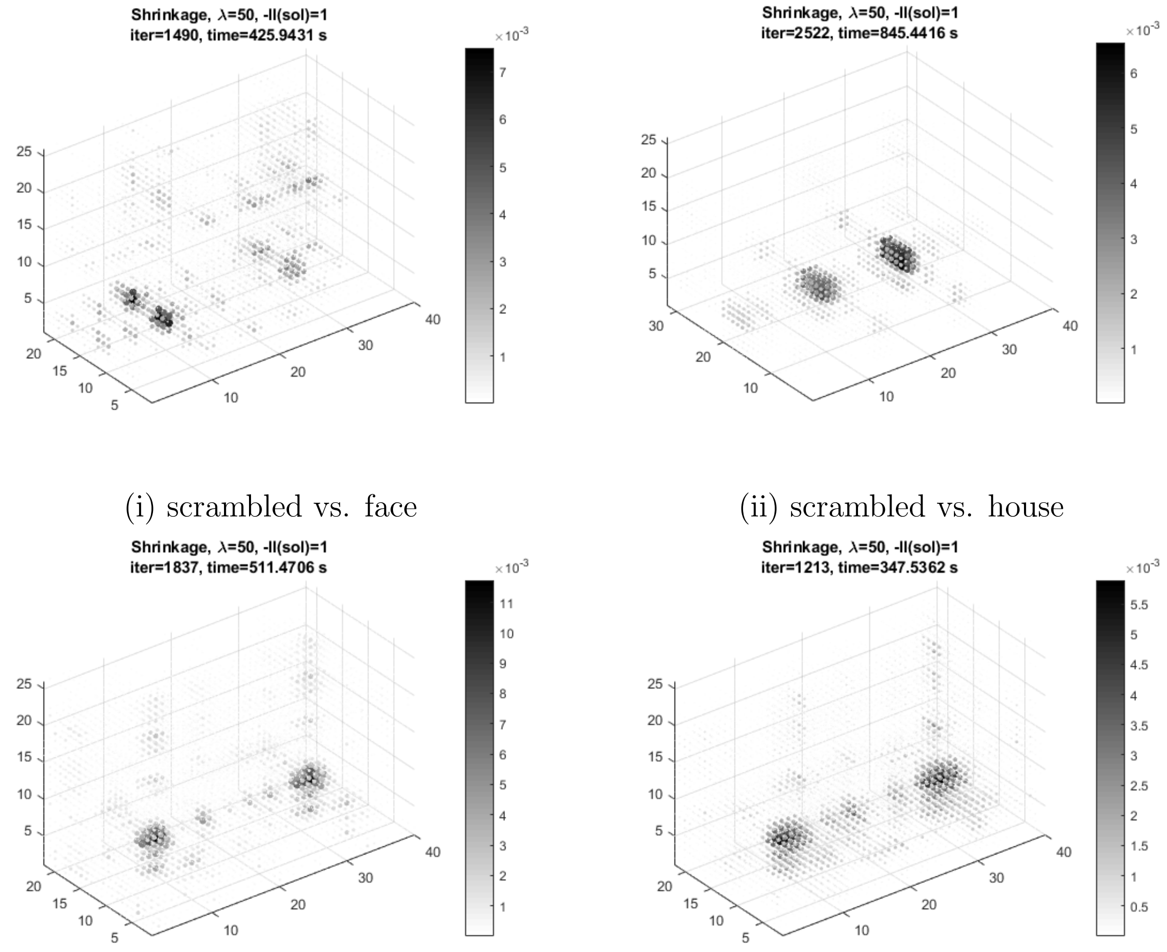
(i) scrambled vs. face

(ii) scrambled vs. house

(iii) scrambled vs. bottle

(iv) scrambled vs. scissors

Figure 2.26: Estimated parameter tensor from the regLROAT model ($\lambda = 1000$) for selected tasks.



(i) scrambled vs. face

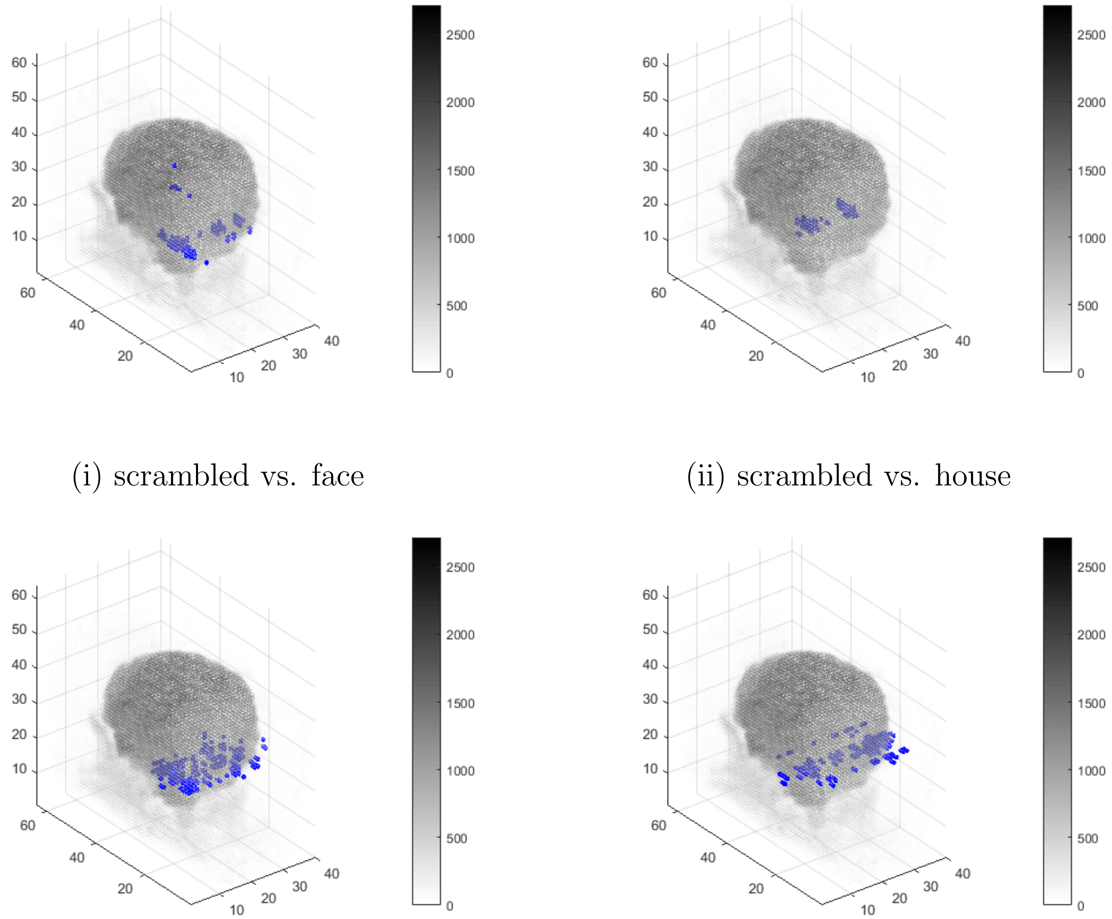
(ii) scrambled vs. house

(iii) scrambled vs. bottle

(iv) scrambled vs. scissors

Figure 2.27: Estimated parameter tensor from the regLROAT model ($\lambda = 50$) for selected tasks.

In Figure 2.28, we plot a randomly selected whole-brain fMRI image and overlay blue points for voxels in which the magnitude of the estimated parameter tensor from the rank-1 LROAT model exceeded 0.01. The 0.01 threshold was chosen for visualization purposes, and the estimate from the rank-1 LROAT model was chosen as a representative example. Our goal is to show where the strongest signal regions are located on a more complete picture of the brain. One noticeable feature of the images in Figure 2.28 is that some of the voxels with large magnitude values are actually located outside the brain region. This can be attributed to an artifact caused by fitting a rank-1 model. From the simulation experiment in Section 2.3, many of the rank-1 estimates had large values outside of the true signal region [see the rank-1 estimates in Figs. 2.11(i)–(ii) or 2.22(i)–(ii) for good examples]. These extraneous values were typically contained within indices associated with the true signal region along one dimension, but occurred outside of indices associated with the signal region along one or more of the other dimensions. Besides the few stray voxels located outside the brain, the other noticeable feature in Figures 2.25 and 2.28 is that many of the voxels with large values are grouped together in spatially compact regions. The regions selected in Figure 2.28(i) and (ii) appear to correspond roughly to the fusiform face area and parahippocampal place area, respectively, while the regions selected in Figure 2.28(iii) and (iv) seem to more closely correspond to the lateral occipital complex (although the selected regions are more diffuse). Note that expert judgment would be necessary to confirm that assessment. The fusiform face area, parahippocampal place area, and lateral occipital complex are known to respond to visual stimuli from faces, places, and things, respectively, which is consistent with tasks shown.



(i) scrambled vs. face

(ii) scrambled vs. house

(iii) scrambled vs. bottle

(iv) scrambled vs. scissors

Figure 2.28: Randomly selected fMRI image with blue points indicating voxels for which the magnitude of the estimated parameter tensor exceeded 0.01.

2.5 Discussion

Using a real fMRI dataset, we demonstrated that the proposed LROAT and regLROAT models have the potential to perform better than existing methods both with respect to predictive performance and interpretation. Because the LROAT model is a special case of the CPD model to which we compared (and the two are in fact equivalent in some instances), the superior performance of the LROAT model can be attributed to some extent to the estimation algorithm rather than the model itself. Particularly in the real data analysis (Section 2.4), the projected gradient descent algorithm used to fit the LROAT model yielded parameter estimates that achieved smaller values of the objective function and were easier to interpret from the biological perspective. In contrast, the alternating minimization algorithm used to fit the CPD model yielded parameter estimates that were barely distinguishable from noise. Both algorithms used the same tolerance criterion and tolerance value, so the differences cannot be attributed to differences in the stopping rule. One possible explanation for the poor performance of the alternating minimization algorithm is that alternating among parameters makes the algorithm more likely to find local optima, especially when the number of parameters is large relative to the sample size. The projected gradient descent algorithm updates all parameters at once, perhaps making it less likely to get stuck in a local optimum.

Even though the parameter estimates from the LROAT and regLROAT models were easier to interpret than the solutions from the CPD model, the interpretations of the parameter estimates were not without difficulty. The examples in Figure 2.28 show that many voxels with large magnitude values were located outside of the brain region. Although one can easily recognize estimated signals outside of the brain region as spurious, their presence casts doubt on the findings inside of the brain region. That is, in the presence of substantial noise, how can one identify regions that are likely to be real signal? Such a question is an inherent aspect of any estimation problem. However, for the proposed methodology, part of the noise in the parameter estimates originates as an artifact of the estimation algorithm. Thus, imaging data analysis poses additional challenges beyond what is in encountered in traditional linear models, where analytic solutions for the parameters are available and the only source of spurious signal is from sampling variability (given that other model assumptions are met). Consequently, the proposed methods are perhaps best used as an exploratory device rather than a tool for hypothesis testing [though hypothesis testing would be possible using the results of Zhou et al. (2013)]. Indeed, the key strength of the methodology is the ability to visualize the estimated parameter tensor, which is more in line with the goals of exploratory data analysis than formal hypothesis testing.

Another issue related to the interpretation of the estimated parameters from both the LROAT and regLROAT models is that the estimated parameters are not sparse. Although the singular values of the estimated parameter tensor are shrunk toward zero in the regLROAT model, none of the elements of the estimated parameter tensor are exactly zero. This poses a challenge for interpretation because one must decide subjectively how large the magnitudes of the estimates must be before a region of the image is considered "important." In Figure 2.28, an arbitrary threshold was chosen such that the estimates overlain on an original image would be easy to visualize. However, methods that obviate the need to choose subjective thresholds, such as those that produce sparse estimates, would be preferred. One way to employ sparse estimation in a low rank tensor model is to penalize the elements of the vectors comprising each of the rank-1 tensors in the tensor decomposition. Zhou et al. (2013) used this approach to develop a regularized version of the CPD model. For the proposed LROAT and regLROAT models, the approach would be challenging to implement because the algorithm used to fit such a model would need to simultaneously maintain sparsity and orthogonality of the factor matrices in each update. If an algorithm could be devised, a sparsity penalty on the elements of the vectors in the assumed tensor decomposition may not be the best way to achieve sparsity within the estimated parameter tensor. In our experience, the regularized version of Zhou et al.'s CPD model was highly sensitive to the initial value when the model was applied to real data (results not shown). For several different starting values, the non-sparse regions of the resulting estimates were rarely in the same locations, making the estimates useless for interpretation. Note, however, that the non-regularized version of their model was also sensitive to the initial value, so it is unclear whether the observed behavior in the estimates was mainly due to the alternating minimization algorithm or the approach used to achieve sparsity.

One limitation of the proposed methods, and any other methods that aim to exploit low rank tensor decompositions, is that the rank depends on both the orientation and resolution of the images. For example, for the butterfly image in Figure 2.2(ii), if the butterfly were rotated such that the image was symmetric across one of the axes, then the rank would be less than it is under the orientation shown. Similarly, the 64×64 triangle image in Figure 2.2(i) is rank-13, but becomes rank-9 if the image is downsized to 32×32. Although the 64×64 and 32×32 images look the same, a higher rank tensor model might be needed to provide an adequate approximation to the 64×64 image. One solution to the orientation problem is to try several different rotations and choose one that balances adequate fit with low rank. However, this approach would be computationally demanding for 2D images and impractical for 3D and higher dimensional images. For the resolution problem, one may downsize the original images. Downsizing was used in Zhou et al. (2013) and Li et al. (2018) in their real data applications out of necessity because, even under a low rank assumption, the number of parameters to estimate exceeded the sample size. Although downsizing may provide a suitable solution in particular cases, it should generally be avoided because it is not clear how to quantify whether or how much information is lost for estimating the parameter tensor after the downsizing.

In future work we hope to explore the convergence properties of the algorithms we have proposed for fitting the LROAT and regLROAT models. Empirically, the algorithms appear to converge based on the simulated and real examples we have tried. Moreover, the solutions are frequently better than the solutions from the alternating minimization algorithm of Zhou et al. (2013). Unfortunately, there are myriad challenges for proving global convergence for tensors of order ≥ 3. One challenge is that both of the algorithms we have proposed rely on Chen and Saad's (2009) algorithm for the LROAT decomposition (Algorithm 2). They were unable to prove global convergence of their algorithm. In addition, their algorithm is not guaranteed to converge to the global optimum of problem (2.3). Thus, the projection steps in the proposed Algorithms 3 and 4 are not necessarily carried out exactly. That creates a difficulty for applying any known results about the convergence of projected gradient descent with non-convex constraints. Another challenge is in dealing with the non-convex constraints imposed by the low rank and orthogonality assumptions. Some results regarding the convergence of projected gradient descent are known in the context of low rank matrices [see Jain and Kar (2017)] and have been applied in the context of the HOSVD for tensors [e.g., Chen et al. (2019)]. However, given that not all tensors admit an orthogonal decomposition, it is not clear how one might extend those results to the LROAT decomposition utilized in Algorithms 3 and 4. Finally, a major challenge in proving convergence of Algorithm 4 for the regLROAT model is that it is neither a projected gradient descent algorithm nor a proximal gradient algorithm in the strict sense, but rather a mixture of the two. Although superficially similar to Zhou and Li's (2014) algorithm for spectral-regularized matrix regression, Algorithm 4 involves an additional projection step onto the set of orthogonally-decomposable tensors. For matrices, this is simply the SVD and is a change of basis rather than a projection. For tensors, it is a non-convex projection for all but the rare cases in which the tensor already admits an orthogonal decomposition. Since Algorithm 4 adds a non-trivial modification to the standard proximal gradient method, it may not be possible to apply any known convergence results for the proximal gradient method.

Chapter 3

A Sparse, Low Rank Matrix Approximation with Application to Variable Selection in High-Dimensional Canonical Correlation Analysis

3.1 Introduction

Much of modern scientific research aims to characterize the associations among high-dimensional, multimodal data. That is, researchers measure many variables representing different, often complementary, sources of information with the goal of performing an integrated analysis that combines information across all of the sources. For example, in epigenetics, one might want to understand how methylation of CpG sites is associated with the expression of genes related to breast cancer (Holm et al., 2010). In a model of ischemic stroke, one might want to know how magnetic resonance imaging (MRI) measurements obtained shortly after stroke are related to short- and long-term recovery patterns in behavior, cognition, and mobility (Platt et al., 2014; Duberstein et al., 2014). In the Human Connectome Project (humanconnectomeproject.org), researchers hope to map the anatomical and functional connectivity of the brain by integrating information obtained by structural MRI, functional MRI, diffusion MRI, and other imaging modalities. In each example, not only are data obtained from different modalities, but also within a modality, the number of variables measured often exceeds the number of subjects. In the breast cancer study by Holm et al. (2010), the methylation levels at 1452 CpG sites and gene expressions for 511 genes were obtained from 179 samples. Thus, the total number of variables measured exceeded the sample size by an order of magnitude.

We propose to study the association between two high-dimensional sets of variables representing different modalities. Suppose the data consist of p variables from one modality and q variables from the other modality measured from a common set of n subjects or experimental units. Denote the data from the first modality as $X_{n\times p}$ and the data from the second modality as $Y_{n\times q}$. We can construct a $p \times q$ nonsymmetric matrix K containing the pairwise associations between the variables in X and the variables in Y. In our work, we focus on the Pearson correlation between X and Y, but in principle, any suitable measure of association could be used. If $\widehat{\Sigma}_{XX}$, $\widehat{\Sigma}_{YY}$, and $\widehat{\Sigma}_{XY}$ denote the sample within- and between-covariance matrices of X and Y, then K can be constructed as $K := [\text{diag}(\widehat{\Sigma}_{XX})]^{-1/2}\ \widehat{\Sigma}_{XY}\ [\text{diag}(\widehat{\Sigma}_{YY})]^{-1/2}$. When all of the variables within each dataset are measured in the same units (but variables between datasets needn't be in the same units), it may be desirable to use the sample covariance as a measure of association rather than the correlation, so that $K := \widehat{\Sigma}_{XY}$.

Our analysis has two primary goals: (1) to describe the dominant modes of co-variation between X and Y and (2) to select variables from X and Y that contribute most to the dominant modes of co-variation. To achieve these goals, we approximate K by a matrix W that is low rank and sparse. By finding a low rank matrix close to K, we reduce the pq pairwise associations to a few modes that capture the most prominent features describing the overall association between X and Y. We use sparsity to select the most important variables contributing to those features. For high-dimensional data, variable selection is an essential tool for improving the interpretation of the analysis.

Our problem falls into a more general class of problems based on nuclear norm regularization. Let $\|W\|_* := \sum_{j=1}^{\min(p,q)} \sigma_j(W)$ denote the nuclear norm, where $\sigma(\cdot)$ is an operator that extracts the singular values of a matrix. Then our problem is a special case of the generic class of problems

$$\min_W f(W;\ \text{data}) + \lambda_1 \|W\|_* + \lambda_2 P(W),$$

where f is a suitable loss function, P is sparsity-inducing penalty function, λ_1 and λ_2 are tuning parameters, and "data" is either an observed matrix to be approximated by W (in our problem, the matrix K) or additional information, depending on the application. The nuclear norm can be expressed as $\|W\|_* = \sum_{j=1}^{\min(p,q)} |\sigma_j(W)| = \|\sigma(W)\|_1$. Since $\text{rank}(W) = \|\sigma(W)\|_0$, the nuclear norm can be viewed as a convex relaxation of the rank. Penalizing the nuclear norm shrinks some of the singular values to exactly zero, yielding a low rank matrix. The choice of sparsity penalty P depends on the goals of the analysis, and in some cases it may be omitted. In our problem, we choose a penalty function that results in variable selection for one of the datasets X or Y.

We are motivated to employ nuclear norm regularization because of its successful application in other problems involving low rank matrix approximations. An example of one such problem is regression with matrix-valued covariates (Zhou and Li, 2014). For that problem, f is the negative log-likelihood of a generalized linear model (GLM) and the data are the pairs $(y_i,\ X_i),\ i = 1, \ldots, n$, where the y_i's are scalar-valued responses and the X_i's are matrix-valued covariates. The model assumes the link function $g\left(E[Y_i]\right) = \langle X_i, W\rangle = \sum_{j,k} x_{ijk} w_{jk}$, where W is a matrix-valued parameter corresponding to the covariates X. Because of the large number of parameters involved, it is necessary to find a regularized estimate of W. Zhou and Li argued that regularization of the rank can be more appropriate in certain contexts (e.g., neuroimaging) than sparse regularization. Thus, they penalized the objective function of the GLM with the nuclear norm of W to find a low rank estimate. They showed that the nuclear norm-regularized objective function is convex, and thus easier to optimize, than a similar non-convex version of the problem based on the rank-R approximation of W (Zhou et al., 2013). In addition, they showed that the effective degrees of freedom (Efron, 2004) of the nuclear norm-regularized model is always dominated by the naive count of the number of parameters. That is, a nuclear norm-regularized model achieves greater shrinkage than a fixed rank-R model, potentially providing a better low rank estimate of W in some situations.

In Zhou and Li's regularized matrix model, the parameter W is unobserved, so their problem is an example of simultaneous estimation and regularization. In other applications, the data is an observed matrix K, so the goal is reduced from simultaneous estimation and regularization to finding a good low rank approximation of K. In latent variable graphical models (Chandrasekaran et al., 2012) and robust principal components analysis (Candès et al., 2011), one assumes the observed matrix K can be decomposed as $K = L + S$, where L is low rank and S is sparse. Then the problem can be written as

$$\min_{L,S} \|K - (L + S)\|_F^2 + \lambda_1 \|L\|_* + \lambda_2 \|S\|_1,$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\|S\|_1$ is interpreted as the LASSO penalty (Tibshirani, 1996) applied element-wise to the matrix S. The constants λ_1 and λ_2 are tuning parameters controlling the rank of L and the level of sparsity of S, respectively. In latent variable graphical models, the marginal precision matrix K is decomposed into the conditional precision matrix of the observed variables S and another matrix L representing the effect of marginalizing over the latent variables; the former is assumed to be sparse, while the latter is assumed to be low rank. In robust PCA, the sparse matrix S serves to absorb the effects of outliers when estimating the principal components contained in L. Note that in either application, the sum $L + S$ is not necessarily sparse or low rank.

Rather than decomposing a matrix as the sum of a sparse matrix and a low rank matrix, one may want to find an approximation to the matrix that is simultaneously sparse and low rank. Then the problem can be written as

$$\min_{W} \|K - W\|_F^2 + \lambda_1 \|W\|_* + \lambda_2 \|W\|_1. \tag{3.1}$$

This version of the sparse and low rank approximation problem has been proposed for denoising applications, such as covariance matrix estimation with small sample size and network estimation when the observed adjacency matrix may have irrelevant or missing edges (Richard et al., 2012).

We formally state the optimization problem we propose to solve in Section 3.2.1. The problem we propose shares a close connection with recent proposals for sparse canonical correlation analysis (CCA). In Section 3.2.2, we introduce sparse CCA and explore its connection to our problem in depth. In particular, we show that, under some assumptions, our problem and sparse CCA aim to select the same subsets of variables from X and Y. In Section 3.2.3, we describe an algorithm based on the alternating direction method of multipliers (ADMM) to solve the proposed problem. In Section 3.2.4, we address tuning parameter selection. Empirically, tuning parameter selection has proven difficult for complex optimization problems such as sparse CCA because of its instability. As an alternative, we propose to rank the variables according to their importance by defining metrics that combine information across the estimates obtained for every combination of the tuning parameters. In Section 3.3, we compare the proposed methods to the sparse CCA methods of Witten et al. (2009) and Safo et al. (2018) through a simulation experiment, where we measure the performance of each method by its variable selection accuracy. In Section 3.4, we apply the proposed method to the breast cancer data from Holm et al. (2010). We provide some concluding remarks in Section 3.5.

3.2 Proposed Methodology

3.2.1 Problem Statement

Let K denote the $p \times q$ matrix containing the pairwise Pearson correlations between the variables in X and the variables in Y. The discussion generalizes to other measures of association. We approximate K by a matrix W that is low rank and has row or column sparsity. Let $\boldsymbol{w}_r,\ r = 1, \ldots, p$, denote the rows of W and $\boldsymbol{w}_c,\ c = 1, \ldots, q$, denote the columns. We solve the problems:

$$\widehat{W}_r = \underset{W}{\arg\min}\ \|K - W\|_F^2 + \lambda_1 \|W\|_* + \lambda_2 \sum_{r=1}^{p} \|\boldsymbol{w}_r\|_2, \tag{3.2}$$

$$\widehat{W}_c = \underset{W}{\arg\min}\ \|K - W\|_F^2 + \lambda_1 \|W\|_* + \lambda_3 \sum_{c=1}^{q} \|\boldsymbol{w}_c\|_2. \tag{3.3}$$

As in other applications, penalizing the nuclear norm shrinks some of the singular values to exactly zero, yielding a low rank approximation of K. The function $\sum \|\cdot\|_2$ is known as the group LASSO penalty (Yuan and Lin, 2006), and in our application causes entire rows or columns to be estimated as exactly zero. Since each row or column of K corresponds to a variable in X or Y, shrinking entire rows or columns to zero achieves variable selection. Thus, when applied together, the nuclear norm penalty and group LASSO penalty enable us to estimate the dominant modes of co-variation characterizing the association between X and Y while simultaneously selecting a subset of variables that contribute most to those modes of co-variation.

Remark 1. In problems (3.2) and (3.3), only the group LASSO penalty affects the sparsity of the estimate, but both the nuclear norm and group LASSO penalties affect the rank. The group LASSO penalty affects the rank because rank($\widehat{W}$) $\leq$ min(# nonzero rows, # nonzero columns). Consequently, we could omit the nuclear norm penalty and find a sparse, low rank approximation of K using the group LASSO penalty alone. However, such an estimate *would not* capture the dominant modes of co-variation between X and Y. For example, if we set $\lambda_1 = 0$ and solve problem (3.2), we would select the variables in X that individually have the highest correlation with all of the variables in Y, as measured by the ℓ_2 norm. In addition, rank($\widehat{W_r}$) $=$ min(n, # nonzero rows) *w.p.* 1. In contrast, if we let $\lambda_1 \neq 0$ and solve problem (3.2), we not only select variables in X that have high correlation with the variables in Y, but also describe the nature of the association in a few dominant modes; that is, the estimate captures the association between Y and the selected variables from X holistically rather than individually.

Remark 2. The proposed problems are similar to problem (3.1) from Section 3.1. Although both problems entail sparse and low rank approximations, our motivations for and formulation of problems (3.2) and (3.3) are distinct from those of problem (3.1). In problem (3.1), the estimate $\widehat{W}$ is low rank and element-wise sparse. In our context of finding a low rank, sparse approximation of the sample correlation matrix, such an estimate would be interpreted as characterizing the dominant modes of co-variation between X and Y while estimating some of the pairwise correlations between variables in X and Y as zero. However, we desire to select a subset of variables that contribute most to the dominant modes of co-variation. To achieve this objective, entire rows or columns of $\widehat{W}$ must be sparse. Thus, we penalize the objective function with the group LASSO rather than the standard LASSO. We also note that, from a computational perspective, problems (3.2) and (3.3) are more challenging than problem (3.1). Although there is some interplay between the LASSO penalty and nuclear norm penalty, the interplay between the group LASSO penalty and nuclear norm penalty is much stronger. For example, after solving problem (3.1), we could in principle obtain an estimate that was sparse except for $\min(p, q)$ elements, but had the same rank as the original matrix K.

Remark 3. As an alternative to problems (3.2) and (3.3), we could solve a single optimization problem that selects the variables from X and Y at the same time. For example, we could solve the problem:

$$\widehat{W} = \arg\min_{W} \ \|K - W\|_F^2 + \lambda_1 \|W\|_* + \lambda_2 \sum_{r=1}^{p} \|\boldsymbol{w}_r\|_2 + \lambda_3 \sum_{c=1}^{q} \|\boldsymbol{w}_c\|_2. \tag{3.4}$$

However, by penalizing both the rows and columns, we penalize each element of W twice. In addition, all three penalties affect the rank of the estimate, making the computational problem even more challenging. Thus, we consider selecting the variables from X and Y as separate problems and solve (3.2) independently of (3.3).

3.2.2 Application to Variable Selection in High-Dimensional CCA

Canonical correlation analysis (CCA) (Hotelling, 1936) seeks to characterize the relationship between a p-dimensional set of continuous variables X and a q-dimensional set of continuous variables Y by finding linear combinations $X\boldsymbol{u}$ and $Y\boldsymbol{v}$ such that the correlation $\rho := Cor(X\boldsymbol{u},\ Y\boldsymbol{v})$ is maximized. Let $Var(X) := \Sigma_{XX}$, $Var(Y) := \Sigma_{YY}$, and $Cov(X, Y) := \Sigma_{XY}$ denote the population within- and between-covariance matrices of X and Y. Then solution to CCA can be obtained from the singular value decomposition (SVD) of the matrix

$$\widetilde{K} \ := \ \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2}.$$

The largest singular value of $\widetilde{K}$ yields the correlation ρ and the corresponding left- and right-singular vectors yield $\Sigma_{XX}^{1/2}\boldsymbol{u}$ and $\Sigma_{YY}^{1/2}\boldsymbol{v}$. We call ρ the canonical correlation and $\boldsymbol{u}, \boldsymbol{v}$ the canonical vectors. In practice, the population covariance matrices are replaced by their sample analogs $\widehat{\Sigma}_{XX}$, $\widehat{\Sigma}_{XY}$, and $\widehat{\Sigma}_{YY}$.

When the dimension p or q exceeds the sample size n, the inverses of one or both of the within-set sample covariances does not exist. A variety of approaches have been implemented to address the invertibility of the within-set covariance matrices, including extracting their diagonals (Witten et al., 2009; Parkhomenko et al., 2009; Chalise and Fridley, 2012; Safo et al., 2018; Jung et al., 2019), making structural assumptions such as Toeplitz (Chen et al., 2013), and adding a ridge correction (Vinod, 1976; Safo et al., 2018). Note that extracting the diagonals of $\widehat{\Sigma}_{XX}$ and $\widehat{\Sigma}_{YY}$ amounts to the SVD of the sample correlation matrix between X and Y and is equivalent to assuming Σ_{XX} and Σ_{YY} are identity for standardized $X_{n\times p}$ and $Y_{n\times q}$ (i.e., each column has mean zero and variance one).

In addition to the computational issues that arise in high dimensional settings, the large number of variables involved also complicates the interpretation of the solution. One can improve the interpretation by selecting a relatively small number of the most relevant variables. Toward this end, many authors have proposed sparse versions of CCA by assuming the canonical vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are sparse. Witten et al. (2009), Parkhomenko et al. (2009), and Chalise and Fridley (2012) develop algorithms that include a soft-thresholding step, which results in some of the elements of $\boldsymbol{u}$ and $\boldsymbol{v}$ estimated as exactly zero. Chen et al. (2013) take a similar approach, but use hard thresholding. Safo et al. (2018) and Jung et al. (2019) consider a more general class of problems known as generalized eigenvalue problems (GEPs), of which CCA is a special case. Safo et al. (2018) begin with a non-sparse estimate of the canonical vectors, such as from the SVD of sample correlation matrix between X and Y, then obtain new estimates via sparse estimation with linear programming (SELP). Jung et al. (2019) modify the orthogonal iteration algorithm (Golub and Van Loan, 1996) for solving GEPs by adding a sparsity-inducing penalty

We now explore the connection between the proposed methodology and certain sparse CCA methods. For a (not necessarily sparse) CCA model with d nonzero canonical correlations, the population covariance for (X, Y) can be written as (Chen et al., 2013)

$$\Sigma = \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix} = \begin{pmatrix} \Sigma_{XX} & \Sigma_{XX} U D V^T \Sigma_{YY} \\ \Sigma_{YY} V D U^T \Sigma_{XX} & \Sigma_{YY} \end{pmatrix},$$

where $D_{d\times d}$ is a diagonal matrix containing the canonical correlations such that $\rho_1 \geq \cdots \geq \rho_d$ and the matrices $U_{p\times d}$, $V_{q\times d}$ contain the canonical vectors. We require $\boldsymbol{u}_j^T \Sigma_{XX} \boldsymbol{u}_j = \boldsymbol{v}_j^T \Sigma_{YY} \boldsymbol{v}_j = 1$ and $\boldsymbol{u}_i^T \Sigma_{XX} \boldsymbol{u}_j = \boldsymbol{v}_i^T \Sigma_{YY} \boldsymbol{v}_j = 0$ $(i \neq j)$. In practice, we are interested in situations in which the largest canonical correlation is nonzero and the rest are relatively small or zero. Thus, for the rest of this section, we focus on the single canonical pair model $(d = 1)$, so that $\Sigma_{XY} = \rho \Sigma_{XX} \boldsymbol{u} \boldsymbol{v}^T \Sigma_{YY}$.

In sparse CCA, we assume the canonical vectors $\boldsymbol{u}$, $\boldsymbol{v}$ are sparse. Without loss of generality, we can assume the first p^{sig} elements of $\boldsymbol{u}$ and the first q^{sig} elements of $\boldsymbol{v}$ are nonzero, with the remaining elements zero. We call the variables corresponding to the nonzero elements of $\boldsymbol{u}$ and $\boldsymbol{v}$ signal variables, and the variables corresponding to the zero elements of $\boldsymbol{u}$ and $\boldsymbol{v}$ noise variables.

To estimate the canonical vectors under the single canonical pair model, we need to find the SVD of the matrix $\widetilde{K} = \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2} = \rho \Sigma_{XX}^{1/2} \boldsymbol{u} \boldsymbol{v}^T \Sigma_{YY}^{1/2}$. To see how variable selection in sparse, high dimensional CCA relates to a low rank matrix decomposition with row or column sparsity, we need to consider the structure of $\widetilde{K}$ implied by sparsity of the canonical vectors. We further assume that the signal variables are uncorrelated with the noise variables (i.e., that Σ_{XX} and Σ_{YY} are block diagonal). Let the superscripts S and N denote the signal and noise variables, respectively. Then, under suitable partitions of $\boldsymbol{u}$, $\boldsymbol{v}$, Σ_{XX}, and Σ_{YY}, we can write $\widetilde{K}$ as

$$
\begin{aligned}
\frac{1}{\rho}\widetilde{K} &= \begin{pmatrix} {\Sigma_{XX}^{SS}}^{1/2} & 0 \\ 0 & {\Sigma_{XX}^{NN}}^{1/2} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^S \\ \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{v}^{ST} & \mathbf{0}^T \end{pmatrix} \begin{pmatrix} {\Sigma_{YY}^{SS}}^{1/2} & 0 \\ 0 & {\Sigma_{YY}^{NN}}^{1/2} \end{pmatrix} \\
&= \begin{pmatrix} {\Sigma_{XX}^{SS}}^{1/2} \boldsymbol{u}^S \boldsymbol{v}^{ST} {\Sigma_{YY}^{SS}}^{1/2} & 0 \\ 0 & 0 \end{pmatrix}. \qquad (3.5)
\end{aligned}
$$

From equation (3.5), we can see that when the canonical vectors are sparse and the signal variables are not correlated with the noise variables, CCA involves finding the SVD of a rank-1 matrix with row and column sparsity. In addition, the within-set correlation structures of the noise variables play no role in the CCA optimization problem [i.e., Σ_{XX}^{NN} and Σ_{YY}^{NN} are absent from equation (3.5)]. Furthermore, the within-set correlation structures of the signal variables (i.e., Σ_{XX}^{SS} and Σ_{YY}^{SS}) affect neither the row and column sparsity patterns nor the rank of $\widetilde{K}$.

Although the structures of the within-set covariance matrices do not alter the overall objectives of the analysis, we must estimate them with their sample equivalents before performing CCA. We do not know which variables are the signal variables and which are the noise variables before the analysis, so we must estimate the entire $p \times p$ and $q \times q$ within-set covariance matrices. Because of the difficulty inherent in high-dimensional covariance estimation, some authors [e.g. Witten et al. (2009)] have suggested to standardize X and Y and avoid estimating the within-set covariance matrices by assuming they are identity; that is, set $\Sigma_{XX} = I_p$ and $\Sigma_{YY} = I_q$. Then the sample version of the problem reduces to the SVD of the $p \times q$ matrix containing the correlations between the variables in X and the variables in Y. We follow that suggestion throughout the rest of the manuscript and work with sample correlation matrix between X and Y. We denote the population correlation matrix as R_{XY} and the sample version as $\widehat{R}_{XY}$.

If we relax the objective of CCA from finding sparse $\boldsymbol{u}$ and $\boldsymbol{v}$ such that $Cor(X\boldsymbol{u},\ Y\boldsymbol{v})$ is maximized to finding a low rank approximation of $\widehat{R}_{XY}$ with the same row and column sparsity pattern as that implied by sparse $\boldsymbol{u}$ and $\boldsymbol{v}$, then we can consider the new objective as variable selection for CCA. That is, we shift the focus from accurate estimation of the canonical correlation to accurate variable selection. Under the assumptions stated in the previous paragraphs, the row sparsity pattern of R_{XY} has a one-to-one correspondence with the sparsity pattern of $\boldsymbol{u}$ and the column sparsity pattern of R_{XY} has a one-to-one correspondence with the sparsity pattern of $\boldsymbol{v}$. The variables corresponding to the nonzero elements of the canonical vectors, or equivalently, the nonzero rows/columns of R_{XY}, are the signal variables that we wish to select. Thus, rather than estimating $\boldsymbol{u}$ and $\boldsymbol{v}$ directly, we instead try to find a low rank approximation of $\widehat{R}_{XY}$ with sparse rows and columns.

We can find a low rank approximation of $\widehat{R}_{XY}$ with sparse rows or columns by setting $K = \widehat{R}_{XY}$ in problems (3.2) and (3.3) introduced in Section 3.2.1. Rows estimated as nonzero by solving problem (3.2) correspond to variables selected from the X dataset, and columns estimated as nonzero by solving problem (3.3) correspond to variables selected from the Y dataset. Moreover, as shown in the preceding paragraphs, the proposed methodology aims to select exactly the same (population-level) signal variables as would a sparse CCA method that standardizes the data and assumes the within-set covariance matrices are identity [such as Witten et al. (2009) and Safo et al. (2018)]. Thus, the proposed methodology shares a close connection with those sparse CCA methods in the sense that the goals regarding the variable selection aspect of the problem are the same.

A crucial difference between the proposed methodology and sparse CCA methods is how the two approaches find low rank structure in $\widehat{R}_{XY}$. To estimate the first canonical vector pair, sparse CCA methods find the leading left- and right-singular vectors of $\widehat{R}_{XY}$ under sparsity constraints. Because they are based on the SVD, such approaches can be interpreted as a sparse versions of finding the best rank-1 approximation of a matrix (in the sense of the Eckart and Young (1936) theorem). Then, sparse CCA methods find low rank structure in $\widehat{R}_{XY}$ by assuming a fixed rank. In contrast, the proposed methodology does not assume a fixed rank approximation. Rather, the penalty on the nuclear norm induces a low rank approximation by continuously shrinking the singular values to zero. Fixing the rank can be thought of as a hard-thresholding approach for finding a low rank approximation, while penalizing the nuclear norm can be thought of as a soft-thresholding approach.

Remark 4. We emphasize that problems (3.2) and (3.3) do not solve the same objective function as CCA. Although one could extract sparse vectors $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{v}}$ from the estimates $\widehat{W_r}$ and $\widehat{W_c}$, respectively, they would not be optimal in the sense that $Cor(X\hat{\boldsymbol{u}},\ Y\hat{\boldsymbol{v}})$ would be maximized. In fact, $Cor(X\hat{\boldsymbol{u}},\ Y\hat{\boldsymbol{v}})$ would not even be guaranteed to be positive. Consequently, the proposed method should not be considered as a sparse CCA method. Rather, we can say that the proposed approach is closely related to sparse CCA because they aim to select the same variables, albeit by solving different objective functions.

Remark 5. We note that, if desired, one could exploit the close connection of the proposed method with sparse CCA through a combined procedure. For example, one could use the proposed methodology to select variables from X and Y, then estimate the canonical vectors and canonical correlation by performing (non-sparse) CCA with the subsets of selected variables. We use that approach in the real data analysis in Section 3.4 for illustrative purposes, but focus most of our attention in other sections on the variable selection problem, leaving a thorough study of the various possibilities for a combined procedure as a subject of future study.

3.2.3 ADMM Algorithm

We propose an algorithm based on the alternating direction method of multipliers (ADMM) to solve problems (3.2) and (3.3). For ease of exposition, we discuss solving problem (3.2), but the similar results apply to problem (3.3). ADMM splits problem (3.2) as

$$\min_{W_1,W_2,W_3} \|K - W_1\|_F^2 + \lambda_1 \|W_2\|_* + \lambda_2 \sum_{r=1}^{p} \|\boldsymbol{w}_{3r}\|_2 \quad \text{s.t.} \quad W_j = \overline{W}, \quad j = 1, 2, 3$$

where the $\boldsymbol{w}_{3r}$'s are the rows of W_3. Splitting the objective function into three components by introducing additional parameters allows one to break the overall problem into three separate subproblems, each of which is easier to solve than the original problem. The constraint $W_j = \overline{W},\ j = 1, 2, 3$, enforces that the new parameters are all equal, which ensures that the new problem is the same as the original. The updates for ADMM consist of solving

$$\begin{aligned}
W_1^{k+1} &= \arg\min_{W_1} \ \|K - W_1\|_F^2 + Y_1^{kT}(W_1 - \overline{W}^k) + \frac{\eta}{2}\|W_1 - \overline{W}^k\|_F^2 \\
W_2^{k+1} &= \arg\min_{W_2} \ \lambda_1\|W_2\|_* + Y_2^{kT}(W_2 - \overline{W}^k) + \frac{\eta}{2}\|W_2 - \overline{W}^k\|_F^2 \\
W_3^{k+1} &= \arg\min_{W_3} \ \lambda_2 \sum_{r=1}^{p} \|\boldsymbol{w}_{3r}\|_2 + Y_3^{kT}(W_3 - \overline{W}^k) + \frac{\eta}{2}\|W_3 - \overline{W}^k\|_F^2 \\
\overline{W}^{k+1} &= (W_1^{k+1} + W_2^{k+1} + W_3^{k+1})/3 \\
Y_j^{k+1} &= Y_j^k + \eta(W_j^{k+1} - \overline{W}^{k+1}), \quad j = 1, 2, 3.
\end{aligned}$$

ADMM can be interpreted as a type of proximal algorithm (Parikh and Boyd, 2014), and the form of the updates described above is known as consensus ADMM (Boyd et al., 2011). The parameter η is an augmented Lagrangian parameter that controls the trade-off between minimizing one component of the objective function and satisfying the constraint. In all of our experiments, we set $\eta = 1$.

The update for W_1 is straightforward to solve using matrix calculus. The updates for W_2 and W_3 are solved by the proximal operators of the functions $f := \|\cdot\|_*$ and $g := \sum_{r=1}^{p} \|\cdot\|_2$. The proximal operators of the nuclear norm f and group LASSO g are given by

$$\begin{aligned} \mathbf{prox}_{\lambda_1 f}(W) &= \sum_{j=1}^{\min(p,q)} (\sigma_j - \lambda_1)_+ \boldsymbol{u}_j \boldsymbol{v}_j^T && \text{(nuclear norm)} \\ \mathbf{prox}_{\lambda_2 g}(W) &= \left(1 - \frac{\lambda_2}{\|\boldsymbol{w}_r\|_2}\right)_+ \boldsymbol{w}_r,\ r = 1, \ldots, p && \text{(group LASSO)} \end{aligned}$$

where $(\cdot)_+ = \max(0, \cdot)$ denotes the soft thresholding operator and $W = \sum_{j=1}^{\min(p,q)} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^T$ denotes the SVD of W. Thus, the proximal operator of the nuclear norm of W involves shrinking its singular values to zero, and the proximal operator of the group LASSO applied to the rows of W involves shrinking entire rows to zero. Note that the only difference in the ADMM algorithm for solving (3.3) as opposed to (3.2) is that the proximal operator of the group LASSO is applied to the columns rather than the rows. For more information about the proximal operators of the nuclear norm and group LASSO, see Parikh and Boyd (2014) and references therein.

The full details of the proposed ADMM algorithm are described in Algorithm 5. In addition to the updates described above, Algorithm 5 describes some other details relevant to ADMM, such as the stopping conditions.

In Algorithm 5, each of the parameter updates satisfies one component of the objective function exactly and the others approximately. That is, $\widehat{W}_1$ is close to K with respect to the Frobenius norm, $\widehat{W}_2$ is low rank, and $\widehat{W}_3$ is row-/column-sparse, but none of the three estimates are simultaneously close to K, low rank, and row-/column-sparse. Similarly, the consensus estimate $\widehat{\widehat{W}}$ is only approximately low rank and row-/column-sparse.

Because the nature of ADMM precludes any single estimate from being simultaneously low rank and sparse, Algorithm 5 returns both $\widehat{W}_2$ and $\widehat{W}_3$ as the estimate of $\widehat{W}_r$. $\widehat{W}_2$ contains the information about the dominant modes of co-variation between X and Y, and $\widehat{W_3}$ contains the information about the subset of variables that contribute most to the dominant modes of co-variation. If one was concerned only with which variables were selected by solving problem (3.2), one would only need the estimate of W_3.

Algorithm 5: ADMM algorithm for Sparse, Low Rank Matrix Approximation

```
Input: Matrix $K$ (such as the sample correlation matrix $\widehat{R}_{XY}$ or covariance matrix $\widehat{\Sigma}_{XY}$), tuning parameters
        $\lambda_1,\ \lambda_2$ and $\lambda_3$, absolute tolerance $\epsilon^{abs}$, relative tolerance $\epsilon^{rel}$, augmented Lagrangian parameter $\eta$.
Output: Final estimate $\widehat{W}_r = \{\widehat{W}_2,\ \widehat{W}_3\}$
.
// intialize parameters
$W_1^0 \leftarrow K;\ \ W_2^0 \leftarrow 0_{p\times q};\ \ W_3^0 \leftarrow 0_{p\times q}$
$\overline{W}^0 = (W_1^0 + W_2^0 + W_3^0)/3$
$Y_j^0 \leftarrow 0_{p\times q},\ j = 1,2,3$
$\epsilon^{primal} \leftarrow \sqrt{3pq} * \epsilon^{abs}$ // primal tolerance
$\epsilon^{dual} \leftarrow \sqrt{3pq} * \epsilon^{abs}$ // dual tolerance
$res^{primal} \leftarrow \epsilon^{primal} * 1.1$ // primal error
$res^{dual} \leftarrow \epsilon^{dual} * 1.1$ // dual error
$k \leftarrow 0$ // iteration counter

while $res^{primal} > \epsilon^{primal}$ or $res^{dual} > \epsilon^{dual}$ do

    // solves $\arg\min_{W_1}\ \ \|K - W_1\|_F^2 + Y_1^{kT}(W_1 - \overline{W}^k) + \frac{\eta}{2}\|W_1 - \overline{W}^k\|_F^2$
    $W_1^{k+1} \leftarrow \frac{1}{1+\eta}(K + \eta\overline{W}^k - Y_1^k)$

    // solves $\arg\min_{W_2}\ \ \lambda_1\|W_2\|_* + Y_2^{kT}(W_2 - \overline{W}^k) + \frac{\eta}{2}\|W_2 - \overline{W}^k\|_F^2$
    $W_2^{k+1} \leftarrow \sum_{j=1}^{\min(p,q)}(\sigma_{2j} - \lambda_1)_+ \boldsymbol{u}_{2j}\boldsymbol{v}_{2j}^T$ // soft threshold singular values from SVD($W_2^k$)

    // solves $\arg\min_{W_3}\ \ \lambda_2\sum_{r=1}^{p}\|\boldsymbol{w}_{3r}\|_2 + Y_3^{kT}(W_3 - \overline{W}^k) + \frac{\eta}{2}\|W_3 - \overline{W}^k\|_F^2$
    $\boldsymbol{w}_{3r}^{k+1} \leftarrow \left(1 - \frac{\lambda_2}{\|\boldsymbol{w}_{3r}^k\|_2}\right)_+ \boldsymbol{w}_{3r}^k,\ r = 1,\dots,p$ // soft threshold rows of $W_3^k$

    // consensus
    $\overline{W}^{k+1} \leftarrow (W_1^{k+1} + W_2^{k+1} + W_3^{k+1})/3$

    // residuals
    $Y_j^{k+1} \leftarrow Y_j^k + \eta(W_j^{k+1} - \overline{W}^{k+1}),\ j = 1,2,3$

    $res^{primal} \leftarrow \|(W_1^{k+1}, W_2^{k+1}, W_3^{k+1}) - (\overline{W}^{k+1}, \overline{W}^{k+1}, \overline{W}^{k+1})\|_F$
    $res^{dual} \leftarrow \| - \eta((\overline{W}^{k+1}, \overline{W}^{k+1}, \overline{W}^{k+1}) - (\overline{W}^k, \overline{W}^k, \overline{W}^k))\|_F$
    $\epsilon^{primal} \leftarrow \sqrt{3pq} * \epsilon^{abs} + \epsilon^{rel} * \max\left(\|(W_1^{k+1}, W_2^{k+1}, W_3^{k+1})\|_F,\ \| - (\overline{W}^{k+1}, \overline{W}^{k+1}, \overline{W}^{k+1})\|_F\right)$
    $\epsilon^{dual} \leftarrow \sqrt{3pq} * \epsilon^{abs} + \epsilon^{rel} * \|(Y_1^{k+1}, Y_2^{k+1}, Y_3^{k+1})\|_F$

    $k \leftarrow k + 1$ // update iteration counter

end

// Repeat procedure to find $\widehat{W}_c$, replacing the update of $W_3$ with soft thesholding of the columns.
```

3.2.4 Variable Importance Metrics

In sparse CCA, there are tuning parameters associated with each canonical vector that affect their sparsity levels. A variety of approaches have been proposed to select the tuning parameters for sparse CCA. Witten and Tibshirani (2009) suggested a permutation approach for both selecting the tuning parameters and testing for significance of the estimated canonical correlation. Wilms and Croux (2015) proposed to select the tuning parameters at each iteration based on BIC under a Gaussian likelihood assumption. Others have proposed to select the tuning parameters by M-fold cross-validation. Waaijenborg et al. (2008) suggested to choose the tuning parameters that minimize the average difference between the training correlation $\hat{\rho}_{train}^{-m} = Cor(X^{-m}\hat{\boldsymbol{u}}^{-m}, Y^{-m}\hat{\boldsymbol{v}}^{-m})$ and the test correlation $\hat{\rho}_{test}^{m} = Cor(X^{m}\hat{\boldsymbol{u}}^{-m}, Y^{m}\hat{\boldsymbol{v}}^{-m})$. Safo et al. (2018) used a similar criterion. Parkhomenko et al. (2009) and Chalise and Fridley (2012) suggested to maximize the average test correlation.

For the proposed methodology, there are tuning parameters that affect the rank and row/column sparsity of the low rank approximation of K. Unlike the sparse CCA methods, the proposed methodology does not solve the CCA objective function, nor does it produce an estimate of the canonical correlation. Thus, we cannot adapt the methods proposed by others in the context of sparse CCA to our problem. One alternative is to use a measure of variable selection stability, such as Cohen's κ (Cohen, 1960), as the criterion in cross-validation, bootstrapping, or related techniques. This approach has been applied in the context of penalized regression (Sun et al., 2013), but in principle could be applied to any objective function as long as variable selection was a key goal of the analysis. However, any approach based on cross-validation or bootstrapping entails a large computational burden.

We take a different approach that avoids selection of the tuning parameters altogether and instead produces a ranking of the variables according to each variable's "importance," which we define formally shortly. There are several reasons that a ranking of the variables might be considered more useful than selecting fixed values of the tuning parameters. Whereas selecting fixed values of the tuning parameters produces a single subset of selected variables, a ranking of the variables can be considered as a kind of continuous variable selection, in the sense that one can set one or more thresholds for the rank that correspond to subsets of selected variables, and within each subset, the variables are ordered according to their importance. One can also think of a ranking of the variables as a sequence of nested subsets of selected variables, where subsets of smaller size include the more important variables. In practice, it is often of interest to explore multiple scales of sparsity, and a ranking of the variables provides a natural way to do so. It also gives researchers the flexibility to include the context of the problem in the variable selection process, rather than relying solely on a data-driven procedure. For example, if a group of highly ranked variables are all part of the same biological pathway, it makes sense to set a threshold that includes all of those variables.

For both the proposed methodology and sparse CCA methods, one solves the optimization problem for a sequence of values of the tuning parameters over a suitable range. Because all of the problems involve two tuning parameters, the pairs of values are chosen from a two-dimensional grid. The endpoints of the grid are chosen to yield a suitable range of sparsity in the solutions. The values of the upper endpoints are typically the smallest values that yield a completely sparse solution (because larger values $\implies$ more penalization $\implies$ higher sparsity). The values of the lower endpoints are typically the largest values that select the maximum number of variables desired. Two common choices for the maximum number of selected variables are all of the variables and, for high dimensional problems, a number of variables equal to or slightly less than the sample size.

We define two simple, intuitive metrics to measure a variable's importance that combine information across the entire sequence of optimization problems. For one importance measure, we simply count the number of times a variable was selected over the entire sequence of values of the tuning parameters tried. For example, for a 30×30 grid, the metric counts the number of times a variable was selected out of 900. The intuition is that the longer a variable persists along the solution path (i.e., the more times it is selected), the more important it is. For a given pair of values of the tuning parameters, the proposed metric weights all of the selected variables as equally important. Traditionally in CCA, a variable's relative importance is interpreted with respect to the magnitude of its canonical vector loading, with larger magnitudes indicating greater importance. Thus, we also consider another importance metric that sums up the magnitudes of the canonical vector loadings over the entire sequence of values of the tuning parameters tried. A metric that accumulates the magnitudes of the canonical vector loadings not only accounts for the the number of times a variable was selected, but also allows for unequal weighting among the selected variables for each pair of values of the tuning parameters.

The second importance metric requires the magnitudes of the loadings of the canonical vectors. Because the proposed methodology does not actually solve the CCA objective function, it does not produce estimates of the canonical vectors. Thus, to calculate the second importance metric, we must first extract two vectors from the estimates $\widehat{W_r}$ and $\widehat{W_c}$ produced by Algorithm 5 that are in some sense like estimated canonical vectors. For $\widehat{W_r}$, we first extract the leading left-singular vector from $\widehat{W_2}$. We then set elements of the singular vector to zero so that it has the same row-sparsity pattern as $\widehat{W_3}$. Thus, we produce a vector that takes advantage of the fact that $\widehat{W_2}$ is exactly low rank and $\widehat{W_3}$ is exactly sparse. We repeat the procedure for $\widehat{W_c}$, extracting the leading right-singular vector from $\widehat{W_2}$ and setting elements to zero to correspond to the column-sparsity pattern of $\widehat{W_3}$. The magnitudes of the elements of the sparse singular vectors are treated like the loadings of the canonical vectors in CCA and are used the calculate the second variable importance metric.

The proposed metrics are less computationally demanding than cross-validation because they only require the optimization problem to be solved for each pair of values of the tuning parameters. In contrast, cross-validation not only requires the optimization problem to be solved for each pair of values of the tuning parameters, but also to repeat the procedure M

times – once for each fold. In addition, for the proposed metrics, one need only choose a sequence of the tuning parameters that explores a reasonable range of sparsity levels. For cross-validation, one needs to first choose an appropriate sequence of values, then select a single pair out of the many pairs tried. The latter task poses a substantially more challenging problem than the former.

The variable importance metrics can be applied to both the proposed methodology and sparse CCA methods, enabling comparisons among the different approaches. In particular, we use the two importance metrics to compare the proposed methodology to two sparse CCA methods with respect to the estimated average rank of the signal variables. The average rank of the signal variables is a proxy for the variable selection accuracy. For example, for a method that achieves perfect variable selection (i.e, selects all p^{sig} signal variables and none of the $p - p^{\text{sig}}$ noise variables), the best possible average rank of the signal variables (assuming no ties) is

$$\frac{1}{p^{\text{sig}}} \sum_{j=1}^{p^{\text{sig}}} j = \frac{(p^{\text{sig}} + 1)}{2}.$$

Note that because we are not defining a true ranking at the population level – we are only defining which variables are signal and which are noise – the ordering of the estimated ranks of the signal variables does not matter (which is why we take the average). Also note that because of the way we have defined the ranks, rank 1 corresponds to the most important variable. In our simulation experiments in Section 3.3, lower values of the average rank indicate better performance.

3.3 Simulation Experiment

We evaluate the performance of the proposed methodology, which we call spLRMA (for *sp*arse *L*ow *R*ank *M*atrix *A*pproximation), through a simulation experiment. We compare to the penalized matrix decomposition (PMD) method of Witten et al. (2009) and SELP-I method of Safo et al. (2018). Both PMD and SELP-I are sparse CCA methods that standardize the data and assume the within-set covariance matrices are identity. Thus, the proposed methodology, PMD, and SELP-I all attempt to find a low rank approximation of the sample correlation matrix $\widehat{R}_{XY}$ while simultaneously performing variable selection. We simulate data under the assumptions stated in Section 3.2.2 so that all three methods aim to select the same sets of signal variables. We measure the performance of each method with respect to its variable selection accuracy. We use the variable importance metrics described in Section 3.2.4 to rank the variables, then use the average rank of the signal variables as a proxy for the variable selection accuracy.

We focus on the high dimensional setting in our experiments. We simulate $p = 200$ variables in the X dataset and $q = 160$ variables in the Y dataset with a sample size of $N = 150$. For one group of settings, we use the single canonical pair model introduced by Chen et al. (2013), which is based on the multivariate normal (MVN) distribution. We fix the within-set covariance matrices Σ_{XX} and Σ_{YY} and generate a set of canonical vectors $(\boldsymbol{u}, \boldsymbol{v})$. Then the single canonical pair model defines the between-covariance matrix as $\Sigma_{XY} = \rho\Sigma_{XX}\boldsymbol{u}\boldsymbol{v}^T\Sigma_{YY}$. We use several different block-diagonal structures for Σ_{XX} and Σ_{YY} and vary the strength of the within-set correlation parameter as well as the canonical correlation. For all settings but one, we use 90% sparsity for the canonical vectors (i.e., 20 signal variables in X, 16 signal variables in Y); for the other setting, we use 95% sparsity (i.e., 10 signal variables in X, 8 signal variables in Y). We draw the values of the signal portion of the canonical vectors independently from $Unif(1, 2)$ and the sign from Bernoulli with $P(+) = P(-) = 0.5$. We set the noise portion equal to zero. We normalize the canonical vectors with respect to their corresponding covariance matrices prior to generating (X, Y) (i.e., we scale $\boldsymbol{u}$ and $\boldsymbol{v}$ by a factor of $1/\sqrt{\boldsymbol{u}^T\Sigma_{XX}\boldsymbol{u}}$ and $1/\sqrt{\boldsymbol{v}^T\Sigma_{YY}\boldsymbol{v}}$, respectively). We give the full details of the simulation settings based on the MVN distribution in Table 3.1.

For another group of settings, we adapt the single canonical pair model to the multivariate t distribution. Our goal is to determine how the methods perform when the data may contain outliers. The covariance matrix of the multivariate t distribution is defined through the degrees of freedom and scale matrix. We fix the degrees of freedom then set the scale matrices as we did the covariance matrices of the MVN distribution. We use $df = 15$ in all of our settings. The rest of the details for the settings based on the multivariate t distribution are included in Table 3.1.

Table 3.1: Within-set covariance/scale matrices and canonical correlations for simulation settings based on the multivariate normal (MVN) or multivariate $t(df = 15)$ distribution. BD: block diagonal, CS: compound symmetry, AR: first-order autoregressive.

Distribution	Σ_{XX}	Σ_{YY}	ρ
MVN	$BD\,[AR(0.9)_{20},\ I_{180}]$	$BD\,[AR(0.9)_{16},\ I_{144}]$	0.9, 0.5
	$BD\,[AR(0.7)_{20},\ I_{180}]$	$BD\,[AR(0.7)_{16},\ I_{144}]$	0.9, 0.7, 0.5, 0.3
	$BD\,[AR(0.2)_{20},\ I_{180}]$	$BD\,[AR(0.2)_{16},\ I_{144}]$	0.9
	$BD\,[CS(0.2)_{20},\ I_{180}]$	$BD\,[CS(0.2)_{16},\ I_{144}]$	0.9, 0.5
	$BD\,[I_{20},\ I_{180}]$	$BD\,[I_{16},\ I_{144}]$	0.9
	$BD\,[I_{10},\ I_{190}]$	$BD\,[I_{8},\ I_{152}]$	0.9
	$BD\,[AR(0.7)_{20},\ AR(0.6)_{180}]$	$BD\,[AR(0.7)_{16},\ AR(0.6)_{144}]$	0.9
	$BD\,[CS(0.7)_{20},\ I_{180}]$	$BD\,[CS(0.7)_{16},\ I_{144}]$	0 (null setting)
MV $t(df = 15)$	$BD\,[AR(0.7)_{20},\ I_{180}]$	$BD\,[AR(0.7)_{16},\ I_{144}]$	0.9, 0.5
	$BD\,[CS(0.2)_{20},\ I_{180}]$	$BD\,[CS(0.2)_{16},\ I_{144}]$	0.9, 0.5

Two settings in Table 3.1 are of particular interest. One setting is when the signal variables are uncorrelated, so that $\Sigma_{XX} = I_p$ and $\Sigma_{YY} = I_q$. For that setting, the methods of Witten et al. (2009) and Safo et al. (2018) assume the correct population-level covariance structure, so we might expect the methods to perform better in that setting relative to the other settings. The other setting of special interest is the null setting. When $\rho = 0$, the true population correlation matrix $R_{XY} = 0$. Although the between-set correlation is zero, we set some variables to have high within-set correlation. We still refer to the variables with high within-set correlation as "signal" variables, even though the corresponding elements of the canonical vectors are zero. Under the null setting, we might expect all of the methods to rank the signal variables about the same as the noise variables. Our purpose in considering such a setting is to determine to what extent the methods are responding to the structure of Σ_{XY} as dictated by ρ, $\boldsymbol{u}$, and $\boldsymbol{v}$ vs. Σ_{XX} and Σ_{YY}.

3.3.1 Results: Multivariate Normal

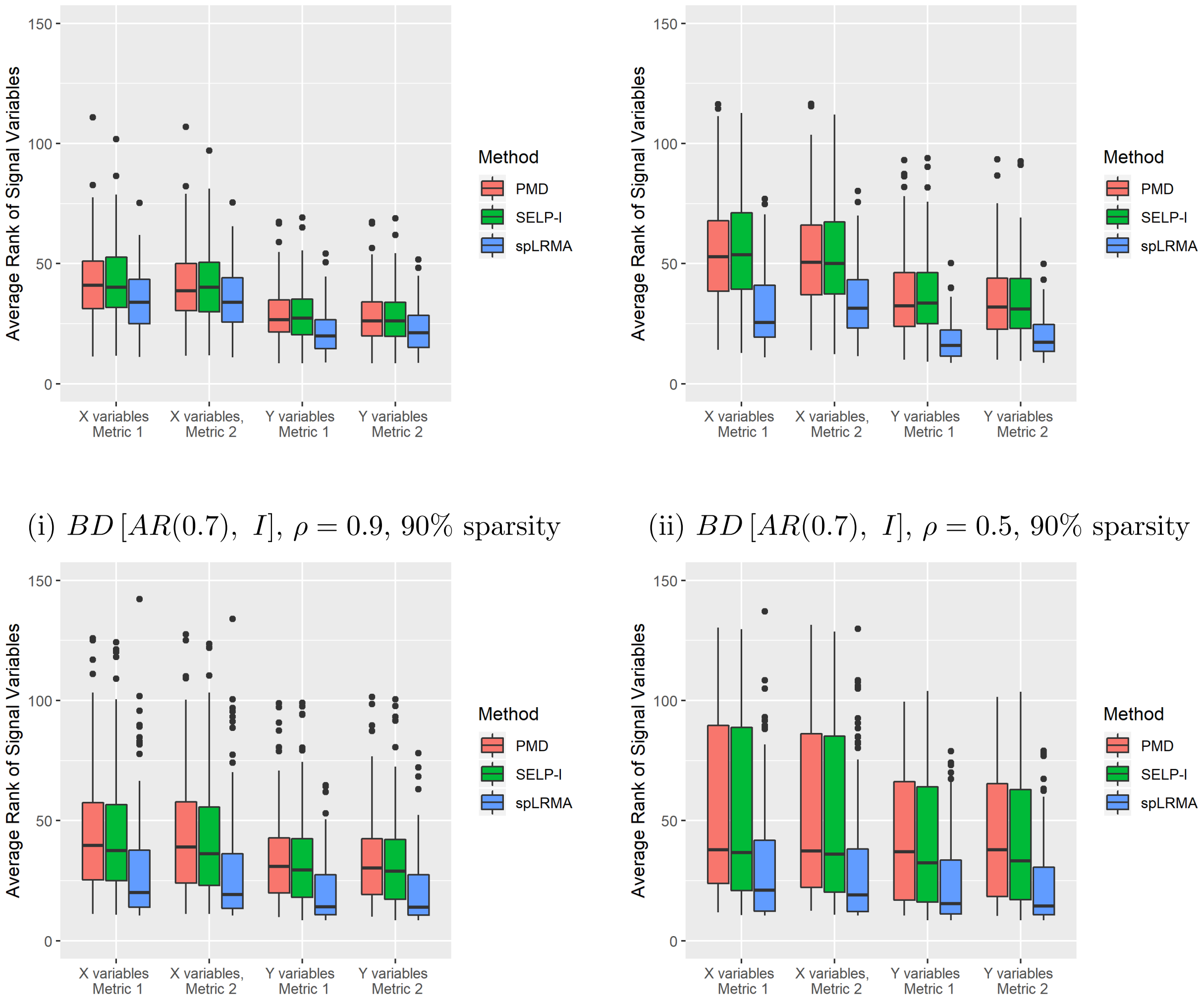
In Figures 3.1–3.4, we show the results for some selected settings in Table 3.1 corresponding to the MVN distribution. The results for all of the settings (reported as mean and SD) can be found in Table 5.1 in the Appendix. We focus on four groups of settings: (1) correlated signal variables, uncorrelated noise variables, (2) correlated signal variables, correlated noise variables, (3) uncorrelated signal variables, uncorrelated noise variables, and (4) the null setting. Note that uncorrelated vs. correlated (i.e., identity vs. non-identity) refers to the block-diagonal structure of the within-set covariance matrices; in all of the settings, the signal variables are uncorrelated with the noise variables.

Figures 3.1–3.4 show boxplots of the average rank of the signal variables for each method. Each figure contains the results for both sets of variables X and Y as well as both variable importance metrics. Metric 1 corresponds to the metric based on the number of times a variable was selected and Metric 2 corresponds to the metric based on the magnitudes of the loadings.

For the X set, the average rank of the signal variables is the average ranks of the first 20 variables for settings with 90% sparsity or the first 10 variables for the setting with 95% sparsity. The best possible average rank (assuming no ties) of the X signal variables is 10.5 for 90% sparsity and 5.5 for 95% sparsity. For the Y set, the average rank of the signal variables is the average ranks of the first 16 variables for settings with 90% sparsity or the first 8 variables for the setting with 95% sparsity. The best possible average rank of the Y signal variables is 8.5 for 90% sparsity and 4.5 for 95% sparsity.

Correlated signal variables, uncorrelated noise variables

In Figure 3.1, we show results for settings using first-order autoregressive (AR) and compound symmetric (CS) covariance structures with different values of the canonical correlation. PMD and SELP-I perform about the same as each other in all four settings. In addition, both methods perform worse when the signal variables are less correlated among themselves [$AR(0.7)$ vs. $CS(0.2)$] or when the value of the canonical correlation becomes smaller ($\rho = 0.9$ vs. 0.5). The performances decline both with respect to the value of the average rank (the values tend to get further from the best possible rank) and the variability across repeated simulation. In contrast, spLRMA outperforms PMD and SELP-I in all of those aspects. For a given setting, the average rank tends to be closer to the best possible rank and is less variable. It is also more robust to changes in the degree of correlation of the signal variables and the value of the canonical correlation, achieving about the same value of the average rank in all four settings [albeit with slightly increased variability for the $CS(0.2)$ setting]. For all three methods, the average rank does not appear to depend on the variable importance metric.



(i) $BD\,[AR(0.7),\ I]$, $\rho = 0.9$, 90% sparsity

(ii) $BD\,[AR(0.7),\ I]$, $\rho = 0.5$, 90% sparsity

(iii) $BD\,[CS(0.2),\ I]$, $\rho = 0.9$, 90% sparsity

(iv) $BD\,[CS(0.2),\ I]$, $\rho = 0.5$, 90% sparsity

Figure 3.1: Boxplots of average rank of signal variables. Selected settings with correlated signal variables, uncorrelated noise variables. 100 simulations.

Correlated signal variables, correlated noise variables

In Figure 3.2, we show results for the setting in which the noise variables are correlated among themselves. Because the data were generated under the assumptions stated in Section 3.2.2, the population correlation matrix has the same block structure as the matrix in equation (3.5). Namely, the population-level matrix R_{XY} does not depend on the within-set correlation structure of the noise variables. Then, in theory, the results should not depend on whether the noise variable are correlated among themselves or not.

Comparing the results in Figure 3.2 to the results in Figure 3.1(i), we can see that the correlation structure of the noise variables *does* affect the analysis, at least empirically. The simulation settings corresponding to those figures are comparable because the only difference is the correlation structure of the noise variables. When the noise variables are correlated among themselves, all of the methods perform worse. The average rank tends to be further from the best possible rank and it is more variable across multiple simulation. In addition, the relative performances of the three methods differ between Figure 3.2 and Figure 3.1(i). In Figure 3.1(i), spLRMA performed better than PMD and SELP-I, but in Figure 3.2, all three methods perform about the same.

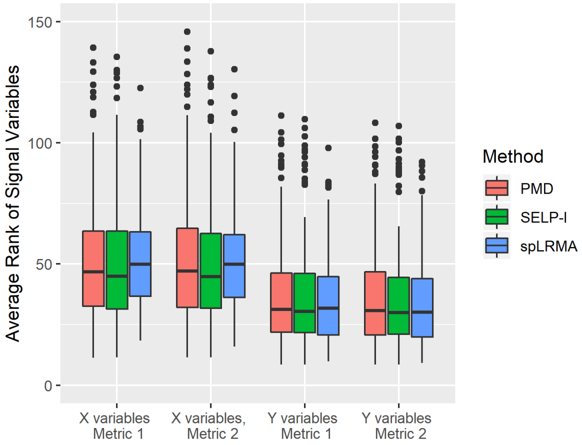


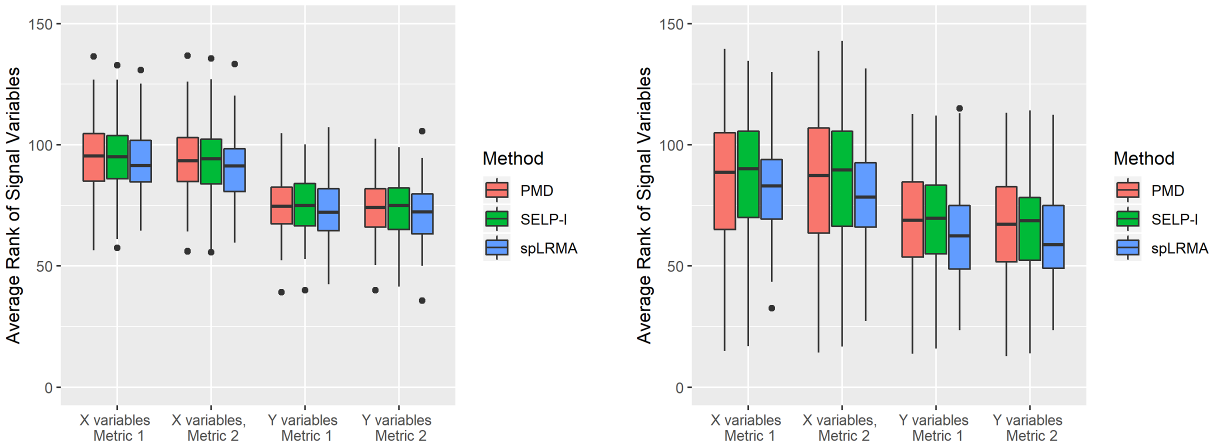
Figure 3.2: Boxplots of average rank of signal variables. Setting with correlated signal variables, correlated noise variables: $BD\left[AR(0.7),\ AR(0.6)\right]$, $\rho = 0.9$, 90% sparsity. 100 simulations.

Uncorrelated signal variables, uncorrelated noise variables

Figure 3.3 shows the results when the signal variables are uncorrelated for two different levels of sparsity. For the settings corresponding to Figure 3.3, $\Sigma_{XX} = I_p$ and $\Sigma_{YY} = I_q$. Because PMD and SELP-I both assume the correct covariance structure, we might expect them to perform better than in settings in which the signal variables are correlated. However, the results in Figure 3.3 indicate that they perform much worse. In fact, all three methods perform worse than in any other simulation setting, barely discriminating between the signal and noise variables. To illustrate, consider the X set of variables under the 90% sparsity setting. In the best case scenario – when the signal and noise variables are ranked perfectly – the average rank of the signal variables is 10.5 while the average rank of the noise variables is 110.5. When the ranks are chosen at random, the average rank of both sets is 100.5. In Figure 3.3(1), the bulks of the boxplots fall just below 100, meaning that the average rank of the signal variables is hardly different than the average rank of the noise variables.

For both 90% sparsity and 95% sparsity, spLRMA seems to perform slightly better than PMD and SELP-I, with the advantage becoming more noticeable with higher sparsity. With higher sparsity, all three methods perform better for single instances of simulated data [the whiskers of the boxplots in Figure 3.3(ii) extend closer toward the best possible rank], but at the expense of greater overall variability.

We provide additional discussion of the results presented in Figure 3.3 in Section 3.5.



(i) $BD\,[I,\ I]$, $\rho = 0.9$, 90% sparsity

(ii) $BD\,[I,\ I]$, $\rho = 0.9$, 95% sparsity

Figure 3.3: Boxplots of average rank of signal variables. Settings with uncorrelated signal variables, uncorrelated noise variables. 100 simulations.

Null setting

In the null setting, there are no true signal variables because the canonical vectors and correlation are zero. However, the first 20 variables in the X set are correlated among themselves and first 16 variables in the Y are correlated among themselves. Because the corresponding 20×16 matrix in the upper left corner of R_{XY} is zero, we should not expect any of the methods to rank those “signal” variables differently from the rest. That is, average rank should be about the same as if the rank were chosen at random: 100.5 in the X set and 80.5 in the Y set.

Although there is high variability, the results in Figure 3.4 indicate that all three methods are able to detect the variables with high within-set correlation using only the information in the sample between-set correlation matrix $\widehat{R}_{XY}$. The average rank of the those variables is frequently much smaller than 100.5 and 80.5. Among the three methods compared, spLRMA does the worst in the sense that it most consistently finds the variables with high within-set correlation using the information from $\widehat{R}_{XY}$, even though the true between-set correlation is zero.

We provide additional discussion of the results for the null setting in Section 3.5.

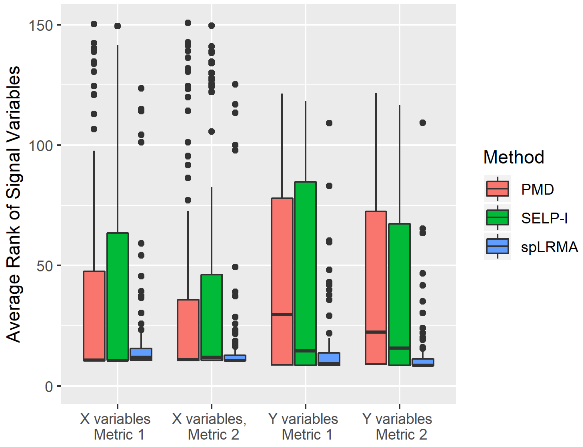
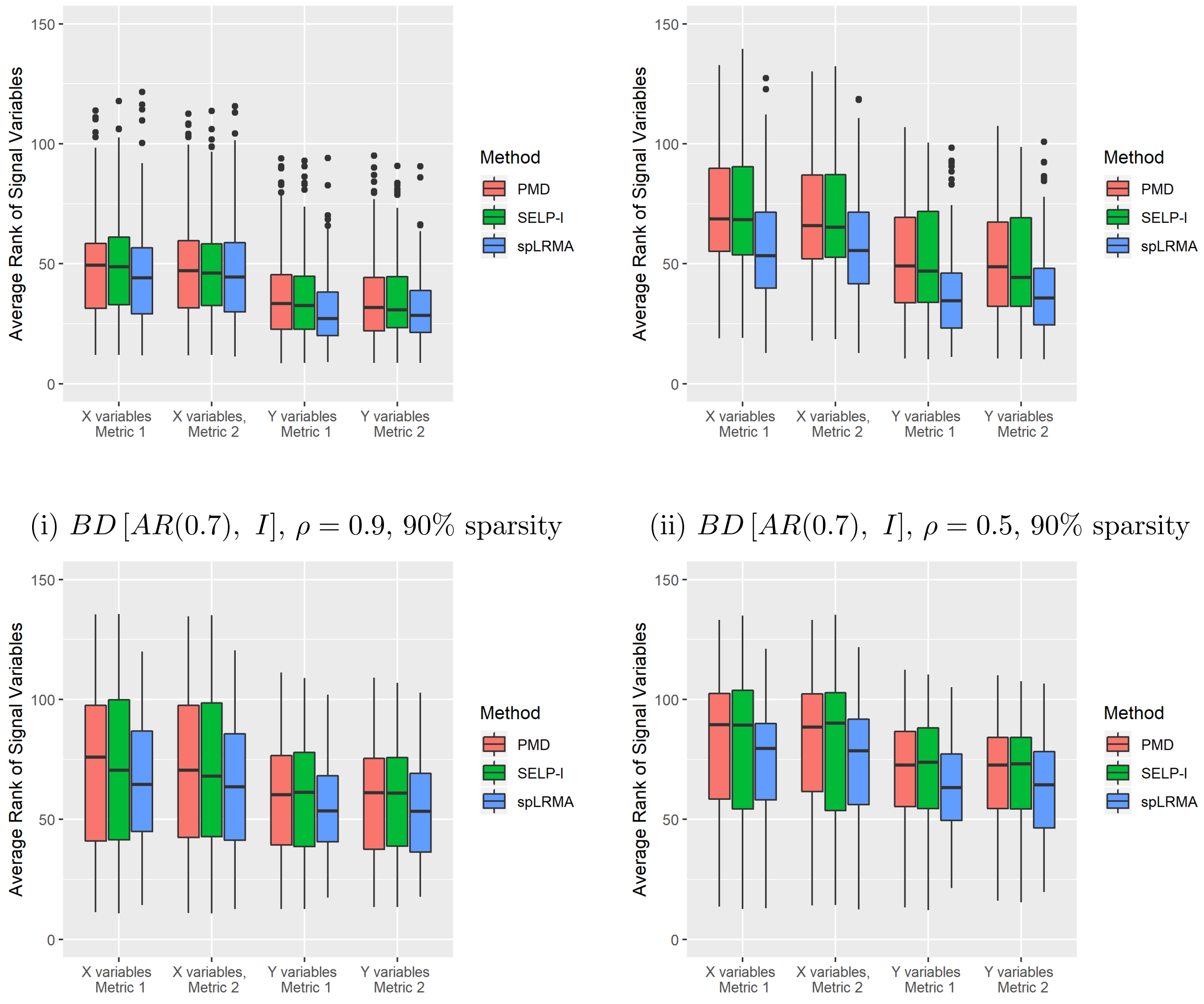


Figure 3.4: Boxplots of average rank of "signal" variables. Null setting ($R_{XY} = 0$). 100 simulations.

3.3.2 Results: Multivariate $t(df = 15)$

Figure 3.5 shows the results for the simulation settings based on the multivariate $t(df = 15)$ distribution. The results in Figure 3.5 correspond to the results in Figure 3.1 in the sense that the scale matrices for the multivariate t settings are the same as the covariance matrices for the MVN settings. Our goal was to determine how the methods fared when the data were drawn from a distribution with heavier tails than the MVN distribution.

Comparing the results in Figure 3.5 with those in Figure 3.1, we can see that the qualitative patterns are similar. PMD and SELP-I perform about the same, and spLRMA performs better than either. Both PMD and SELP-I perform worse when the value of the canonical correlation becomes smaller ($\rho = 0.9$ vs. 0.5), but the performance of spLRMA is more robust. Unlike the MVN settings, all three methods perform noticeably worse when the signal variables are less correlated among themselves [$AR(0.7)$ vs. $CS(0.2)$]. The variability of the methods' performances also appears to increase as the signal variables become less correlated.



(i) $BD\,[AR(0.7),\ I]$, $\rho = 0.9$, 90% sparsity

(ii) $BD\,[AR(0.7),\ I]$, $\rho = 0.5$, 90% sparsity

(iii) $BD\,[CS(0.2),\ I]$, $\rho = 0.9$, 90% sparsity

(iv) $BD\,[CS(0.2),\ I]$, $\rho = 0.5$, 90% sparsity

Figure 3.5: Boxplots of average rank of signal variables. Settings with correlated signal variables, uncorrelated noise variables. 100 simulations.

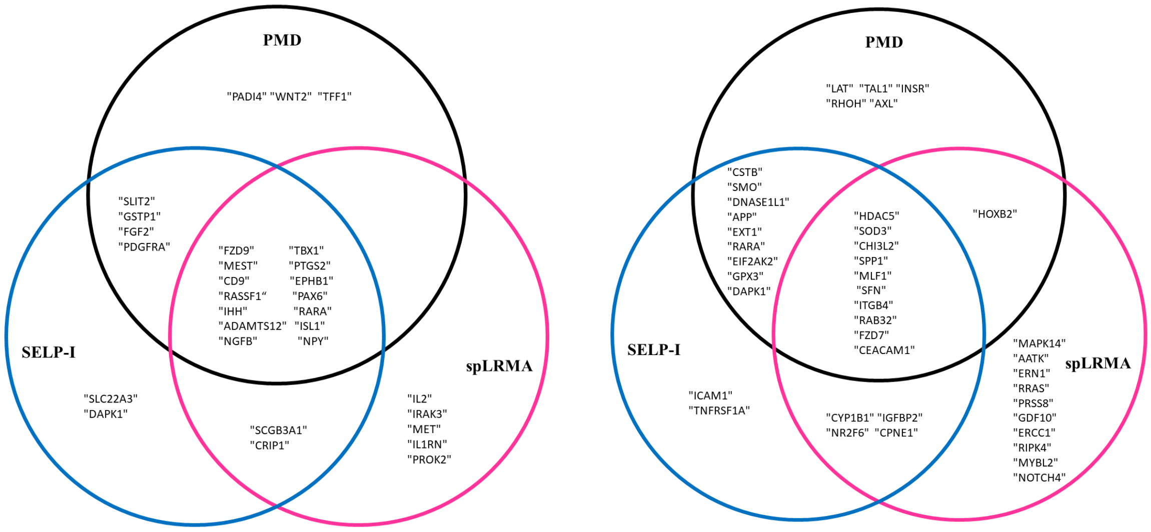
3.4 Real Data Analysis

We apply the proposed method (spLRMA), PMD (Witten et al., 2009), and SELP-I (Safo et al., 2018) to the breast cancer data from Holm et al. (2010). The breast cancer data consist of two sets of variables measured from a common set of $N = 179$ samples. One set contains methylation levels at 1452 CpG sites. The other set contains gene expressions as measured by 511 gene probes. CpG sites are often found in high frequency in the promoter regions of genes. Epigenetic changes such as methylation can affect the expression of the downstream gene. Hypermethylation is associated with gene silencing, while hypomethylation is associated with overexpression. Because the expression of one gene can affect the expression of others, changes in the methylation level at one CpG site can affect the expression levels of an entire group of genes. Moreover, changes at multiple CpG sites may affect the expression of overlapping groups of genes. Thus, the ultimate goal of the analysis is to select a group of CpG sites whose methylation levels are associated with the expression of a group of genes.

In both the original analysis by Holm et al. and a previous analysis by Safo et al., the dimension of the methylation set was reduced in a preprocessing step by removing CpG sites with standard deviation less than 0.3. We follow that procedure here, resulting in $p = 334$ CpG sites included in our analysis. We retain all of the genes from the expression set, resulting in expressions measured by $q = 511$ probes. Note that each CpG site is located at a gene, and multiple CpG sites can be located at the same gene. As a consequence, the 334 CpG sites correspond to only 249 unique genes. Similarly, multiple probes were sometimes used to measure the expression of one gene, so the 511 probes measure the expression of only 470 unique genes. Among the unique genes in the methylation and expression sets, 139 genes are common to both sets.

In our analysis, we aim to demonstrate empirically the close connection between spLRMA and sparse CCA that was explored theoretically in Section 3.2.2 and demonstrated through simulation in Section 3.3. Namely, we aim to demonstrate that: (1) spLRMA selects similar sets of variables as does PMD and SELP-I and (2) the overall association between the two sets of selected variables can be described by a high canonical correlation. In Section 3.2.2, the fact that spLRMA should select the same subset of variables as sparse CCA methods relied on two key assumptions: sparsity of the canonical vectors and uncorrelatedness of the signal and noise variables. Neither of those assumptions is likely to hold for real data because real data are not generated from a hypothetical CCA model. Nevertheless, we hope that spLRMA selects similar variables as sparse CCA in practice as well as in theory. In addition, we hope that the overall association between the two sets of variables selected by spLRMA can be described by a high canonical correlation. spLRMA captures the dominant modes of co-variation between two sets of variables, but it *does not* maximize the same objective function as CCA. Then if we use the estimate of the dominant modes of co-variation produced by Algorithm 5 to construct two vectors $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{v}}$, we cannot expect the linear combinations $X\hat{\boldsymbol{u}}$ and $Y\hat{\boldsymbol{v}}$ to have high correlation. However, the canonical correlation *does* capture *one aspect* of the dominant modes of co-variation between two sets of variables. Thus, if we perform (non-sparse) CCA with the sets of variables selected by spLRMA, we might expect that canonical correlation to be high.

We apply spLRMA, PMD, and SELP-I to the breast cancer data, using the metric based on the magnitudes of the canonical vector loadings to rank the variables in the methylation and expression sets according to their relative importances. Figure 3.6 shows the overlap in the top 25 genes selected by spLRMA, PMD, and SELP-I for each set of variables. Note that the choice of top 25 is purely for illustrative purposes. For the methylation set, CpG sites at 14 genes were selected by all three methods (and actually 3 CpG sites were selected at the MEST gene, 2 sites were selected at the RASSF1 gene, and 2 sites were selected at the ISL1 gene, so the overlap is slightly more than 14). For the expression set, 10 genes were selected by all three methods. The high degree of overlap among the top 25 genes indicates that spLRMA, PMD, and SELP-I are ranking similar groups of variables as important. Among the rest of the genes, there is more overlap between PMD and SELP-I than between PMD and spLRMA or between SELP-I and spLRMA. As a consequence, there are more genes selected uniquely by spLRMA – a pattern that is more apparent in the expression set than in the methylation set. Given that the objectives of PMD and SELP-I are more similar to each other than either is to spLRMA, it is not surprising that the greatest amount of overlap (beyond that for all three methods) occurs between PMD and SELP-I.



(i) Methylation set (CpG sites)

(ii) Gene Expression set

Figure 3.6: Overlap of top 25 genes selected by spLRMA, PMD, and SELP-I.

We next examine how well the top-ranked variables rate with respect to achieving a high canonical correlation. In Figure 3.7, we show the estimated canonical correlation based on non-sparse CCA using the top-ranked variables from the methylation and expression sets. We show how the estimated canonical correlation changes as we change the threshold for the number of top-ranked variables included in CCA. The x-axis in Figure 3.7 shows the number of top-ranked variables included from each set, which we vary from 5 to 50. Thus, the minimum number of total variables we include in CCA is 10 and the maximum is 100. spLRMA performs competitively with PMD and SELP-I over the entire range considered. In fact, the estimated canonical correlation for the variables selected by spLRMA is actually higher than the corresponding value for PMD or SELP-I over a large portion of the range. That fact is rather surprising given that spLRMA selects variables by optimizing an objective function that is not directly related to maximizing the canonical correlation.

Our analysis of the breast cancer data was primarily meant to illustrate that, despite not being a sparse CCA methodology, spLRMA can be a highly effective tool for one of the main goals of sparse CCA – variable selection. Our final step in the analysis would be to interpret our findings with respect to the underlying biology; that is, we desire to describe how the methylation levels at the top-ranked CpG sites are associated with the expression of the top-ranked genes, and what role that association plays in the development of cancers. Some genes, such as RASSF1, are well known to function as tumor suppressors, so silencing through hypermethylation has a clear role in cancer development. However, the functions of many of the top-ranked genes from both the methylation and expression sets have not been fully described, so the judgment of a subject-matter expert would be necessary to make an assessment about any potential findings regarding the association between DNA methylation and gene expression.

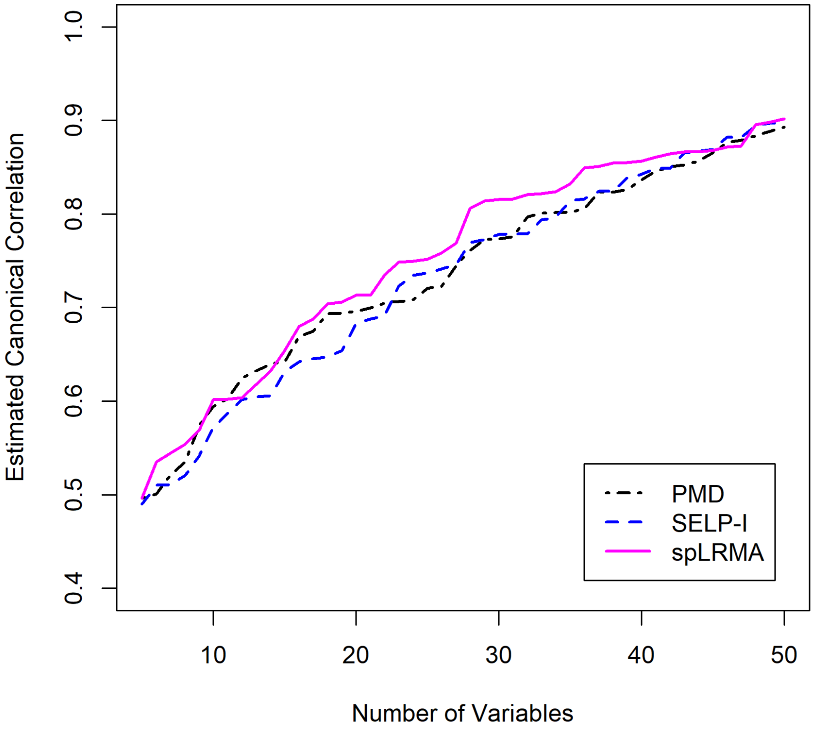


Figure 3.7: Estimated canonical correlation based on non-sparse CCA using the top-ranked variables selected from the methylation and expression sets. The x-axis indicates the number of top variables included from each set (so the total number of variables included in CCA is twice the value on the x-axis).

3.5 Discussion

We developed a methodology that describes the dominant modes of co-variation between variables in two datasets while simultaneously performing variable selection. We studied certain aspects of the method's variable selection properties theoretically in Section 3.2.2, through simulation in Section 3.3, and empirically in 3.4. In addition, in Section 3.2.4 we proposed two metrics to measure a variable's importance and a procedure to rank variables according to their importance. The procedure avoids tuning parameter selection and is applicable to both the proposed method and related methods, such as sparse CCA. In principle, the procedure could be adapted to other analyses (e.g., regression) where variable selection is a primary goal and regularization is employed to achieve the variable selection property.

In Section 3.2.2, we showed that, under some assumptions, the proposed method and certain sparse CCA methods aim to select the same set of variables. In our simulation experiment in Section 3.3, we generated data under a sparse CCA model and showed that the proposed method performed as well as or better than two sparse CCA methods with respect to variable selection accuracy (as measured by the average rank of the signal variables). In our real data analysis in Section 3.4, we provided further evidence for the close relationship between the proposed method and sparse CCA. For real data, neither the assumptions of Section 3.2.2 are likely to hold, nor do we know the identities of the true signal variables (or if there even is a concept of signal vs. noise variables). Nevertheless, we were able to show that the proposed method and the two sparse CCA methods rank similar subsets of variables as important, and that the most highly ranked variables have high canonical correlation if we perform non-sparse CCA with them.

Although we showed that the proposed method and sparse CCA aim to select the same set of signal variables, a natural question is: Why does the proposed method in some scenarios actually perform *better* than sparse CCA methods with respect to variable selection accuracy? Because the data were generated under a sparse CCA model, and the proposed method does not optimize the CCA objective function, one might expect methods developed as sparse versions of CCA to always perform the best. One possible explanation is that we have relaxed the objectives of CCA to focus on a single aspect of the problem. Rather than simultaneously maximizing the canonical correlation and performing variable selection, our formulation of the problem foregoes estimation of the canonical correlation and instead emphasizes variable selection. Given the difficulty of accurate estimation of the canonical correlation in high dimensional settings, it makes sense that relaxing that aspect of the analysis could result in better performance for other aspects.

Another possible explanation for the proposed method's better performance is that the proposed method only attempts to select variables from one dataset at a time, whereas the two sparse CCA methods – PMD and SELP-I – select variables from both datasets at the same time. Because of the additional challenge in selecting variables from multiple datasets simultaneously, the proposed method perhaps enjoys an unfair advantage by considering variable selection in each dataset as separate problems. We plan to address that possibility in future work. We aim to develop an algorithm to solve problem (3.4), which would select variables from both datasets as part of a single optimization.

In our simulation experiment, we designed some of the scenarios to highlight some of the major challenges for CCA in high dimensions. Two scenarios use within-set covariance structures $\Sigma_{XX} = I_p$ and $\Sigma_{YY} = I_q$. The PMD method of Witten et al. (2009) and the SELP-I method of Safo et al. (2018) make such an assumption; that is, both methods assume the correct model generating the data. One might expect a method to perform better when its assumptions match the underlying model. Counterintuitively, the simulation results in Section 3.3.1 indicate that the methods actually perform *worse* for scenarios in which the methods' assumptions hold. Why then do the methods perform poorly when their assumptions hold and perform better when their assumptions are violated?

Comparing the true between-set covariance matrix Σ_{XY} to the observed sample matrix $\widehat{\Sigma}_{XY}$ lends some insight into that question. Figures 5.15–5.18 in the Appendix show heatmaps comparing the true Σ_{XY} to several realizations of the observed $\widehat{\Sigma}_{XY}$ for selected simulation scenarios listed in Table 3.1. When the signal variables have high within-set correlation (e.g., Fig. 5.15), the region of $\widehat{\Sigma}_{XY}$ corresponding to the nonzero part of Σ_{XY} is readily apparent. In contrast, when the signal variables have little (e.g., Figs. 5.16 and 5.17) or no (e.g., Fig.

5.18) within-set correlation, the region is barely detectable. The nonzero region of Σ_{XY} corresponds to the signal variables that PMD, SELP-I, and the proposed method aim to select, and the extent to which that region is apparent in the sample matrix $\widehat{\Sigma}_{XY}$ reflects the level of difficulty of each simulation scenario. Thus, although the assumptions of PMD and SELP-I are satisfied in scenarios that set $\Sigma_{XX} = I_p$ and $\Sigma_{YY} = I_q$, those are actually the most difficult scenarios with respect to detecting the signal variables, and all of the methods perform poorly. Conversely, the assumptions of PMD and SELP-I are violated for scenarios in which the signal variables are highly correlated, but those are the easiest scenarios with respect to detecting the signal variables, and so all of the methods perform well.

Another challenge for analyses involving high dimensional datasets is the potential for spurious findings. The null scenario of the simulation experiment was designed to determine whether the methods detected structure in the data arising from spurious sources. Under the null scenario, the true between-set covariance matrix was $\Sigma_{XY} = 0$, but a group of variables in each dataset had high within-set correlation. As in the other scenarios, 10% of the variables within a dataset were correlated among themselves. Although none of the variables correlated among themselves can accurately be called "signal" variables (because the canonical vectors are zero), all three methods tended to rank those variables as more important than the others (see Fig. 3.4). Thus, all three methods responded to dependence structure in $\widehat{\Sigma}_{XY}$ arising from blocks of correlated variables in Σ_{XX} and Σ_{YY}, even though the true structure of Σ_{XY} was null.

Again, inspection of heatmaps of the observed $\widehat{\Sigma}_{XY}$ sheds some light into why the methods are susceptible to spurious signal. Figure 5.19 in the Appendix shows four realizations of $\widehat{\Sigma}_{XY}$ under the null scenario. In some realizations [such as Fig. 5.19(iii)], the observed $\widehat{\Sigma}_{XY}$ contains a region of apparent "signal" that is virtually indistinguishable from scenarios in which the canonical vectors are non-null. Thus, all of the methods have a tendency to select the variables corresponding to that region. The phenomenon is analogous to observing large, spurious correlations in the context of high dimensional regression, as discussed in Fan and Lv (2008) and Fan and Lv (2010). However, rather than observing a large, spurious correlation between a single variable in X and a single variable in Y, we tend to observe an entire block of spurious of correlations in $\widehat{\Sigma}_{XY}$ due to the high degree of correlation among the group of variables within each set.

Although all three methods ranked the spurious signal variables as more important than the other noise variables, spLRMA did so to a greater extent (and with less variability). Figure 5.20 in the Appendix helps explain why. It compares the null scenario with three other scenarios with respect to several measures associated with the observed $\widehat{\Sigma}_{XY}$. It shows the measures for elements/columns corresponding to the signal variables, and also an equal number of noise variables (for reference). Note that similar results for the rows could be obtained, but the columns are sufficient to illustrate the idea. One measure is the Frobenius norm of the upper left block of $\widehat{\Sigma}_{XY}$, which serves as a measure of the strength of the signal (i.e., how dark the block is). For the null scenario, there is very large variability, which is consistent with Figure 5.19 (namely, sometimes the block isn't dark at all, sometimes it's very dark). That helps explain why PMD and SELP-I rank the spurious signal variables highly, but have large variability, but it doesn't help explain why spLRMA is far less variable. The other two measures help explain the results for spLRMA. One measure is the average ℓ_2 norm of the columns of $\widehat{\Sigma}_{XY}$ corresponding to the signal variables. Since the objective function for spLRMA penalizes the ℓ_2 norm, if it is large on average for the spurious signal variables, then spLRMA will tend to rank them highly. We see that the ℓ_2 norm is indeed large on average in the null scenario. The last measure tries to quantify the degree of linear dependence of the columns of $\widehat{\Sigma}_{XY}$ corresponding to the signal variables. We can take those columns, form a matrix A (which is just the $p \times q_{sig}$ left submatrix of $\widehat{\Sigma}_{XY}$), then calculate the condition number of A^TA, which serves as measure of the degree of linear dependence of the columns (with larger values indicating more linearly dependent). We see that the first few columns of $\widehat{\Sigma}_{XY}$ are more linearly dependent when the within-in set correlation is high, regardless of whether the canonical vectors are null or not. Since PMD, SELP-I, and spLRMA are all trying to find a low rank matrix close to $\widehat{\Sigma}_{XY}$, greater linear dependence among columns will cause the methods to rank the corresponding variables more highly.

The simulation scenarios with high within-set correlation (including the null scenario) highlight another issue related to the evaluation of the performances of new sparse CCA and related methodologies: Researchers should test their methods against more challenging simulation settings. The visualizations of the signal region of $\widehat{\Sigma}_{XY}$ in Figures 5.15–5.18, as well as the methods' tendencies to find spurious signal when groups of variables are highly correlated among themselves, suggest that the methods are responding more to structure in the data arising from the within-set correlation rather than structure arising from the sparsity pattern of the canonical vectors. When the groups of highly correlated variables match the sparsity pattern of the canonical vectors, the results can give the false impression that a method has very good performance. In reality, the good performance is essentially coincidence: The method is effective at detecting the signal variables not because it is inherently good at accurately estimating the sparsity pattern of the canonical vectors, but because the within-set correlation structure coincides with the sparsity pattern of the canonical vectors. This phenomenon is especially apparent in our scenario with within-set correlation $BD[AR(0.9),\ I)]$. All three methods seem to perform extremely well with respect to ranking the signal variables highly. However, the good performance is mainly an artifact resulting from the high correlation of the signal variables among themselves, as evidenced by the decline in performance for the $BD[AR(0.7),\ I)]$, $BD[AR(0.2),\ I)]$, and $BD[I,\ I)]$ scenarios. If researchers want to test their method's ability to detect the underlying sparsity pattern of the canonical vectors, the more challenging scenarios with weak within-set correlation provide more information toward that purpose.

Finally, we focused on the variable selection aspect of the proposed method for this work.

However, variable selection was only one of our primary goals. Our other main goal was to describe the dominant modes of co-variation between the variables in two datasets. Although the proposed method and sparse CCA aim to select similar sets of variables, the proposed method and CCA differ with respect to how they characterize the co-variability between two datasets. CCA captures one specific notion of co-variability: the correlation between $X\boldsymbol{u}$ and $Y\boldsymbol{v}$. In contrast, the proposed method provides a more general description of the dominant modes of co-variation. In fact, the proposed method is more similar to multivariate partial least squares (PLS) with respect to how it describes the modes of co-variation between two datasets [at least, the version of PLS described in McIntosh et al. (1996), Worsley (1997), and Lorenzi et al. (2018)]. Our approach may even be considered as a sparse version of PLS, similar to the method Chun and Keleş (2010) proposed. We intend to compare our method with PLS in another line of investigation. Rather than variable selection, we will turn our attention toward describing the dominant modes of co-variation, evaluating the methods both with respect to how well they capture the modes of co-variation and how to interpret them.

Chapter 4

Sparse Canonical Correlation Analysis as a Regularized Least Squares Problem with Quadratic Equality Constraints

4.1 Introduction

Let X denote a set of p variables and Y denote a different set of q variables, where both X and Y are centered. Canonical correlation analysis (CCA) (Hotelling, 1936) seeks to characterize the relationship between X and Y by finding linear combinations $X\boldsymbol{u}$ and $Y\boldsymbol{v}$ such that the correlation $\rho := Cor(X\boldsymbol{u},\ Y\boldsymbol{v})$ is maximized. The vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ of linear coefficients are called the canonical vectors and the correlation ρ is called the canonical correlation.

Let $\Sigma = \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix}$ denote the population covariance of $(X,\ Y)$. The objective of CCA can be expressed mathematically as

$$\rho \ = \ \max_{\boldsymbol{u},\boldsymbol{v}} \ \frac{\boldsymbol{u}^T\Sigma_{XY}\boldsymbol{v}}{\sqrt{\boldsymbol{u}^T\Sigma_{XX}\boldsymbol{u}}\sqrt{\boldsymbol{v}^T\Sigma_{YY}\boldsymbol{v}}}. \tag{4.1}$$

The objective function (4.1) is invariant to scaling, so (4.1) can be equivalently expressed as

$$\rho \;=\; \max_{\boldsymbol{u},\boldsymbol{v}} \; \boldsymbol{u}^T \Sigma_{XY} \boldsymbol{v} \quad \text{s.t. } \boldsymbol{u}^T \Sigma_{XX} \boldsymbol{u} = \boldsymbol{v}^T \Sigma_{YY} \boldsymbol{v} = 1.$$

Replacing Σ with its sample equivalent, an estimate of the canonical correlation defined by (4.1) can be obtained from the singular value decomposition (SVD) of the matrix

$$K \;:=\; \hat{\Sigma}_{XX}^{-1/2} \hat{\Sigma}_{XY} \hat{\Sigma}_{YY}^{-1/2}.$$

The estimate of the canonical correlation $\hat{\rho}$ is given by the largest singular value of K. The corresponding left- and right-singular vectors give $\hat{\Sigma}_{XX}^{1/2} \hat{\boldsymbol{u}}$ and $\hat{\Sigma}_{YY}^{1/2} \hat{\boldsymbol{v}}$, so the estimates of the canonical vectors can be obtained by transformation. One may find additional canonical correlations $\hat{\rho}_2, \ldots, \hat{\rho}_{\min(p,q)}$ from the other nonzero singular values of K, but we focus on the situation where one is only interested in the largest canonical correlation.

For many modern applications of CCA, the datasets corresponding to X and Y consist of many more variables than observations (e.g., microarray studies in genetics). When the number of variables in either X or Y exceeds the sample size, the inverses $\hat{\Sigma}_{XX}^{-1/2}$ and/or $\hat{\Sigma}_{YY}^{-1/2}$ do not exist, so it is not possible to perform CCA via the SVD of K. Vinod (1976) proposed a ridge correction to address the invertibility issue, so that the matrix K is computed using the inverses of $\hat{\Sigma}_{XX} + \lambda_x I_p$ and $\hat{\Sigma}_{YY} + \lambda_y I_q$. One could also apply more extreme regularization, such as extracting the diagonals of $\hat{\Sigma}_{XX}$ and $\hat{\Sigma}_{YY}$ (so that K is the sample correlation matrix between X and Y) or assuming $\Sigma_{XX} = I_p$ and $\Sigma_{YY} = I_q$ (so that K is the sample covariance matrix between X and Y).

Although methods that directly address the invertibility of $\hat{\Sigma}_{XX}$ and $\hat{\Sigma}_{YY}$ yield a solution for CCA, the solution itself is often difficult to interpret. Interpretation of the results from CCA focuses on the magnitudes of the loadings in $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{v}}$. Variables with loadings that are large in magnitude relative to the others are considered more important (in some sense).

However, it is often unclear what should be considered a "large" loading or where a threshold should be set to distinguish important variables from relatively unimportant variables. One approach to overcome those difficulties and improve the interpretability of the results is to invoke the principle of sparsity: We assume that only a small subset of the variables in X and Y are relevant to the analysis, and the rest do not contribute substantially. By finding sparse estimates of the canonical vectors, the results can be interpreted with respect to the subset of variables in X and Y with nonzero coefficients, while the variables with zero coefficients are considered unimportant.

Several sparse CCA proposals have been put forward. One group of proposals exploits the power method of calculating the SVD of a matrix by incorporating a thresholding step in the SVD of K to achieve a sparse solution to CCA. Witten et al. (2009) calculated the sample correlation matrix $\hat{R}_{XY} := [\text{diag}(\hat{\Sigma}_{XX})]^{-1/2}\ \hat{\Sigma}_{XY}\ [\text{diag}(\hat{\Sigma}_{YY})]^{-1/2}$, then applied the power method to the SVD of $\hat{R}_{XY}$ with a thresholding step derived from a bound on the ℓ_1 norm of the canonical vectors. Note that performing CCA with $\hat{R}_{XY}$ is equivalent to standardizing the variables and assuming $\Sigma_{XX} = I_p$ and $\Sigma_{YY} = I_q$. Parkhomenko et al. (2009) took a similar approach as Witten et al. (2009), but derived the thresholding step from the Lagrange form of the ℓ_1 constraint. Chalise and Fridley (2012) extended the methods of Parkhomenko et al. (2009) by considering additional penalty functions and combining with a variable filtering procedure proposed by Zhou and He (2008). Chen et al. (2013) estimated the precision matrices $\hat{\Sigma}_{XX}^{-1}$ and $\hat{\Sigma}_{YY}^{-1}$ directly by assuming a particular form of either the precision matrices (e.g., sparse) or covariance matrices (e.g., Toeplitz), then applied the power method with hard thresholding.

Other proposals cast CCA into the regression framework by rewriting the bilinear form of the objective function as a least squares form. Waaijenborg et al. (2008) proposed to solve the least squares problem as two separate regression problems, each with an elastic net penalty (Zou and Hastie, 2005). However, they approximated the elastic net penalty by univariate soft thresholding, so their method is very similar to the methods of Parkhomenko et al. (2009) and Witten et al. (2009). Wilms and Croux (2015) took an alternating regression approach, where they applied a LASSO penalty (Tibshirani, 1996) to each regression subproblem.

More recent proposals consider the more general class of problems called generalized eigenvalue problems (GEPs), of which CCA is a special case. See Safo et al. (2018) or Jung et al. (2019) for sparse CCA methods based on the GEP formulation.

Following Waaijenborg et al. (2008) and Wilms and Croux (2015), we solve CCA within the regression framework. We propose two sparse CCA algorithms, each based on a type of proximal algorithm. First, we rewrite the CCA objective function as minimization of a quadratic form. We then solve a regularized version of the optimization problem that yields sparse estimates of the canonical vectors. We consider the LASSO (Tibshirani, 1996), SCAD (Fan and Li, 2001), and MC (Zhang, 2010) penalties, all of which result in a thresholding step as a consequence of their proximal operators. We compare the proposed method to methods based on alternating minimization.

4.2 Proposed Methodology

4.2.1 Minimization of a Quadratic Form

For a sample of n observations, Izenman (1975) characterizes CCA as a prediction problem by expressing the objective function as

$$\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}} \;=\; \underset{\boldsymbol{u},\boldsymbol{v}}{\arg\min}\;\; (X\boldsymbol{u} - Y\boldsymbol{v})^T (X\boldsymbol{u} - Y\boldsymbol{v}) \quad \text{s.t. } \boldsymbol{u}^T \Sigma_{XX} \boldsymbol{u} = \boldsymbol{v}^T \Sigma_{YY} \boldsymbol{v} = 1. \tag{4.2}$$

For fixed $\boldsymbol{v}$, (4.2) is a least squares problem in $\boldsymbol{u}$ under quadratic equality constraints. Similarly, for fixed $\boldsymbol{u}$, (4.2) is a least squares problem in $\boldsymbol{v}$ under quadratic equality constraints. Based on that observation, Wilms and Croux (2015) proposed an alternating regression procedure similar to Wold's algorithm (Wold, 1968) to solve (4.2). To yield a sparse solution for the canonical vectors, they formulate a penalized version of the objective function,

$$\begin{aligned}\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}} &= \underset{\boldsymbol{u},\boldsymbol{v}}{\arg\min} \ (X\boldsymbol{u} - Y\boldsymbol{v})^T (X\boldsymbol{u} - Y\boldsymbol{v}) + \lambda_x ||\boldsymbol{u}||_1 + \lambda_y ||\boldsymbol{v}||_1 \\ &\quad \text{s.t. } \boldsymbol{u}^T \Sigma_{XX} \boldsymbol{u} = \boldsymbol{v}^T \Sigma_{YY} \boldsymbol{v} = 1,\end{aligned}$$

which they propose to solve by an alternating LASSO-regularized regression procedure. We take a similar approach, but we rewrite the objective function in (4.2) to facilitate simultaneous (as opposed to alternating) estimation of $(\boldsymbol{u}, \boldsymbol{v})$. We also consider additional penalties besides the ℓ_1 norm.

Let $A = (X, \ -Y)_{n\times(p+q)}$ and $\boldsymbol{w} = (\boldsymbol{u}^T, \ \boldsymbol{v}^T)^T$. We write a penalized version of (4.2) as

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\arg\min} \ \frac{1}{2}\boldsymbol{w}^T A^T A \boldsymbol{w} + P_{\lambda_x,\lambda_y}(\boldsymbol{w}) \quad \text{s.t. } \boldsymbol{w}^T C_1 \boldsymbol{w} = \boldsymbol{w}^T C_2 \boldsymbol{w} = 1, \tag{4.3}$$

where C_1, C_2 are block diagonal matrices $C_1 = BD(\Sigma_{XX}, 0_{q\times q})$ and $C_2 = BD(0_{p\times p}, \Sigma_{YY})$ and $P_{\lambda_x,\lambda_y}(\boldsymbol{w})$ is a penalty function. We consider three different penalties:

1. LASSO (Tibshirani, 1996)

2. SCAD (Fan and Li, 2001)

3. MC (Zhang, 2010)

For example, for the LASSO penalty, $P_{\lambda_x,\lambda_y}(\boldsymbol{w}) = ||F\boldsymbol{w}||_1$, where $F = BD(\lambda_x I_p, \ \lambda_y I_q)$. Then (4.3) is identical to the penalized objective function defined by Wilms and Croux (2015) (besides scaling by the factor 1/2) because $||F\boldsymbol{w}||_1 = \lambda_x ||\boldsymbol{u}||_1 + \lambda_y ||\boldsymbol{v}||_1$. See the references listed for the form of $P_{\lambda_x,\lambda_y}(\boldsymbol{w})$ for the other penalties.

Without the penalty term, the objective function (4.3) involves minimization of a quadratic form under quadratic constraints. The unconstrained minimization of such a quadratic form would yield $\hat{\boldsymbol{w}} = \mathbf{0}$ for any matrix A. The additional constraints $\boldsymbol{w}^T C_1 \boldsymbol{w} = \boldsymbol{w}^T C_2 \boldsymbol{w} = 1$ ensure that the solution is not trivially the zero vector; however, the constraints also cause the optimization problem to be non-convex.

4.2.2 Proximal Algorithms

The motivation for writing the penalized objective function in the form of (4.3) is to devise an algorithm that makes use of proximal operators. A general class of algorithms that make use of proximal operators are called proximal algorithms and are often applied to optimization problems of the form

$$\text{minimize } f(x) + g(x),$$

where f and g are convex functions and f is differentiable. A proximal algorithm typically splits the minimization of $f+g$ into two subproblems by minimizing f and g separately. The proximal operator is applied to the function g, and for some algorithms, to the function f as well. The proximal operator is related to the Moreau-Yosida regularization of a function. For a function f, it is defined as

$$\mathbf{prox}_{\frac{1}{\eta}f}(v) := \arg\min_{x} \; f(x) + \frac{\eta}{2}||x - v||_2^2.$$

The parameter $\eta > 0$ controls the trade-off between minimizing f and staying close to v. See Parikh and Boyd (2014) for a detailed discussion of proximal operators and their applications in function optimization.

We devise two proximal algorithms for solving problem (4.3). The first is based on the proximal gradient method. The proximal gradient method is an iterative algorithm with updates

$$x^{k+1} = \mathbf{prox}_{\gamma^k g}(x^k - \gamma^k \nabla f(x^k)),$$

where $\mathbf{prox}_g$ denotes the proximal operator of the function g and γ^k is a step size. Following the general approach of a proximal algorithm, the proximal gradient method splits the minimization of $f + g$ into two subproblems. The term $x^k - \gamma^k \nabla f(x^k)$ is simply the gradient descent method for minimizing f. The term $\mathbf{prox}_g(x)$ minimizes g with an additional regularity term that ensures the solution lies close to the argument x. Thus, the update $x^{k+1} = \mathbf{prox}_{\gamma^k g}(x^k - \gamma^k \nabla f(x^k))$ can be interpreted as a compromise between minimizing g and staying close to a gradient descent step for minimizing f.

For the optimization problem defined by (4.3), we split the objective as $f(\boldsymbol{w}) := \frac{1}{2}\boldsymbol{w}^T A^T A \boldsymbol{w}$ and $g(\boldsymbol{w}) := P_{\lambda_x,\lambda_y}(\boldsymbol{w})$. Then the general form of the proximal gradient method applied to problem (4.3) is

$$\boldsymbol{w}^{k+1} = \mathbf{prox}_{\gamma^k g}(\boldsymbol{w}^k - \gamma^k A^T A \boldsymbol{w}^k).$$

The proximal operators of the three penalty functions we consider have closed-form solutions. Moreover, the proximal operators each involve element-wise thresholding of the vector $\boldsymbol{w}^k - \gamma^k A^T A \boldsymbol{w}$, so the final solution will be sparse, with the level of sparsity depending on the tuning parameters λ_x and λ_y. The proximal operators of the three penalty functions we consider are summarized in Table 4.1.

Table 4.1: Proximal operators of the LASSO, SCAD, and MC penalty functions. Each thresholding operator is applied element-wise to the vector $\boldsymbol{w}$ and $\lambda_x = \lambda_y := \lambda$ is assumed for notational convenience.

Penalty Function	**Proximal Operator**	**Additional Notes**
LASSO	$\hat{w} = \text{sgn}(w)(\lvert w\rvert - \lambda)_+$	The $(\cdot)_+$ operator is defined as: $(\cdot)_+ = \max(0, \cdot)$.
SCAD	$\hat{w} = \begin{cases} \text{sgn}(w)(\lvert w\rvert - \lambda)_+ & \lvert w\rvert \leq 2\lambda \\ \frac{(a-1)w - \text{sgn}(w)a\lambda}{a-2} & 2\lambda < \lvert w\rvert \leq a\lambda \\ w & \lvert w\rvert > a\lambda \end{cases}$	We use $a = 3.7$.
MC	$\hat{w} = \text{sgn}(w)\min\left\{\lvert w\rvert, \frac{a(\lvert w\rvert-\lambda)_+}{a-1}\right\}$	We use $a = 3$.

Although each component in the splitting of problem (4.3) is a convex function, the overall problem is non-convex. The non-convexity is due to the quadratic equality constraints involving $\boldsymbol{u}$ and $\boldsymbol{v}$ (i.e., $\boldsymbol{w}$). The updates in the standard proximal gradient method do not enforce those constraints, so we propose to follow each update with a normalization step. The full details of the proposed algorithm are given in Algorithm 6.

Algorithm 6: Proximal Gradient Method for Sparse CCA

```
Input: Centered data matrices $X$ and $Y$, initial estimate $\boldsymbol{w}^{(0)} = (\boldsymbol{u}^{(0)}; \boldsymbol{v}^{(0)})$ s.t.
       $||\boldsymbol{u}^{(0)}||_2 = ||\boldsymbol{v}^{(0)}||_2 = 1$, tuning parameters $\lambda_x$ and $\lambda_y$.
Output: Final estimates $\hat{\boldsymbol{u}}$, $\hat{\boldsymbol{v}}$, and $\hat{\rho} = Cor(X\hat{\boldsymbol{u}}, Y\hat{\boldsymbol{v}})$.

$A \leftarrow (X,\ -Y)$ // create A matrix
$\gamma^0 \leftarrow 1$ // initialize step size
$k \leftarrow 0$ // iteration counter

while objective criterion not met do
    $\boldsymbol{w}^{*(temp)} = (\boldsymbol{u}^{*(temp)}; \boldsymbol{v}^{*(temp)}) \leftarrow \mathbf{prox}_{\gamma^k g}(\boldsymbol{w}^k - \gamma^k A^T A \boldsymbol{w}^k)$
    // $\mathbf{prox}_g$ applied element-wise according to the penalty $g$ listed in Table 4.1

    $\boldsymbol{w}^{(temp)} \leftarrow \left(\frac{\boldsymbol{u}^{*(temp)}}{||\boldsymbol{u}^{*(temp)}||_2}; \frac{\boldsymbol{v}^{*(temp)}}{||\boldsymbol{v}^{*(temp)}||_2}\right)$ // normalize

    // the function $f$ is $f(\boldsymbol{w}) = 0.5 * \boldsymbol{w}^T A^T A \boldsymbol{w}$
    if $f(\boldsymbol{w}^{(temp)}) < f(\boldsymbol{w}^{(k)}) + \nabla f(\boldsymbol{w}^{(k)})^T(\boldsymbol{w}^{(temp)} - \boldsymbol{w}^{(k)}) + \frac{1}{2\gamma^k}||\boldsymbol{w}^{(temp)} - \boldsymbol{w}^{(k)}||_2^2$ then
        $\boldsymbol{w}^{(k+1)} \leftarrow \boldsymbol{w}^{(temp)}$ // accept update
        $\gamma^{k+1} \leftarrow 1.2 * \gamma^k$ // increase step size
    else
        $\boldsymbol{w}^{(k+1)} \leftarrow \boldsymbol{w}^{(k)}$ // reject update; perform line search for step size
        $\gamma^{k+1} \leftarrow 0.5 * \gamma^k$ // shrink step size
    end

    $k \leftarrow k + 1$ // update iteration counter
end

$(\hat{\boldsymbol{u}}; \hat{\boldsymbol{v}}) \leftarrow \boldsymbol{w}^{(final)}$
$\hat{\rho} \leftarrow Cor(X\hat{\boldsymbol{u}}, Y\hat{\boldsymbol{v}})$
```

The normalization step that we insert into the proximal gradient method is an ad hoc solution for handling the quadratic equality constraints of problem (4.3). We develop another algorithm based on the alternating direction method of multipliers (ADMM) to handle the constraints in a more rigorous way. Again keeping with the general approach of proximal algorithms, ADMM splits the minimization of a function $f + g$ by separately minimizing f and g. To accomplish this, the problem $\min\ f(x) + g(x)$ is rewritten as

$$\min\ f(x) + g(z) \ \text{ s.t. } \ x - z = 0.$$

The generic form of ADMM for the new formulation of the problem is

$$\begin{aligned} x^{n+1} &= \mathbf{prox}_{\frac{1}{\eta}f}(z^n - r^n) \\ z^{n+1} &= \mathbf{prox}_{\frac{1}{\eta}g}(x^{n+1} + r^n) \\ r^{n+1} &= r^n + x^{n+1} - z^{n+1}. \end{aligned}$$

The basic intuition of ADMM is to minimize one function while ensuring the new update is close to the update for minimizing the other function. The variable r accumulates the residuals for the difference $x - z$. As the algorithm progresses, x converges to a point close to minimizing f, z converges to a point close to minimizing g, and the residual term encourages x and z to converge to each other.

Using the same splitting as for the proximal gradient method (but explicitly incorporating the quadratic equality constraints), the updates for the ADMM algorithm for solving problem (4.3) can be written as

$$\begin{aligned}
\boldsymbol{w}^{n+1} &= \mathbf{prox}_{\frac{1}{\eta}f}(\boldsymbol{z}^n - \boldsymbol{r}^n) \\
&= \arg\min_{\boldsymbol{w}} \ \frac{1}{2}\boldsymbol{w}^T A^T A\boldsymbol{w} + \frac{\eta}{2}||\boldsymbol{w} - \boldsymbol{z}^n + \boldsymbol{r}^n||_2^2 \ \text{ s.t. } \ \boldsymbol{w}^T C_1 \boldsymbol{w} = \boldsymbol{w}^T C_2 \boldsymbol{w} = 1 \\
\boldsymbol{z}^{n+1} &= \mathbf{prox}_{\frac{\lambda}{\eta}g}(\boldsymbol{w}^{n+1} + \boldsymbol{r}^n) \\
\boldsymbol{r}^{n+1} &= \boldsymbol{r}^n + \boldsymbol{w}^{n+1} - \boldsymbol{z}^{n+1},
\end{aligned}$$

where we have again assumed $\lambda_x = \lambda_y := \lambda$ for notational convenience. As in the proximal gradient method, the proximal operator for the function λg is one of the thresholding operators listed in Table 4.1. Note that after convergence, the final solution for $\boldsymbol{w}$ is not sparse – it is only approximately sparse. However, the final solution for $\boldsymbol{z}$ is exactly sparse, so we take $(\hat{\boldsymbol{u}}; \hat{\boldsymbol{v}}) = \boldsymbol{z}^{(final)}$.

Writing the update for $\boldsymbol{w}$ as

$$\begin{aligned}
\boldsymbol{w}^{n+1} = \arg\min_{\boldsymbol{w}} \ & \frac{1}{2}\boldsymbol{w}^T(A^T A + \eta I)\boldsymbol{w} - \eta\boldsymbol{w}^T(\boldsymbol{z}^n - \boldsymbol{r}^n) + c \\
\text{s.t.} \quad & \boldsymbol{w}^T C_1 \boldsymbol{w} = \boldsymbol{w}^T C_2 \boldsymbol{w} = 1,
\end{aligned} \tag{4.4}$$

where c is a constant not involving $\boldsymbol{w}$, we can see that it is a quadratically constrained quadratic program (QCQP). Because the constraint is an equality, rather than an inequality, the problem is non-convex and many of the well-known off-the-shelf QCQP solvers cannot be used. Furthermore, we cannot apply a convex relaxation to the constraints, such as $\boldsymbol{w}^T C_1 \boldsymbol{w} \leq 1$ and $\boldsymbol{w}^T C_2 \boldsymbol{w} \leq 1$, because the solution to the problem would be the zero vector. Instead, we solve the $\boldsymbol{w}$ update by sequential quadratic programming (Kraft, 1988, 1994) using a solver available through the NLopt library (Johnson, 2019). In the context of the

ADMM algorithm, the final solution for $\boldsymbol{w}$ will satisfy the constraints exactly, while the final solution for $\boldsymbol{z}$ will satisfy the constraints approximately.

The stopping condition for the proximal gradient method can be based on a simple criterion, such as the change between iterations in the estimate or the objective value evaluated at the estimate. However, the stopping conditions for ADMM are more involved. The full details of the proposed ADMM algorithm are described in Algorithm 7.

Algorithm 7: ADMM algorithm for Sparse CCA

```
Input: Centered data matrices $X$ and $Y$, initial guesses $\boldsymbol{w}^0$ and $\boldsymbol{z}^0$, tuning
       parameters $\lambda_x$ and $\lambda_y$, absolute tolerance $\epsilon^{abs}$, relative tolerance $\epsilon^{rel}$,
       augmented Lagrangian parameter $\eta$.
Output: Final estimates $\hat{\boldsymbol{u}}$, $\hat{\boldsymbol{v}}$, and $\hat{\rho} = Cor(X\hat{\boldsymbol{u}}, Y\hat{\boldsymbol{v}})$.

$A \leftarrow (X, \ -Y)$ // create A matrix
$\boldsymbol{r}^0 \leftarrow \boldsymbol{0}$ // initialize residual vector
$\epsilon^{primal} \leftarrow \sqrt{p_x + p_y} * \epsilon^{abs}$ // initialize primal tolerance
$\epsilon^{dual} \leftarrow \sqrt{p_x + p_y} * \epsilon^{abs}$ // initialize dual tolerance
$res^{primal} \leftarrow \epsilon^{primal} * 1.1$ // initialize primal error
$res^{dual} \leftarrow \epsilon^{dual} * 1.1$ // initialize dual error
$k \leftarrow 0$ // iteration counter

while $res^{primal} > \epsilon^{primal}$ or $res^{dual} > \epsilon^{dual}$ do

    // $\boldsymbol{w}$-update (solved by sequential quadratic programming)
    $\boldsymbol{w}^{k+1} \leftarrow \arg\min_{\boldsymbol{w}} \ \frac{1}{2}\boldsymbol{w}^T A^T A \boldsymbol{w} + \frac{\eta}{2}||\boldsymbol{w} - \boldsymbol{z}^k + \boldsymbol{r}^k||_2^2 \ s.t. \ \boldsymbol{w}^T C_1 \boldsymbol{w} = \boldsymbol{w}^T C_2 \boldsymbol{w} = 1$

    // $\boldsymbol{z}$-update
    $\boldsymbol{z}^{k+1} \leftarrow \mathbf{prox}_{\frac{\lambda}{\eta}g}(\boldsymbol{w}^{k+1} + \boldsymbol{r}^k)$
    // $\mathbf{prox}_{\lambda g}$ applied element-wise according to the penalty $\lambda g$ listed in
    Table 4.1

    // residual update
    $\boldsymbol{r}^{k+1} \leftarrow \boldsymbol{r}^k + \boldsymbol{w}^{k+1} - \boldsymbol{z}^{k+1}$

    $res^{primal} \leftarrow ||\boldsymbol{w}^{k+1} - \boldsymbol{z}^{k+1}||_2$
    $res^{dual} \leftarrow || - \eta(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)||_2$
    $\epsilon^{primal} \leftarrow \sqrt{p_x + p_y} * \epsilon^{abs} + \epsilon^{rel} * \max(||\boldsymbol{w}^{k+1}||_2, \ || - \boldsymbol{z}^{k+1}||_2)$
    $\epsilon^{dual} \leftarrow \sqrt{p_x + p_y} * \epsilon^{abs} + \epsilon^{rel} * ||\eta \boldsymbol{r}^{k+1}||_2$

    $k \leftarrow k + 1$ // update iteration counter

end

$(\hat{\boldsymbol{u}}; \hat{\boldsymbol{v}}) \leftarrow \boldsymbol{z}^{(final)}$
$\hat{\rho} \leftarrow Cor(X\hat{\boldsymbol{u}}, Y\hat{\boldsymbol{v}})$
```

Algorithm 7 requires an initial guess for both $\boldsymbol{w}$ and $\boldsymbol{z}$. From problem (4.4), we can see that the quadratic form in $\boldsymbol{w}$ involves a positive definite matrix, so the problem has a unique solution (up to signs). Consequently, the ADMM algorithm is robust to the initial guess for $\boldsymbol{w}$. The vector $\boldsymbol{z}$ can be (and typically is) initialized as the zero vector. However, if one desires to run the algorithm multiple times for a sequence of values of the tuning parameters λ_x and λ_y, it may be beneficial to warm start the algorithm to reduce computation time. In that case, one may use the final solution for $\boldsymbol{z}$ from a previous fit as the initial guess.

Remark 6. We note that when $P_{\lambda_x,\lambda_y}(\boldsymbol{w})$ is the LASSO penalty, the problem (4.3) is the same objective function considered by Wilms and Croux (2015). However, both the proximal gradient method and ADMM applied to problem (4.3) solve for $\boldsymbol{u}$ and $\boldsymbol{v}$ simultaneously at each iteration. Thus, the solutions from the proximal gradient and ADMM algorithms will not in general be the same as the solution from Wilms and Croux's alternating minimization algorithm (even with identical initial values $\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)}$). Part of our purpose in rewriting the objective to facilitate simultaneous estimation of the canonical vectors is that we hope the proposed algorithms will converge faster and result in a solution that achieves a better objective value, at least for "good" initial guesses $\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)}$ ("good" meaning closer to a global optimum). However, since the problem (4.3) is non-convex, neither the proximal gradient method, ADMM, nor alternating minimization is guaranteed to converge to a global optimum.

4.2.3 Extension to Multiple Datasets

The proposed methodology can also be extended to accommodate multiple datasets. Suppose there are $D > 2$ datasets $X_1 \ldots, X_D$ containing $p_1, \ldots, p_D$ variables, respectively. When there are multiple datasets, one may choose to perform separate analyses for each pair of datasets. However, in some contexts, it may be desirable to perform an integrated analysis instead of multiple, separate analyses. For example, one may desire to study the association between two datasets while controlling for the others (Iaci et al., 2010). In addition, separate analyses entail estimating $\binom{D}{2}$ canonical vectors, whereas an integrated analysis only involves estimating D canonical vectors. One way to perform an integrated analysis is to generalize the objective function in (4.2) by writing

$$\begin{aligned} \hat{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{u}}_D &= \underset{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_D}{\arg\min} \sum_{d<d'} ||X_d \boldsymbol{u}_d - X_{d'} \boldsymbol{u}_{d'}||_2^2 \\ & \qquad\qquad \text{s.t. } \boldsymbol{u}_1^T \Sigma_{X_1 X_1} \boldsymbol{u}_1 = \cdots = \boldsymbol{u}_D^T \Sigma_{X_D X_D} \boldsymbol{u}_D = 1. \end{aligned}$$

We can rewrite the objective function above by combining the matrices $X_1 \ldots, X_D$ into a single matrix A and concatenating the canonical vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D$ into a single vector $\boldsymbol{w}$. The objective function can then be expressed as minimization of a quadratic form, which we can solve with either the proposed proximal gradient algorithm or ADMM algorithm.

For illustration of how to rewrite the multi-set objective function as a quadratic form as in (4.3), consider the case where $D = 3$. We can write

$$A = \begin{pmatrix} X_1 & -X_2 & 0 \\ X_1 & 0 & -X_3 \\ 0 & X_2 & -X_3 \end{pmatrix}, \quad \boldsymbol{w} = \begin{pmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \end{pmatrix}.$$

Then

$$\boldsymbol{w}^T A^T A \boldsymbol{w} \;=\; ||X_1 \boldsymbol{u}_1 - X_2 \boldsymbol{u}_2||_2^2 + ||X_1 \boldsymbol{u}_1 - X_3 \boldsymbol{u}_3||_2^2 + ||X_2 \boldsymbol{u}_2 - X_3 \boldsymbol{u}_3||_2^2.$$

The penalty function and constraints in (4.3) are extended in a natural way.

4.2.4 Extension to Grouped Data

A regularized regression approach can also be useful for handling grouped data in CCA. For illustration, consider two groups of observations on two sets of variables X and Y. One would like to study the association between the variables in X and the variables in Y, but part of the association may differ depending on group. One could perform CCA on each group of observations separately, but the group differences may only occur for a small subset of the variables involved. Thus, one would like to use the information from all of the observations for estimating the shared association, but use group-specific information for estimating aspects of the association that differ between groups.

One could formulate the problem as minimization of a quadratic form, as in Section 4.2.1, with an additional fused LASSO-type penalty. Let $X_1 \in \mathbb{R}^{n_1 \times p}$, $Y_1 \in \mathbb{R}^{n_1 \times q}$ denote the data from the first group and $X_2 \in \mathbb{R}^{n_2 \times p}$, $Y_2 \in \mathbb{R}^{n_2 \times q}$ denote the data from the second group. Similarly, let $(\boldsymbol{u}_1,\ \boldsymbol{v}_1)$ denote the canonical vectors for the first group and $(\boldsymbol{u}_2,\ \boldsymbol{v}_2)$ denote the canonical vectors for the second group. Then, subject to appropriate size constraints on the canonical vectors, the objective function can be written as

$$\hat{\boldsymbol{w}}_1, \hat{\boldsymbol{w}}_2 = \underset{\boldsymbol{w}_1, \boldsymbol{w}_2}{\arg\min} \begin{pmatrix} \boldsymbol{w}_1^T & \boldsymbol{w}_2^T \end{pmatrix} \begin{pmatrix} A_1^T A_1 & 0 \\ 0 & A_2^T A_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{pmatrix}$$

$$+\ \lambda_x \sum_{j=1}^{p} |u_{1j} - u_{2j}|\ +\ \lambda_y \sum_{j'=1}^{q} |v_{1j'} - v_{2j'}|,$$

$$\text{where } \boldsymbol{w}_1 := \begin{pmatrix} \boldsymbol{u}_1 \\ \boldsymbol{v}_1 \end{pmatrix}, \quad \boldsymbol{w}_2 := \begin{pmatrix} \boldsymbol{u}_2 \\ \boldsymbol{v}_2 \end{pmatrix},$$

$$A_1^T A_1 := \begin{pmatrix} X_1^T X_1 & -X_1^T Y_1 \\ -Y_1^T X_1 & Y_1^T Y_1 \end{pmatrix},$$

$$A_2^T A_2 := \begin{pmatrix} X_2^T X_2 & -X_2^T Y_2 \\ -Y_2^T X_2 & Y_2^T Y_2 \end{pmatrix}.$$

When $\lambda_x = \lambda_y = 0$, the objective function is equivalent to performing CCA separately for each group of observations. When $\lambda_x > 0$ and $\lambda_y > 0$, the penalty term encourages some of the elements of $\boldsymbol{u}_1$ to be estimated the same as the corresponding elements in $\boldsymbol{u}_2$ and some of the elements of $\boldsymbol{v}_1$ to be estimated the same as the corresponding elements in $\boldsymbol{v}_2$. Elements estimated exactly the same are more likely to represent shared associations across the two groups of observations, while elements that are different are more likely to represent group-specific associations.

The size constraints for the canonical vectors are not as straightforward to specify as in sparse CCA. For example, consider taking λ_x sufficiently large so that $\hat{\boldsymbol{u}}_1 = \hat{\boldsymbol{u}}_2$. If one were to use the same constraints as Section 4.2.1, one cannot simultaneously satisfy $\hat{\boldsymbol{u}}_1 = \hat{\boldsymbol{u}}_2$ and $\hat{\boldsymbol{u}}_1^T \widehat{\Sigma}_{X_1X_1} \hat{\boldsymbol{u}}_1 = \hat{\boldsymbol{u}}_2^T \widehat{\Sigma}_{X_2X_2} \hat{\boldsymbol{u}}_2 = 1$. One could relax the constraints as $\hat{\boldsymbol{u}}_1^T \hat{\boldsymbol{u}}_1 = \hat{\boldsymbol{u}}_2^T \hat{\boldsymbol{u}}_2 = 1$, but then the problem does not reduce to the usual CCA problem when $\lambda_x = \lambda_y = 0$. Another idea is to replace the group-specific sample covariance matrices in the constraints with weighted averages of the form $\widetilde{\Sigma}_{X_1X_1} := \alpha_{\lambda_x} \widehat{\Sigma}_{X_1X_1} + (1 - \alpha_{\lambda_x}) \widehat{\Sigma}_{XX}$ and $\widetilde{\Sigma}_{X_2X_2} := \alpha_{\lambda_x} \widehat{\Sigma}_{X_2X_2} + (1 - \alpha_{\lambda_x}) \widehat{\Sigma}_{XX}$, where $\alpha_{\lambda_x} = 1$ if $\lambda_x = 0$ and $\alpha_{\lambda_x} \to 0$ as $\lambda_x \to \infty$. Then, when $\lambda_x = \lambda_y = 0$, the problem reduces to the usual CCA problem. Similarly, as $\lambda_x \to \infty$, the matrix used for the constraints approaches the sample covariance matrix for the full set of observations. One function satisfying the requirements is $\alpha(\lambda_x) = e^{-\lambda_x}$. An ADMM algorithm similar the the one described in Section 4.2.2 could be used to optimize the objective function.

The formulation can be extended to accommodate more than two groups. The approach could also be combined with the sparse CCA formulation of Section 4.2.1, albeit at the cost of substantially increased computational complexity.

4.3 Simulation Experiments

We evaluate the proposed proximal gradient and ADMM algorithms in two separate simulation experiments. As in Chapter 3, we compare the proposed methods to existing approaches based on their performances according to the variable importance metrics described in Section 3.2.4.

4.3.1 Proximal Gradient Algorithm

For the proximal gradient algorithm, we focus on the high dimensional setting. We simulate $p = 200$ variables in the X dataset and $q = 160$ variables in the Y dataset with a sample size of $N = 150$. We use 90% sparsity, resulting in 20 signal variables in the X dataset and 16 signal variables in the Y dataset. We set the within-covariance matrices as $\Sigma_{XX} = BD[AR_{p_{signal}}(0.7), I_{p_{noise}}]$ and $\Sigma_{YY} = BD[AR_{q_{signal}}(0.7), I_{q_{noise}}]$, where BD means block-diagonal and AR denotes a first-order autoregressive structure. We use the single canonical pair model to define the between-covariance matrix as $\Sigma_{XY} = \rho\Sigma_{XX}\boldsymbol{u}\boldsymbol{v}^T\Sigma_{YY}$ with $\rho = 0.9$. We generate a different set of canonical vectors $(\boldsymbol{u}, \boldsymbol{v})$ for each simulated dataset. For the signal portion of the canonical vectors, we generate the value from $Unif(1, 2)$ and the sign from Bernoulli with $P(+) = P(-) = 0.5$. We set the noise portion of the canonical vectors to zero. We normalize the canonical vectors with respect to their corresponding covariance matrices prior to generating (X, Y) (i.e., we scale $\boldsymbol{u}$ and $\boldsymbol{v}$ by a factor of $1/\sqrt{\boldsymbol{u}^T\Sigma_{XX}\boldsymbol{u}}$ and

$1/\sqrt{\boldsymbol{v}^T\Sigma_{YY}\boldsymbol{v}}$, respectively).

We compare the proposed proximal gradient algorithm to four existing sparse CCA approaches and to the proposed spLRMA method from Chapter 3. We compare to Waaijenborg et al. (2008) and Wilms and Croux (2015) because they both approach sparse CCA from a regression framework, as we have. Waaijenborg et al. (2008) use univariate soft thresholding (UST), while Wilms and Croux (2015) use an alternating LASSO-penalized regression procedure. Note that, although Waaijenborg et al. describe the $\boldsymbol{u}$ and $\boldsymbol{v}$ updates in separate steps, their algorithm does not alternate between solving for $\boldsymbol{u}|\boldsymbol{v}$ and $\boldsymbol{v}|\boldsymbol{u}$. We also compare to the penalized matrix decomposition (PMD) approach proposed by Witten et al. (2009) and the sparse estimation with linear programming (SELP-I) approach proposed by Safo et al. (2018). For each method, we vary the sparsity tuning parameters over a 2-dimensional grid of values.

The proposed proximal gradient algorithm, the UST approach of Waaijenborg et al. (2008), and the alternating LASSO regression approach of Wilms and Croux (2015) all require an initial guess for the estimated canonical vectors. Because the quality of the results may depend on the closeness of the initial guess to the global solution, we will investigate several different approaches for choosing the initial value. Using several different initial guesses will also allow us to evaluate the sensitivity of the different methods to the choice of initial value.

We use four different approaches to set the initial guess for the canonical vectors. For the first and second approaches, we obtain $(\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)})$ as the left and right singular vectors from the SVD of the sample covariance and sample correlation matrices of X and Y. The former assumes the variables within sets are uncorrelated and have variance one, while the latter only assumes the variables within sets are uncorrelated. For the third approach, we add a ridge correction to make the estimated within-set covariance matrices invertible; that is, we take $(\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)})$ to be the left and right singular vectors from the SVD of $K := \widetilde{\Sigma}_{XX}^{-1/2}\widehat{\Sigma}_{XY}\widetilde{\Sigma}_{YY}^{-1/2}$, where $\widetilde{\Sigma}_{XX} = \widehat{\Sigma}_{XX} + \sqrt{\frac{\log p}{n}} I_p$ and $\widetilde{\Sigma}_{YY} = \widehat{\Sigma}_{YY} + \sqrt{\frac{\log q}{n}} I_q$. For the fourth approach, we set $(\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)})$ to be the true population canonical vectors plus a small amount of noise. The true canonical vectors should be closer to the global solution than the initial guesses from the other approaches (on average).

Results

Tables 4.2 and 4.3 summarize the mean rank and standard deviation (SD) for the signal variables and noise variables, respectively, where the rank was determined by the importance measure based on whether the canonical vector loading is nonzero. To obtain the values reported in the tables, we first calculated the mean over the group of variables (signal or noise) for each simulated dataset, and then calculated the mean and SD over the 100 simulations. Note that the mean is equivalent to calculating the mean over both the group of variables and the simulations, but the SD is not. In Table 4.2 lower values indicate better performance of a method (because "1" corresponds to the most important variable, and we want the signal variables to be ranked as important), while in Table 4.3 higher values indicate better performance of a method (because we want the noise variables to be ranked as relatively unimportant).

Excluding the results that used the true canonical vector + noise as the initial value, PMD, SELP-I, spLRMA, and UST performed better than the methods of Wilms and Croux (2015) and the proposed proximal gradient method. The spLRMA method performed slightly better than PMD, SELP-I, and UST, which is consistent with the simulation results reported in Chapter 3. Comparing the method of Wilms and Croux (2015) with the proposed proximal gradient method, we can see that the proposed method performed better for all three penalty functions, with the largest improvement obtained for the MC and SCAD penalties. Comparing the proposed method across the different penalty functions, we can see that MC and SCAD performed better than LASSO, with SCAD achieving a 10 point improvement in the mean rank of the signal variables in many cases. Thus, simultaneous estimation of the canonical vectors via the proximal gradient method appears to work better than alternating minimization, and the MC and SCAD penalties appear to work better than the LASSO penalty.

For the methods that depend on the initial value, every method compared benefited from using the true canonical vector + noise as the initial guess, with the degree of improvement depending on the method. For UST, the gain was slight, with a less than 1 point improvement in the mean rank of the signal variables. For the method of Wilms and Croux (2015), the gain was larger, with a 4–6 point improvement in the mean rank of the signal variables. The proposed proximal gradient method achieved the largest gains. In fact, the proposed method with MC or SCAD penalty achieved nearly perfect variable ranking because under a perfect ranking, the mean of the ranks in the X set would be $\frac{1}{20}(1 + \cdots + 20) = 10.5$ and the mean of the ranks in the Y set would be $\frac{1}{16}(1 + \cdots + 16) = 8.5$. On the one hand, the results indicate that the proximal gradient algorithm can achieve a better solution than can UST or alternating minimization when started from an initial value close to the global optimum. On the other hand, the proximal gradient algorithm appears to be more sensitive to the initial value than the other methods. Since one cannot guarantee an initial value close to the global optimum in practice, the cost may outweigh the benefit.

Table 4.2: Mean (SD) rank of signal variables in the X and Y datasets according to the importance measure based on whether the canonical vector loading was nonzero. Results based on 100 simulations. Lower values of the rank are better. DNF: Did Not Finish.

Dataset	X variables				Y variables			
Initial Value†	1	2	3	4	1	2	3	4
PMD (Witten et al. 2009)	37.98 (15)				25.71 (11)			
SELP-I (Safo et al. 2018)	38.92 (13)				26.97 (12)			
spLRMA	34.88 (14)				23.7 (11)			
UST (Waaijenborg et al. 2008)	38.83 (13)	38.45 (13)	DNF	37.88 (12)	26.8 (12)	26.7 (12)	DNF	26.08 (11)
Wilms and Croux (2015)	70.56 (16)	70.09 (15)	70.19 (16)	64.72 (13)	50.59 (12)	50.73 (11)	53.85 (14)	46.53 (9)
Proximal Gradient - LASSO	63.43 (13)	62.77 (13)	67.03 (15)	17.78 (5)	46.74 (11)	46.29 (11)	48.47 (13)	13.15 (3)
Proximal Gradient - MCP	54.82 (13)	54.45 (13)	DNF	10.65 (1)	39.22 (11)	39.23 (12)	DNF	8.57 (0)
Proximal Gradient - SCAD	51.31 (13)	51.32 (13)	DNF	10.77 (1)	36.79 (12)	36.52 (12)	DNF	8.63 (0)

†Initial values: 1–SVD of sample covariance matrix, 2–SVD of sample correlation matrix, 3–ridge CCA, 4–true canonical vector + noise

Table 4.3: Mean (SD) rank of noise variables in the X and Y datasets according to the importance measure based on whether the canonical vector loading was nonzero. Results based on 100 simulations. Higher values of the rank are better. DNF: Did Not Finish.

Dataset	X variables				Y variables			
Initial Value†	1	2	3	4	1	2	3	4
PMD (Witten et al. 2009)	107.45 (2)				86.59 (1)			
SELP-I (Safo et al. 2018)	107.34 (1)				86.45 (1)			
spLRMA	107.79 (2)				86.81 (1)			
UST (Waaijenborg et al. 2008)	107.35 (1)	107.39 (1)	DNF	107.46 (1)	86.47 (1)	86.48 (1)	DNF	86.55 (1)
Wilms and Croux (2015)	103.83 (2)	103.88 (2)	103.87 (2)	104.48 (1)	83.82 (1)	83.81 (1)	83.46 (2)	84.27 (1)
Proximal Gradient - LASSO	104.62 (1)	104.69 (1)	104.22 (2)	109.69 (1)	84.25 (1)	84.3 (1)	84.06 (1)	87.98 (0)
Proximal Gradient - MCP	105.58 (1)	105.62 (1)	DNF	110.48 (0)	85.09 (1)	85.09 (1)	DNF	88.49 (0)
Proximal Gradient - SCAD	105.97 (1)	105.96 (1)	DNF	110.47 (0)	85.36 (1)	85.39 (1)	DNF	88.49 (0)

†Initial values: 1–SVD of sample covariance matrix, 2–SVD of sample correlation matrix, 3–ridge CCA, 4–true canonical vector + noise

Tables 4.4 and 4.5 summarize the mean rank and standard deviation (SD) for the signal variables and noise variables, respectively, where the rank is determined by the importance measure based on the magnitude of the canonical vector loading. Qualitatively, the differences among methods are similar to the results in Tables 4.2 and 4.3. Quantitatively, the most obvious difference in the results is the performance of the proximal gradient method relative to Wilms and Croux's (2015) method, and the relative performances of the different penalty functions. Although the proximal gradient method with MC or SCAD penalty performs better than Wilms and Croux's method in all cases, Wilms and Croux's method does slightly better than the proximal gradient method with LASSO penalty in several cases (i.e., depending on the initial value). In addition, the improvement in the mean rank of the signal variables gained by using MC or SCAD penalty is not as large as when using the other importance measure. Most of the differences can be attributed to the fact that, for Wilms and Croux's method and the proximal gradient method with LASSO penalty, the mean rank of the signal variables is better when using the importance measure based on the magnitude of the canonical vector loading, while the mean rank of the signal variables is the same or slightly worse for the proximal gradient method with MC or SCAD penalty.

Table 4.4: Mean (SD) rank of signal variables in the X and Y datasets according to the importance measure calculated from the magnitude of the canonical vector loading. Results based on 100 simulations. Lower values of the rank are better. DNF: Did Not Finish.

Dataset	X variables				Y variables			
Initial Value[†]	1	2	3	4	1	2	3	4
PMD (Witten et al. 2009)	36.27 (14)				24.25 (11)			
SELP-I (Safo et al. 2018)	38.13 (12)				26.43 (12)			
spLRMA	35.14 (13)				23.78 (11)			
UST (Waaijenborg et al. 2008)	38.05 (12)	37.63 (12)	DNF	36.4 (12)	26.29 (12)	26.17 (12)	DNF	24.61 (11)
Wilms and Croux (2015)	63 (15)	62.6 (15)	62.2 (17)	55.07 (12)	40.53 (11)	40.72 (11)	45.32 (14)	35.45 (8)
Proximal Gradient - LASSO	58.47 (14)	58.97 (14)	63.11 (17)	11.54 (2)	42.32 (12)	42.4 (12)	45.18 (14)	9.16 (1)
Proximal Gradient - MCP	54.57 (13)	54.5 (13)	DNF	10.58 (0)	38.83 (12)	38.93 (12)	DNF	8.54 (0)
Proximal Gradient - SCAD	52.45 (13)	52.89 (13)	DNF	10.6 (0)	37.09 (12)	37.32 (12)	DNF	8.55 (0)

[†] Initial values: 1–SVD of sample covariance matrix, 2–SVD of sample correlation matrix, 3–ridge CCA, 4–true canonical vector + noise

Table 4.5: Mean (SD) rank of noise variables in the X and Y datasets according to the importance measure calculated from the magnitude of the canonical vector loading. Results based on 100 simulations. Higher values of the rank are better. DNF: Did Not Finish.

Dataset	X variables				Y variables			
Initial Value[†]	1	2	3	4	1	2	3	4
PMD (Witten et al. 2009)	107.64 (2)				86.75 (1)			
SELP-I (Safo et al. 2018)	107.43 (1)				86.51 (1)			
spLRMA	107.76 (1)				86.8 (1)			
UST (Waaijenborg et al. 2008)	107.44 (1)	107.49 (1)	DNF	107.62 (1)	86.52 (1)	86.54 (1)	DNF	86.71 (1)
Wilms and Croux (2015)	104.67 (2)	104.71 (2)	104.76 (2)	105.55 (1)	84.94 (1)	84.92 (1)	84.41 (2)	85.51 (1)
Proximal Gradient - LASSO	105.17 (2)	105.11 (2)	104.65 (2)	110.38 (0)	84.74 (1)	84.73 (1)	84.42 (2)	88.43 (0)
Proximal Gradient - MCP	105.6 (1)	105.61 (1)	DNF	110.49 (0)	85.13 (1)	85.12 (1)	DNF	88.5 (0)
Proximal Gradient - SCAD	105.84 (1)	105.79 (1)	DNF	110.49 (0)	85.32 (1)	85.3 (1)	DNF	88.49 (0)

[†] Initial values: 1–SVD of sample covariance matrix, 2–SVD of sample correlation matrix, 3–ridge CCA, 4–true canonical vector + noise

4.3.2 ADMM Algorithm

For the ADMM algorithm, we consider both a low dimensional setting and a high dimensional setting. For the low dimensional setting, we simulate $p = 20$ variables in the X dataset and $q = 20$ variables in the Y dataset with a sample size of $N = 50$. For the high dimensional setting, we simulate $p = 60$ variables in the X dataset and $q = 60$ variables in the Y dataset with a sample size of $N = 50$. We use 75% sparsity in both cases, resulting in 5 signal variables in the X and Y sets for the low dimensional setting, and 15 signal variables in the X and Y sets for the high dimensional setting. We use the same covariance structures and method of generating the (true population) canonical vectors as in Section 4.3.1.

We compare the ADMM algorithm to the methods of Wilms and Croux (2015) and Witten et al. (2009). Like the proximal gradient algorithm, the ADMM algorithm requires an initial value. However, because of the augmented Lagrangian term added to the objective function, the matrix involved in the quadratic form for the $\boldsymbol{w}$ update is always positive definite, and so the algorithm is robust to the choice of initial value. Thus, we choose a single approach among the four described in Section 4.3.1 for initializing the ADMM algorithm and Wilms and Croux's method. We obtain $(\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)})$ as the left and right singular vectors from the SVD of the sample covariance matrix of X and Y. In addition, we use two different values of the augmented Lagrangian parameter: $\eta = 1$ and $\eta = 10$.

The sequential quadratic programming solver used for the $\boldsymbol{w}$ update in the ADMM algorithm is substantially slower than the matrix multiply operations used for the gradient descent step in the proximal gradient algorithm. As a consequence, it is not practical to fit the method over a 2-dimensional grid of values of the tuning parameters. To save computation time, we search over a 1-dimensional grid by setting $\lambda_x = \lambda_y$. That approach is partially justified by using $p = q$ and the same level of sparsity for the X and Y sets in our simulation settings. To ensure fair comparisons across methods, we also restrict the grid of tuning parameters to $\lambda_x = \lambda_y$ for the methods of Witten et al. (2009) and Wilms and Croux (2015).

Results

Low dimension: $p = 20,\ q = 20,\ N = 50$

Tables 4.6 and 4.7 summarize the mean rank and standard deviation (SD) for the signal variables and noise variables, respectively, for the low dimensional simulation setting. The rank was determined by one of the two importance measures described in Section 3.2.4. The relative performances of the methods depended both on the importance measure and value of the augmented Lagragian parameter η. For the importance measure based on whether the canonical vector loading was nonzero, PMD performed better than the method of Wilms and Croux (2015) and the ADMM algorithm, regardless of the choice of penalty function. Comparing Wilms and Croux's method to the proposed ADMM algorithm, one can see that the relative performance depended on both the value of η and the penalty function. For $\eta = 1$, Wilms and Croux's method performed better than ADMM with LASSO or SCAD penalty, but for $\eta = 10$, the opposite occurred. In contrast, ADMM with MC penalty performed better than Wilms and Croux's method regardless of the value of η. Among the different variants of the ADMM algorithm, the results tended to be better using MC or SCAD than LASSO and better using $\eta = 10$ than $\eta = 1$. Note that for all of the comparisons, the differences in mean rank of the signal variables among the methods was slight.

For the importance measure based on the magnitude of the canonical vector loading, the ADMM algorithm tended to outperform PMD and the method of Wilms and Croux, with a couple of exceptions (*viz.*, ADMM with LASSO and $\eta = 1$). Similar to the other importance measure, the results tended to be better with a larger value of η and with the SCAD or MC penalty than with the LASSO penalty, but the differences in the mean rank of the signal variables among the methods were quite small overall. Note that for all of the methods, the mean rank of the signal variables was better using the importance measure based on the magnitude of the canonical vector loading than the measure based on whether the loading was nonzero.

Table 4.6: Mean (SD) rank of signal variables in the X and Y datasets according to two importance measures. Results based on 100 simulations. Lower values of the rank are better.

Dataset	X variables		Y variables	
Importance Measure†	1	2	1	2
PMD (Witten et al. 2009)	4.94 (2)	4.88 (2)	5.12 (2)	5.08 (2)
Wilms and Croux (2015)	6.17 (2)	5 (2)	5.95 (2)	4.71 (2)
ADMM - LASSO ($\eta = 1$)	6.37 (3)	4.99 (2)	6.29 (3)	4.86 (2)
ADMM - LASSO ($\eta = 10$)	5.93 (2)	4.77 (2)	5.83 (2)	4.6 (2)
ADMM - MCP ($\eta = 1$)	5.96 (3)	4.75 (2)	5.69 (3)	4.52 (2)
ADMM - MCP ($\eta = 10$)	5.58 (2)	4.63 (2)	5.41 (2)	4.27 (1)
ADMM - SCAD ($\eta = 1$)	6.19 (3)	4.78 (2)	5.98 (3)	4.49 (2)
ADMM - SCAD ($\eta = 10$)	5.34 (2)	4.49 (2)	5.22 (2)	4.24 (1)

†1–No. of times a variable was selected,
2–Magnitude of canonical vector loading

Table 4.7: Mean (SD) rank of noise variables in the X and Y datasets according to two importance measures. Results based on 100 simulations. Higher values of the rank are better.

Dataset	X variables		Y variables	
Importance Measure[†]	1	2	1	2
PMD (Witten et al. 2009)	12.35 (1)	12.37 (1)	12.29 (1)	12.31 (1)
Wilms and Croux (2015)	11.94 (1)	12.33 (1)	12.02 (1)	12.43 (1)
ADMM - LASSO ($\eta = 1$)	11.88 (1)	12.34 (1)	11.9 (1)	12.38 (1)
ADMM - LASSO ($\eta = 10$)	12.02 (1)	12.41 (1)	12.06 (1)	12.47 (1)
ADMM - MCP ($\eta = 1$)	12.01 (1)	12.42 (1)	12.1 (1)	12.49 (1)
ADMM - MCP ($\eta = 10$)	12.14 (1)	12.46 (1)	12.2 (1)	12.58 (0)
ADMM - SCAD ($\eta = 1$)	11.94 (1)	12.41 (1)	12.01 (1)	12.5 (1)
ADMM - SCAD ($\eta = 10$)	12.22 (1)	12.5 (1)	12.26 (1)	12.59 (0)

[†] 1–No. of times a variable was selected,
2–Magnitude of canonical vector loading

High dimension: $p = 60,\ q = 60,\ N = 50$

Tables 4.8 and 4.9 summarize the mean rank and standard deviation (SD) for the signal variables and noise variables, respectively, for the high dimensional simulation setting. The rank was determined by one of the two importance measures described in Section 3.2.4. Unlike the low dimensional setting, PMD outperformed both Wilms and Croux's method and the proposed ADMM algorithm, regardless of the penalty function or value of η. Wilms and Croux's method either performed similar to or slightly better than ADMM with $\eta = 1$. However, with $\eta = 10$, the proposed ADMM algorithm performed much better than Wilms and Croux's method.

One noticeable difference in the relative performances of the penalty functions used for the ADMM algorithm was that the MC and SCAD penalties only had an advantage over LASSO for larger values of the augmented Lagrangian parameter η. For the low dimensional setting, the mean rank of the signal variables was better using MC or SCAD penalties than LASSO whether one set the augmented Lagrangian parameter as $\eta = 1$ or $\eta = 10$. For the high dimensional setting, the mean rank of the signal variables was better using MC or SCAD penalties for $\eta = 10$, but the LASSO, MC, and SCAD penalties all performed about the same for $\eta = 1$.

Another difference between the low and high dimensional settings was that the choice of importance measure did not influence the relative performances of the methods substantially. In the low dimensional setting, PMD performed best if using the importance measure based on whether the canonical vector loading was nonzero, but some other methods performed better if using the importance measure based on the magnitude of the canonical vector loading. In the high dimensional setting, PMD performed much better than the other methods for both importance measures, with a 10 point improvement in the mean rank of the signal variables in many cases. Like in the low dimensional setting, the importance measure based on magnitude of the canonical vector loading improved the mean rank of the signal variables slightly for all of the methods.

Table 4.8: Mean (SD) rank of signal variables in the X and Y datasets according to two importance measures. Results based on 100 simulations. Lower values of the rank are better.

Dataset	X variables		Y variables	
Importance Measure[†]	1	2	1	2
PMD (Witten et al. 2009)	19.24 (5)	18.91 (5)	18.79 (5)	18.32 (5)
Wilms and Croux (2015)	28.37 (4)	27.43 (4)	26.34 (4)	24.98 (5)
ADMM - LASSO ($\eta = 1$)	28.68 (4)	27.97 (4)	29.01 (5)	28.4 (5)
ADMM - LASSO ($\eta = 10$)	26.34 (5)	25.57 (5)	26.38 (4)	25.69 (5)
ADMM - MCP ($\eta = 1$)	27.86 (4)	27.03 (4)	27.77 (4)	26.82 (4)
ADMM - MCP ($\eta = 10$)	22.41 (4)	22.52 (4)	22.51 (4)	22.74 (5)
ADMM - SCAD ($\eta = 1$)	28.39 (4)	26.59 (4)	29.27 (4)	27.34 (5)
ADMM - SCAD ($\eta = 10$)	22.14 (4)	22.15 (4)	22.41 (4)	22.59 (4)

[†]1–No. of times a variable was selected,
2–Magnitude of canonical vector loading

Table 4.9: Mean (SD) rank of noise variables in the X and Y datasets according to two importance measures. Results based on 100 simulations. Higher values of the rank are better.

Dataset	X variables		Y variables	
Importance Measure[†]	1	2	1	2
PMD (Witten et al. 2009)	34.25 (2)	34.36 (2)	34.4 (2)	34.56 (2)
Wilms and Croux (2015)	31.21 (1)	31.52 (1)	31.89 (1)	32.34 (2)
ADMM - LASSO ($\eta = 1$)	31.11 (1)	31.34 (1)	31 (2)	31.2 (2)
ADMM - LASSO ($\eta = 10$)	31.89 (2)	32.14 (2)	31.87 (1)	32.1 (2)
ADMM - MCP ($\eta = 1$)	31.38 (1)	31.66 (1)	31.41 (1)	31.73 (1)
ADMM - MCP ($\eta = 10$)	33.2 (1)	33.16 (1)	33.16 (1)	33.09 (2)
ADMM - SCAD ($\eta = 1$)	31.2 (1)	31.8 (1)	30.91 (1)	31.55 (2)
ADMM - SCAD ($\eta = 10$)	33.28 (1)	33.28 (1)	33.2 (1)	33.14 (1)

[†] 1–No. of times a variable was selected,
2–Magnitude of canonical vector loading

4.4 Limitations and Future Directions

Compared to other methods, neither of the proposed algorithms described in Section 4.2.2 performed well with respect to all aspects of the optimization procedure. The weaknesses of the proximal gradient algorithm tended to be the strengths of the ADMM algorithm, and vice versa. We discuss some of the major weaknesses and directions for potential improvement.

Sensitivity to Initial Value

The update for $\boldsymbol{w}$ in the ADMM algorithm is solved by sequential quadratic programming, and the solver available in the `nloptr` package in R requres an initial guess. As shown in problem (4.4), the matrix involved in the quadratic form in $\boldsymbol{w}$ is always positive definite, so the `nloptr` solver is robust to the choice of initial value. As a consequence, the ADMM algorithm as a whole is robust to the choice of initial value. The ADMM algorithm also requires an initial value for $\boldsymbol{z}$, but it is typically initiated as the zero vector or warm-started.

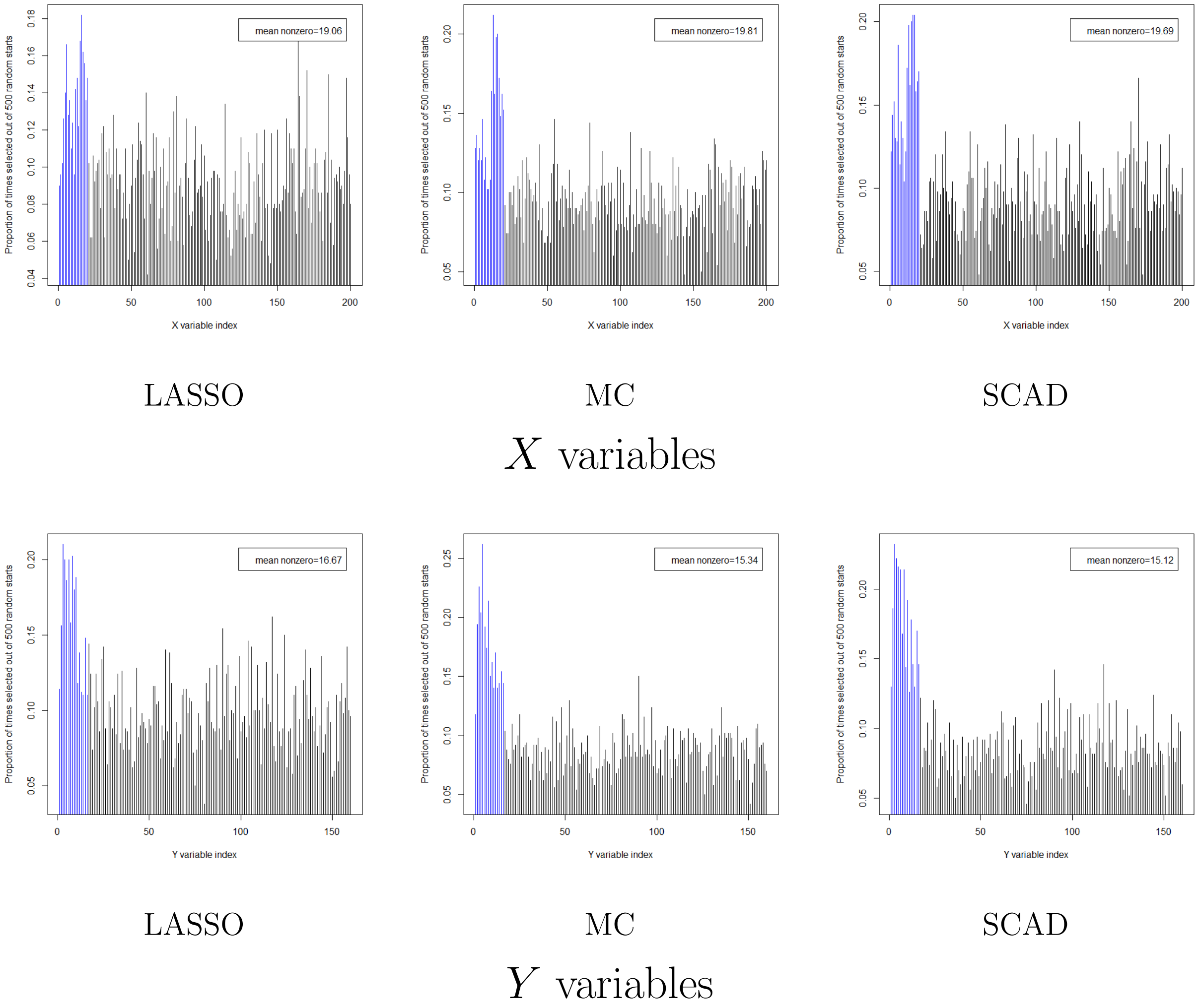
The positive-definiteness of the matrix involved in the quadratic form in $\boldsymbol{w}$ is a consequence of augmented Lagrangian term added to the objective function for the update. It is also a general property of the proximal operator: if a function f is convex, then the proximal operator of f is strongly convex (Parikh and Boyd, 2014). [Though note that the $\boldsymbol{w}$ update itself is not actually convex because of the additional quadratic equality constraints.] In contrast to the ADMM algorithm, the proximal gradient algorithm does not make use of an augmented Lagrangian, so the quadratic form in $\boldsymbol{w}$ involves a positive semi-definite matrix (unless $p+q < n$). Thus, the intermediate update in the proximal gradient algorithm, which is carried out by a gradient descent step, does not arrive at a unique solution. As a result, the proximal gradient algorithm tends to be sensitive to the initial value.

Although the proximal gradient algorithm is sensitive to the initial value, it has the potential to outperform other methods if one can supply a “good” initial value. In the simulation experiment in Section 4.3.1, we used the true canonical vector + noise as the initial value, reasoning that the true canonical vector should be closer to the global minimum than values obtained through other approaches. Since one cannot use this approach in practice, any improvement to the proximal gradient algorithm via the initialization strategy would have to come through another avenue.

A common strategy for problems with many local optima is to try many different randomly selected starting values. One can then choose the value that resulted in the best value of the objective function. Since our goal is primarily accurate variable selection (or ranking), a variation on that strategy may be more appropriate. One could fix the values of the tuning parameters to achieve some desired level of sparsity. One could then obtain estimates for many different randomly selected starting values and count the proportion of times each variable is selected (i.e., has a nonzero canonical vector loading). Those variables selected most often might represent the true signal variables, while those selected least often might represent the true noise variables. One could set a threshold for the proportion of times selected and use variables that fall above the threshold for constructing a new initial value.

For example, one could perform CCA with the reduced set of variables falling above the threshold, use the loadings from the reduced CCA to initialize those variables, and initialize all other variables as zero. The new initial value could then be used to run the algorithm for the entire 2D grid of tuning parameters.

We illustrate the first part of the approach described in the preceding paragraph in Figure 4.1. Using the same covariance structures as in Section 4.3.1, a single dataset was generated and the proximal gradient algorithm was applied with the LASSO, MC, and SCAD penalties. The values of the sparsity tuning parameters were set to achieve approximately the correct level of sparsity (20 signal variables in the X set and 16 signal variables in the Y set); for reference, the mean number of nonzero canonical vector loadings over the 500 random starts is shown in each plot. The x-axis of each plot denotes the variable index and the y-axis shows the proportion of times a variable was selected. The blue vertical bars correspond to the signal variables and black bars correspond to the noise variables. On average, the signal variables tend to be selected more often than the noise variables, with MC and SCAD penalties differentiating the signal group from the noise group somewhat more than the LASSO penalty.



LASSO MC SCAD

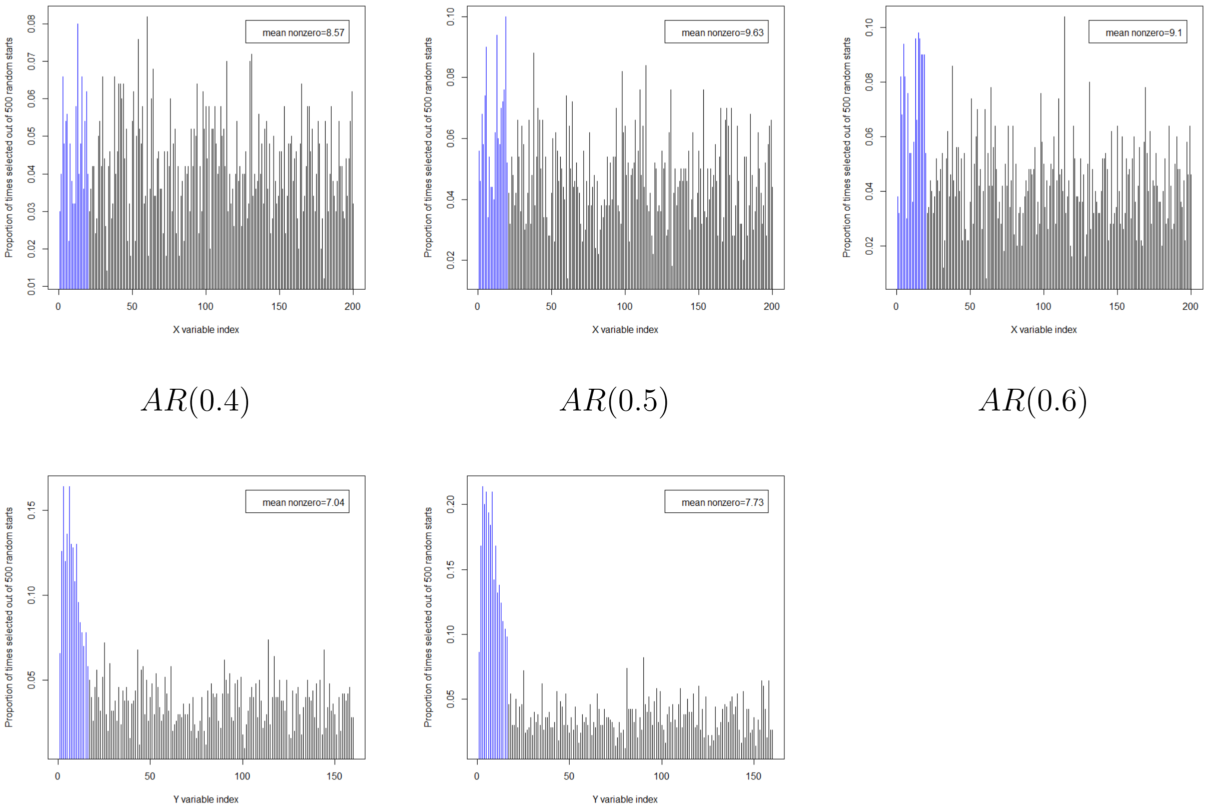
X variables

LASSO MC SCAD

Y variables

Figure 4.1: Proportion of times a variable was selected out of 500 random starts.

Based on the plots in Figure 4.1, it appears that one can identify the signal variables by trying many different random starting values and setting a threshold for the proportion of times a variable was selected. However, the success of this approach strongly depends on the within-set covariance structures Σ_{XX} and Σ_{YY}. To illustrate, we simulate datasets with the same block-diagonal AR covariance structure as before, but we vary the autocorrelation parameter from 0.4 to 0.8. In Figure 4.2, we show the proportion of times each X variable was selected over 500 random starts using the SCAD penalty (the salient features are similar for the Y set or using the LASSO or MC penalty). First, comparing the panel corresponding to $AR(0.7)$ to the appropriate panel in Figure 4.1, we can see that setting the tuning parameter to achieve higher sparsity than the true level of sparsity actually does a better job of differentiating the signal and noise variables. Second, we can see how the success of the approach changes as the within-set covariance structure becomes stronger or weaker. When the signal variables have high within-set correlation, as in the $AR(0.8)$ panel, the signal variables are selected far more often than the noise variables. However, when the signal variables have low within-set correlation, as in the $AR(0.4)$ panel, the signal and noise variables are selected about the same proportion of the time. Thus, overcoming the sensitivity of the proximal gradient method to the initial value by trying many random starts only appears to be feasible when one can expect the signal variables to be highly correlated among themselves (and uncorrelated with the noise variables). For real data, one cannot generally expect this, so an approach based on trying many random starting values cannot be relied upon in practice.



$AR(0.4)$ $AR(0.5)$ $AR(0.6)$

$AR(0.7)$ $AR(0.8)$

Figure 4.2: Proportion of times an X variable was selected out of 500 random starts. Note: All results use SCAD penalty.

Sensitivity to Algorithmic Parameters

In addition to the sparsity tuning parameters, both the proximal gradient method and ADMM algorithm have algorithmic parameters that one must choose. For both algorithms, one must define a stopping a criterion based on an absolute and/or relative tolerance. One can choose the stopping criterion and tolerance based on standard suggestions, making adjustments if the situation warrants it. For example, for the proximal gradient algorithm, we defined the error in terms of the objective value: $\frac{|f(\boldsymbol{w}^k)-f(\boldsymbol{w}^{k-1})|}{1+|f(\boldsymbol{w}^{k-1})|}$, where f denotes the function specified in equation (4.3). We terminated the algorithm when the error fell below a tolerance of 1E−3. The quality of the solutions did not change substantially with smaller values for the tolerance or other stopping criteria, such as $||\boldsymbol{w}^k - \boldsymbol{w}^{k-1}||_2$.

Besides the stopping criterion and tolerance, the only additional algorithmic parameter for proximal gradient algorithm is the step size in the gradient descent step (γ^k in Algorithm 6). We determine the step size using a standard line search, so the proposed proximal gradient method is robust to the initial choice of step size (although there is some computational cost involved if the initial choice is not near an appropriate value).

For the ADMM algorithm, one must specify the value of the augmented Lagrangian parameter (η in Algorithm 7). In the context of the sparse CCA problem, the augmented Lagrangian parameter η controls the trade-off between minimizing the quadratic form in $\boldsymbol{w}$ and staying close to $\boldsymbol{z}$ (where $\boldsymbol{z}$ can be interpreted as a sparse version of $\boldsymbol{w}$). For many applications, ADMM is not sensitive to the choice of η, and so it can be set as $\eta = 1$. However, as shown in Tables 4.6–4.9, the quality of the solution depends on the value of η, especially for the high dimensional setting $p, q > n$. To understand why the quality of the solution changes with η, recall equation (4.4) and consider the case in which we initialize $\boldsymbol{z}$ as the zero vector. The matrix $A^TA + \eta I$ can be expressed as

$$A^TA + \eta I = \begin{pmatrix} X^TX + \eta I_p & -X^TY \\ -Y^TX & Y^TY + \eta I_q \end{pmatrix}.$$

Rescaling by a factor of $\frac{1}{n}$ (which does not change the solution for the $\boldsymbol{w}$ update), we can write

$$\frac{1}{n}A^TA + \eta^* I = \begin{pmatrix} \widehat{\Sigma}_{XX} + \eta^* I_p & -\widehat{\Sigma}_{XY} \\ -\widehat{\Sigma}_{YX} & \widehat{\Sigma}_{YY} + \eta^* I_q \end{pmatrix}.$$

The matrix above is the same matrix that would be involved in solving the problem of ridge CCA. [Note that the solutions for the first $\boldsymbol{w}$ update and ridge CCA are *not* identical because the constraints for ridge CCA are $\boldsymbol{u}^T(\widehat{\Sigma}_{XX} + \eta I_p)\boldsymbol{u} = \boldsymbol{v}^T(\widehat{\Sigma}_{YY} + \eta I_q)\boldsymbol{v} = 1$, whereas the constraints for the $\boldsymbol{w}$ update are $\boldsymbol{u}^T\widehat{\Sigma}_{XX}\boldsymbol{u} = \boldsymbol{v}^T\widehat{\Sigma}_{YY}\boldsymbol{v} = 1$.] Thus, for the sparse CCA problem, we can see that the augmented Lagrangian parameter has an alternative interpretation beyond its typical interpretation in ADMM; namely, η plays a very similar role as the tuning parameters in ridge CCA.

Because the first $\boldsymbol{w}$ update in the ADMM algorithm is so similar to ridge CCA, we might expect the ADMM algorithm to produce good solutions when ridge CCA produces a good solution and poor solutions when ridge CCA produces a poor solution. Since the solutions from ridge CCA are not sparse, a good solution would have large magnitude loadings for the signal variables and small magnitude loadings for the noise variables. For the high dimensional simulation settings examined in this work, larger values of the ridge CCA tuning parameters tended to produce better solutions than did smaller values. In fact, the choice of $\eta = 10$ for one setting of the ADMM algorithm was not arbitrary, but rather a consequence of this observation. When we realized the results using $\eta = 1$ were not as good as expected, we used the connection between the ADMM algorithm and ridge CCA to guide our selection of a better value of η. We plotted the loadings from ridge CCA as we varied the values of tuning parameters and chose $\eta = 10$ because it produced better solutions on average.

Although the connection between the ADMM algorithm and ridge CCA is helpful for understanding why the quality of the solution depends on the augmented Lagrangian parameter, the connection also suggests there is little room to improve the ADMM algorithm. That is, when ridge CCA does not work well, neither will the ADMM algorithm. In contrast, the proximal gradient algorithm does not use an augmented Lagrangian, so it does not share the connection to ridge CCA and its solution can be improved substantially with modifications such as using a better initial value.

Computation Time

The proximal gradient algorithm only requires matrix-vector multiply operations and element-wise vector thresholding. In addition, only a single gradient descent step is taken at each iteration (assuming the step size is chosen suitably). Consequently, the proximal gradient algorithm is relatively fast.

For the ADMM algorithm, the $\boldsymbol{z}$ update requires the same element-wise vector thresholding operations as for the proximal gradient algorithm. However, the $\boldsymbol{w}$ update in the ADMM algorithm must itself be solved iteratively. As a consequence, the ADMM algorithm is much slower than the proximal gradient algorithm. In addition, the sequential quadratic programming solver does not scale well with the dimension $(p+q)$, making the proposed ADMM algorithm impractical for high dimensional problems. Since the bottleneck is due the use of sequential quadratic programming for the $\boldsymbol{w}$ update (as opposed to the total number of ADMM iterations), the computation time could be improved by replacing sequential quadratic programming with another algorithm that is both capable of solving QCQP problems and scales well with the dimension, such as those based on interior-point methods (Byrd et al., 1999, 2000). Although it may be possible to improve the computation time of the ADMM algorithm, its sensitivity to the augmented Lagrangian parameter appears to be an inherent issue and changing the method for solving the $\boldsymbol{w}$ update will not improve the variable selection accuracy. Thus, rather than searching for a faster QCQP solver to incorporate into Algorithm 7, it may be more fruitful to develop an alternative to ADMM to solve problem (4.3).

Variable Ranking Accuracy

For the high dimensional setting, both of the proposed algorithms achieved better results than Wilms and Croux's (2015) alternating regression approach (though for the ADMM algorithm, the improvement was contingent on an appropriate value of augmented Lagrangian parameter η). For the LASSO penalty, the proximal gradient algorithm, ADMM algorithm, and alternating regression are all solving the same problem, so the improvement in the results can be attributed to the algorithms themselves. One possible explanation is that the proximal gradient and ADMM algorithms update the estimates of the canonical vectors simultaneously, and simultaneous estimation has an advantage over alternating minimization. However, there may be other aspects of the algorithms that explain their better performance. Both the proximal gradient algorithm and ADMM algorithm benefited further from the use of MC and SCAD penalties, which is consistent with other applications (e.g., high dimensional regression).

Although the proposed algorithms improved on the alternating regression approach, none of the methods based on the regression formulation of CCA achieved as high variable ranking accuracy as methods of Waaijenborg et al. (2008), Witten et al. (2009), or Safo et al. (2018). All of these methods avoid the need to estimate the within-set covariance matrices by assuming they are identity – i.e., they assume $\Sigma_{XX} = I_p$ and $\Sigma_{YY} = I_q$. [Note: Although Waaijenborg et al. (2008) begin with a regression formulation using the elastic net penalty, they take the limit of the ridge parameters $\lambda_x^{\text{ridge}}, \lambda_y^{\text{ridge}} \to \infty$, which effectively assumes the within-set covariance matrices are identity.] Given that all of the simulation settings involved some within-set correlation among the signal variables, the better performance of the methods assuming identity covariance matrices is somewhat counterintuitive. That is, the methods making no assumptions performed worse than methods making incorrect assumptions. Because covariance matrices are very difficult to estimate in the high dimensional setting, it may be that the loss of efficiency outweighs the benefit of relaxing the assumptions regarding the covariance structure. The assumption that the within-set covariance are identity can be viewed as form of regularization (separate from sparse regularization), and the benefits of regularization resulting from the bias-variance trade-off are well known in the high dimensional setting [e.g., the discussion in Tibshirani (1996)].

Future Directions

Barring a better approach for initializing the proximal gradient algorithm, it is unlikely that either of the proposed algorithms can be improved to such an extent as to perform competitively against the best performing methods for the high dimensional setting. However, the proposed algorithms, and more generally, the formulation of the sparse CCA problem as in equation (4.3), are not without their merits. In the low dimensional setting, the ADMM algorithm outperformed the method of Witten et al. (2009) in some instances, indicating that the proposed methodology may find applications in problems for which the number of variables is moderate compared to the sample size.

One of the main advantages of formulating CCA as minimization of a quadratic form are the potential extensions discussed in Sections 4.2.3 and 4.2.4. In modern research, it is becoming increasingly common to collect multimodal data and data from heterogeneous groups of subjects. For multimodal data, it is natural to treat each data type as a separate dataset, and in some situations, it may be sensible to perform an integrated analysis of all of the datasets simultaneously. The extension to the proposed methodology discussed in Section 4.2.3 not only provides a way to solve such a problem, but also to solve the problem under the additional assumption of sparsity. For grouped data, the proposed extension again relies on the similarity of equation (4.3) to linear regression to leverage one of the many extensions of the LASSO that have been proposed for regression. In particular, a fused LASSO-type penalty Tibshirani et al. (2005) provides a way to account for associations in the data that are shared across groups, while allowing for group-specific components in the association. Although the extensions of the proposed methodology provide some advantages over other methods, it is also important to consider their limitations. The major limitations encountered for sparse CCA are likely to be encountered, and possibly exacerbated, in the extensions. For example, with two datasets, the ADMM algorithm became less effective as $p + q$ approached n. For three datasets with p_1, p_2, and p_3 variables, one can expect the methodology to become less effective as $p_1 + p_2 + p_3$ approaches n. More generally, the performance can be expected to decline as the number of datasets increases (assuming each additional dataset has about the same number of variables as the others). Thus, the potential extensions are likely to be the most useful for low dimensional problems.

Chapter 5

Conclusions and Future Directions

Two important general conclusions from the work in Chapters 2 – 4 are: 1) shrinking the singular values of a matrix can be an effective alternative to a fixed-rank approximation for statistical methods relying on or incorporating low rank matrix approximations and 2) incorporating sparsity into low rank matrix approximation problems can be challenging.

Singular value shrinkage vs. fixed-rank approximations

In Chapter 2, we demonstrated that an orthogonal tensor regression model that penalizes the singular values (and hence shrinks some singular values to zero) can provide a better low rank approximation of the parameter tensor than the corresponding fixed rank version of the model. In the case of our real data analysis, "better" translated to more accurate prediction for a test dataset and an estimate that was easier to interpret when visualized. In Chapter 3, we demonstrated that variable selection accuracy in sparse CCA can be improved by relaxing the problem to focus solely on variable selection rather than simultaneously maximizing the canonical correlation. In that case, we redefined the problem so that we could incorporate shrinkage, but it was unclear to what extent the improved variable selection accuracy was a result of shrinkage vs. relaxing the objectives of the analysis.

Some natural follow-up questions are: For low rank matrix approximation problems, why do methods based on shrinkage sometimes outperform methods based on fixed-rank approximations? When can we expect methods based on shrinkage to perform better than methods based on fixed-rank approximations? How can we incorporate shrinkage into other statistical problems that rely on low rank matrix approximations? Can we extend methods traditionally based on matrices to higher-order tensors?

Some intuition into why methods based on shrinkage can outperform methods based on fixed-rank approximations has already been discussed in Chapters 2 and 3. In the context of regression with matrix-valued parameters, Zhou and Li (2014) showed that the effective number of parameters in the version of the problem based on shrinkage is dominated by the naive count of the number of parameters, which means that the effective number of parameters in the fixed-rank version of the problem is always larger (as long as the estimates have the same rank). When the sample size is limited, there can be an advantage to reducing the effective number of parameters beyond what the fixed-rank version of the problem permits.

In other contexts, such as CCA, it is more difficult to evaluate the relative merits of shrinkage vs. fixed-rank approximations in terms of the effective number of parameters. We may think of the canonical vectors themselves as the parameters, in which case the number of parameters is the number of elements of the vectors (i.e., the total number of variables if we are only concerned with the largest canonical correlation). The regression formulation of CCA discussed in Chapter 4 also provides some support for the notion of canonical vectors as parameters. However, it is unclear how one might define the effective number of parameters such that the definition generalizes to methods based on shrinkage, where determining the effective number of parameters may not simply be a matter of counting the number of elements of the canonical vectors. Zhou and Li (2014) used the definition of effective degrees of freedom studied by Ye (1998) and Efron (2004) to derive their estimate of the effective number of parameters in regularized matrix regression. For a response $Y := (Y_1, \ldots, Y_n)^T$ such that $Y \sim N(\boldsymbol{\mu}, \sigma^2 I)$ and corresponding estimate $\hat{\boldsymbol{\mu}} := (\hat{Y}_1, \ldots, \hat{Y}_n)^T$, that definition of effective degrees of freedom is

$$df^{eff} := \frac{1}{\sigma^2} \sum_{i=1}^{n} Cov(\hat{Y}_i, Y_i).$$

Besides that CCA involves a more general covariance structure Σ, the dual nature of CCA presents a challenge for extending that concept of df^{eff}. In regression, $Y_{n\times 1}$ is the only random variable, and we predict it using information in some fixed covariates $X_{n\times p}$. In CCA, $Y_{n\times q}$ and $X_{n\times p}$ are both random, and each plays the role of both the (multivariate) response and the predictor. We also have two sets of estimates: $\hat{x}_i := \boldsymbol{x}_i^T \hat{\boldsymbol{u}}$ and $\hat{y}_i := \boldsymbol{y}_i^T \hat{\boldsymbol{v}}$. Considering the regression formulation of CCA,

$$\min_{\boldsymbol{u},\boldsymbol{v}} \ (X\boldsymbol{u} - Y\boldsymbol{v})^T (X\boldsymbol{u} - Y\boldsymbol{v}),$$

it would appear that $\hat{y}_i$ is the estimate of $\hat{x}_i$ and vice versa. Should we then define the effective degrees of freedom in terms of $Cov(\hat{y}_i, \hat{x}_i)$? Or is the entire framework ill-suited for CCA? Developing ideas such as those is a possible direction of future study.

Previous work has investigated the effectiveness of different techniques for the general problem of recovering a low rank signal matrix X from a noisy realization $Y = X + E$, where E represents random noise. That work may yield some insight into the question of when methods based on shrinkage to outperform those based on fixed-rank approximations. Gavish and Donoho (2014) and Gavish and Donoho (2017) studied the asymptotic mean squared error (AMSE) of various methods of recovering X through a low rank approximation of Y. Among the methods compared were singular value hard-thresholding (SVHT), which is similar to methods based on fixed-rank approximations, and singular value soft-thresholding (SVST), which is similar to methods based on shrinkage. In the settings they studied, the AMSE of optimally-tuned SVST was larger than the AMSE of optimally-tuned SVHT for high signal-to-noise ratio (SNR), but lower for low SNR. That is, shrinking the singular values may result in better performance than a fixed-rank approximation when the signal is weak. Some of the CCA simulation results in Section 3.3 are in (loose) agreement with their observations. For the simulation settings that used high within-set correlation structure for the signal variables [e.g., $AR(0.9)$], the signal region of $\widehat{\Sigma}_{XY}$ was easily detectable, and in those settings the proposed method based on shrinkage performed similarly (with respect to variable selection accuracy) to two sparse CCA methods that find a rank-1 approximation of $\widehat{\Sigma}_{XY}$. For the simulation settings that used lower within-set correlation structure, the signal region of $\widehat{\Sigma}_{XY}$ was less apparent, and the advantage of proposed method over the sparse CCA methods was more obvious. However, we caution that the reasons for the apparent agreement between our results and those of Gavish and Donoho (2014) and Gavish and Donoho (2017) are purely speculative. We only considered a method's variable selection accuracy when measuring its performance, not its AMSE or even the observed MSE in simulation (i.e., we may be trying to compare apples to oranges). In addition, all of the methods we compared in Section 3.3 incorporate sparsity into the low rank approximation problem, so it is difficult to isolate the effect of shrinkage vs. fixed-rank approximation from the effect of using different approaches to achieve sparsity when analyzing the methods' relative performances.

Determining whether a method's advantage over another depends on the SNR, and whether the advantage extends across multiple measures of performance, is another possible future direction of study. Certainly in sparse CCA, it would be interesting to know whether the approach used to obtain the low rank approximation affected the variable selection accuracy. However, because the approach for incorporating sparsity most directly affects the variable selection accuracy, it would be necessary to compare methods that achieve sparsity in the same way. That may be challenging for CCA because methods based on the fixed-rank approximation penalize the elements of the canonical vectors. It is difficult to imagine how to write the problem such that one can both penalize the elements of canonical vectors and incorporate shrinkage; the fact that there are canonical vectors to penalize implies a fixed-rank formulation of CCA. The problem would perhaps be easier to study in the context of regression with matrix- or tensor-valued parameters. In our simulation experiment in Section 2.3, we did not vary the SNR. However, it would not be difficult to design such an experiment, though it would be easier to study in the context of matrices than tensors because of computation time. One possible confounding factor is the algorithm used to fit the model. The fixed-rank matrix regression model proposed by Zhou et al. (2013) is non-convex, and our simulation and real data analysis results suggest their algorithm does a poor job of finding the global optimum. In the case of matrices, the fixed-rank version of our orthogonal tensor regression model is simply a reparametrization of Zhou et al.'s model, and our projected gradient descent algorithm seems to find relatively good estimates of the parameter. Then comparing the fixed-rank and shrinkage-based versions of our orthogonal tensor regression model may be best way to address how the SNR affects the relative performances of two approaches.

Some other statistical methods that can be viewed as low rank matrix approximations based on a fixed-rank decomposition include PCA and Fisher's LDA. As discussed in Chapter 1, the solutions to PCA and LDA can be obtained from the spectral decompositions of Σ and $\Sigma_w^{-1/2}\Sigma_b\Sigma_w^{-1/2}$, respectively. As we did for CCA in Chapter 3, we might develop alternatives to PCA and LDA that achieve a low rank approximation of the relevant matrix by shrinking the eigenvalues rather than finding a fixed-rank approximation. In addition, we could incorporate sparsity that results in variable selection. We might also ask whether we can relax one objective of the analysis so that we can improve another, such as we did for CCA by forgoing the maximization of the canonical correlation so that we could improve the variable selection accuracy. In that case, it would not be technically correct to call the methods sparse PCA or sparse LDA, just as it is not correct to call the method proposed in Chapter 3 sparse CCA.

One challenge in developing sparse versions of PCA or LDA based on eigenvalue shrinkage is how to actually incorporate sparsity into the problem. The matrices involved in PCA and LDA are symmetric, so if we were to incorporate sparsity by penalizing the ℓ_2 norms of the rows or columns, as we did in Chapter 3, we would also need to develop a new algorithm that respects the symmetry of the matrices. We could consider alternative approaches for incorporating sparsity, but it is unclear how to do so without penalizing the elements of the eigenvectors (which would imply a fixed-rank form of the problem). We could also consider alternative methods of obtaining the solutions of the problems. The solution to PCA can be found from the SVD of the data matrix $X_{n\times p}$, and the solution to LDA can be found from the eigendecomposition of the (non-symmetric) matrix $\Sigma_w^{-1}\Sigma_b$. At first glance, it would seem we could achieve variable selection in PCA by penalizing the ℓ_2 norms of the columns of X. However, in practice, we would often want to standardize the variables because variables are often measured on different scales. In that case, all of the column norms of X are equal to one, so it wouldn't make sense to penalize them. It is neither clear whether penalizing the column norms of $\Sigma_w^{-1}\Sigma_b$ would result in variable selection, nor whether such an approach would have unintended consequences, such as it does for PCA. In either case, the matter requires additional thought.

Sparsity in low rank matrix approximation problems

The methods proposed in Chapters 3 and 4 incorporated sparsity into low rank matrix approximation problems as a way to achieve variable selection. The sparse CCA methods proposed in Chapter 4 utilized the popular approach of penalizing the objective function with a sparsity-inducing penalty. In the case of CCA, the objective function was based on maximizing the canonical correlation, and the penalty was applied to the canonical vectors themselves. Such a problem is difficult to solve because of the non-convexity implied by the quadratic equality constraints. Although the standard formulation of CCA admits an analytical solution, sparse CCA requires a numerical solution. Neither of the two algorithms proposed to solve the sparse CCA problem in Chapter 4 performed well. The algorithm based on the proximal gradient method was highly sensitive to the initial value, and the algorithm based on ADMM did not scale well to high dimensional problems. The proximal gradient algorithm implemented an ad hoc approach for handling the quadratic equality constraints, while the ADMM algorithm included a sub-routine explicitly designed to handle quadratic equality constraints, perhaps accounting for the proximal gradient algorithm's sensitivity to the initial value and the better performance of the ADMM algorithm in low dimensional problems.

Gaynanova et al. (2017) pointed out that the penalized and constrained versions of non-convex problems such as PCA and CCA are not equivalent, adding to and possibly explaining some of the difficulties encountered in Chapter 4. The biggest consequence of that lack of equivalence in practice is that the penalized version of the problem may not be able to achieve every level of sparsity. Gaynanova et al. (2017) observed a threshold beyond which sparser solutions to the problem could not be obtained. The threshold increased as the number of variables p increased relative to the sample size n. For example, with $p = 2000$ variables and a sample of size $n = 50$, it may not be possible to select fewer than 500 variables when the problem is formulated as a penalty on the ℓ_1 norm [e.g., Fig. 1 in Gaynanova et al. (2017)]. Given the frequency of high dimension, low sample size problems in practice, and the need to obtain interpretable results based on a small number of variables, the inability to obtain very sparse solutions is problematic. Although we did not directly investigate the sparsity levels of the solutions produced by the algorithms proposed in Chapter 4, an inability to explore all levels of sparsity could explain the poor performance of the algorithms for high dimensional problems.

The sparse, low rank matrix approximation problem proposed in Chapter 3 circumvented the issues described by Gaynanova et al. (2017) by reformulating the objective function as a convex problem. The quadratic equality constraints in PCA, CCA, and similar problems cause the non-convexity. When a low rank matrix approximation is formulated as a fixed-rank approximation, the ℓ_2 norms of the eigenvectors or singular vectors must be constrained to equal one, which is a quadratic equality constraint. In contrast, the problem in Chapter 3 does not involve any constraints. The ℓ_1 norm of the singular values is a convex function, so we can obtain a low rank approximation without introducing non-convex constraints. Reformulating the problem as a convex problem as in Chapter 3 is not without drawbacks. One issue was that the objective function no longer maximized the canonical correlation, so it cannot accurately be called CCA. However, the more important issue was again sparsity. Sparsity was incorporated as a penalty on the ℓ_2 norms of the rows or columns of the low rank matrix. Such a penalty causes entire rows or columns to shrink to zero. Then, both the penalty on the singular values and penalty on the row/columns norms affect the rank of the low rank matrix that is the solution to the problem. Because of the interplay between the two penalties with respect to their effect on the rank of the solution, the problem proposed in Chapter 3 was challenging in the sense that it was difficult to choose the tuning parameters such that every combination of rank and sparsity level of the solution was explored. So rather than avoiding the challenge posed by incorporating sparsity into a non-convex problem such as CCA, we simply traded one kind of challenge for another.

A direction of future research is to explore whether the methods proposed in Chapter 4 are susceptible to the issues Gaynanova et al. (2017) studied. The methods described in Chapter 4 reformulated the CCA objective function as a quadratic program, then added a penalty term involving the canonical vectors and one of the LASSO, SCAD, or MC penalties. The problem could also be formulated with a bound constraint on the ℓ_1 norm of the canonical vectors (i.e., the constrained version of the LASSO) rather than a penalty on the ℓ_1 norm. That would allow one to study whether different levels of sparsity are achievable in sparse CCA for the constrained vs. penalized versions of the LASSO, just as Gaynanova et al. (2017) studied. In addition, it is not clear whether the same phenomenon occurs for concave penalties such as SCAD and MC. Although it is not obvious what the constrained version of

SCAD or MC penalty would be, one could at least compare the levels of sparsity achievable with SCAD or MC penalty vs. the constrained form of the LASSO. It may also be possible to extend the approach for sparse CCA developed in Chapter 4 to PCA and LDA. In which case, one could study the relationship between sparsity level and constrained vs. penalized versions of the problem in a wider context.

Although it would take little additional effort to reformulate the sparse CCA methods of Chapter 4 as a constraint on the ℓ_1 norm of the canonical vectors, the main challenge that would need to be addressed before pursuing the research directions described in the previous paragraph is developing an algorithm that scales well with the dimension. The phenomenon Gaynanova et al. (2017) described is most prominent when $p \gg n$. The sequential quadratic programming sub-routine that is part of the ADMM algorithm for sparse CCA is the bottleneck with respect to computation time, but is also the piece of the algorithm that is responsible for its robustness to the initial value. To make the overall algorithm for sparse CCA scale better, sequential quadratic programming would need to be replaced with a faster algorithm that is still capable of handling quadratic equality constraints. Section 4.4 discussed some possibilities involving interior-point algorithms.

Bibliography

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122.

Byrd, R. H., Gilbert, J. C., and Nocedal, J. (2000). A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185.

Byrd, R. H., Hribar, M. E., and Nocedal, J. (1999). An interior point algorithm for large-scale nonlinear programming. *SIAM J. on Optimization*, 9(4):877–900.

Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011). Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37.

Chalise, P. and Fridley, B. L. (2012). Comparison of penalty functions for sparse canonical correlation analysis. *Computational statistics and data analysis*, 56(2):245–254.

Chandrasekaran, V., Parrilo, P. A., and Willsky, A. S. (2012). Latent variable graphical model selection via convex optimization. *Ann. Statist.*, 40(4):1935–1967.

Chen, H., Raskutti, G., and Yuan, M. (2019). Non-convex projected gradient descent for generalized low-rank tensor regression. *Journal of Machine Learning Research*, 20(5):1–37.

Chen, J. and Saad, Y. (2009). On the tensor SVD and the optimal low rank orthogonal approximation of tensors. *SIAM Journal on Matrix Analysis and Applications*, 30(4):1709–1734.

Chen, M., Gao, C., Ren, Z. H., and Zhou, H. H. (2013). Sparse cca via precision adjusted iterative thresholding. http://arxiv.org/abs/1311.6186.

Chun, H. and Keleş, S. (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(1):3–25.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37 – 46.

De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000a). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278.

De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000b). On the best rank-1 and rank-(R1, R2,...,RN) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342.

Duberstein, K. J., Platt, S. R., Holmes, S. P., Dove, C. R., Howerth, E. W., Kent, M., Stice, S. L., Hill, W. D., Hess, D. C., and West, F. D. (2014). Gait analysis in a pre- and post-ischemic stroke biomedical pig model. *Physiology and Behavior*, 125:8 – 16.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Efron, B. (2004). The estimation of prediction error: Covariance penalties and cross-validation. *Journal of the American Statistical Association*, 99(467):619–642.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911.

Fan, J. and Lv, J. (2010). A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20:101–148.

Gavish, M. and Donoho, D. L. (2014). The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053.

Gavish, M. and Donoho, D. L. (2017). Optimal shrinkage of singular values. *IEEE Transactions on Information Theory*, 63(4):2137–2152.

Gaynanova, I., Booth, J. G., and Wells, M. T. (2017). Penalized versus constrained generalized eigenvalue problems. *Journal of Computational and Graphical Statistics*, 26(2):379–387.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA.

Guhaniyogi, R., Qamar, S., and Dunson, D. B. (2017). Bayesian tensor regression. *Journal of Machine Learning Research*, 18(79):1–31.

Guo, W., Kotsia, I., and Patras, I. (2012). Tensor learning for regression. *IEEE Transactions on Image Processing*, 21(2):816–827.

Hanson, S. J., Matsuka, T., and Haxby, J. V. (2004). Combinatorial codes in ventral temporal lobe for object recognition: Haxby (2001) revisited: is there a “face” area? *NeuroImage*, 23(1):156 – 166.

Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., and Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293(5539):2425–2430.

Hitchcock, F. L. (1927). The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189.

Hoff, P. D. (2015). Multilinear tensor regression for longitudinal relational data. *Ann. Appl. Stat.*, 9(3):1169–1193.

Holm, K., Hegardt, C., Staaf, J., Vallon-Christersson, J., Jönsson, G., Olsson, H., Borg, Å., and Ringnér, M. (2010). Molecular subtypes of breast cancer are associated with characteristic dna methylation patterns. *Breast Cancer Research*, 12(3):R36.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28(3-4):321–377.

Hung, H. and Wang, C.-C. (2012). Matrix variate logistic regression model with application to EEG data. *Biostatistics*, 14(1):189–202.

Iaci, R., Sriram, T., and Yin, X. (2010). Multivariate association and dimension reduction: A generalization of canonical correlation analysis. *Biometrics*, 66(4):1107–1118.

Izenman, A. J. (1975). Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248 – 264.

Jain, P. and Kar, P. (2017). Non-convex optimization for machine learning. *Foundations and Trends in Machine Learning*, 10(3-4):142–363.

Johnson, S. G. (2019). *The NLopt nonlinear-optimization package.* R version 3.5.2.

Jung, S., Ahn, J., and Jeon, Y. (2019). Penalized orthogonal iteration for sparse estimation of generalized eigenvalue problem. *Journal of Computational and Graphical Statistics*, 28(3):710–721.

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

Kraft, D. (1988). *A software package for sequential quadratic programming.* Technical Report DFVLR-FB 88-28. Institut für Dynamik der Flugsysteme, Oberpfaffenhofen.

Kraft, D. (1994). Algorithm 733: TOMP-Fortran modules for optimal control calculations. *ACM Transactions on Mathematical Software*, 20(3):262–281.

Kruskal, J. B. (1977). Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications*, 18(2):95–138.

Kruskal, J. B. (1989). Multiway data analysis. chapter Rank, Decomposition, and Uniqueness for 3-way and N-way Arrays, pages 7–18. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands.

Li, X., Xu, D., Zhou, H., and Li, L. (2018). Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences*, 10(3):520–545.

Li, Z., Suk, H., Shen, D., and Li, L. (2016). Sparse multi-response tensor regression for alzheimer's disease study with multivariate clinical assessments. *IEEE Transactions on Medical Imaging*, 35(8):1927–1936.

Lock, E. F. (2018). Tensor-on-tensor regression. *Journal of Computational and Graphical Statistics*, 27(3):638–647.

Lorenzi, M., Altmann, A., Gutman, B., Wray, S., Arber, C., Hibar, D. P., Jahanshad, N., Schott, J. M., Alexander, D. C., Thompson, P. M., and Ourselin, S. (2018). Susceptibility of brain atrophy to trib3 in alzheimer's disease, evidence from functional prioritization in imaging genetics. *Proceedings of the National Academy of Sciences*, 115(12):3162–3167.

McIntosh, A., Bookstein, F., Haxby, J., and Grady, C. (1996). Spatial pattern analysis of functional brain images using partial least squares. *NeuroImage*, 3(3):143 – 157.

Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate O(1/k2). *Soviet Mathematics Doklady*, 27(2):372–376.

O'Toole, A. J., Jiang, F., Abdi, H., and Haxby, J. V. (2005). Partially distributed representations of objects and faces in ventral temporal cortex. *Journal of Cognitive Neuroscience*, 17(4):580–590.

Parikh, N. and Boyd, S. (2014). Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239.

Parkhomenko, E., Tritchler, D., and Beyene, J. (2009). Sparse canonical correlation analysis with application to genomic data integration. *Statistical applications in genetics and molecular biology*, 8(1):Article 1.

Platt, S. R., Holmes, S. P., Howerth, E. W., Duberstein, K. J. J., Dove, C. R., Kinder, H. A., Wyatt, E. L., Linville, A. V., Lau, V. W., Stice, S. L., Hill, W. D., Hess, D. C., and West, F. D. (2014). Development and characterization of a yucatan miniature biomedical pig permanent middle cerebral artery occlusion stroke mode. *Experimental and translational stroke medicine*, 6:1–14.

Richard, E., Savalle, P.-A., and Vayatis, N. (2012). Estimation of simultaneously sparse and low rank matrices. In *Proceedings of the 29th International Conference on Machine Learning*, ICML'12, pages 51–58, USA. Omnipress.

Safo, S. E., Ahn, J., Jeon, Y., and Jung, S. (2018). Sparse generalized eigenvalue problem with application to canonical correlation analysis for integrative analysis of methylation and gene expression data. *Biometrics*.

Sidiropoulos, N. D. and Bro, R. (2000). On the uniqueness of multilinear decomposition of N-way arrays. *Journal of Chemometrics*, 14(3):229–239.

Signoretto, M., Tran Dinh, Q., De Lathauwer, L., and Suykens, J. A. K. (2014). Learning with tensors: a framework based on convex optimization and spectral regularization. *Machine Learning*, 94(3):303–351.

Sun, W., Wang, J., and Fang, Y. (2013). Consistent selection of tuning parameters via variable selection stability. *Journal of Machine Learning Research*, 14:3419–3440.

Tan, X., Zhang, Y., Tang, S., Shao, J., Wu, F., and Zhuang, Y. (2013). Logistic tensor regression for classification. In Yang, J., Fang, F., and Sun, C., editors, *Intelligent Science and Intelligent Data Engineering*, pages 573–581, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108.

Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311.

Vinod, H. (1976). Canonical ridge and econometrics of joint production. *Journal of Econometrics*, 4(2):147 – 166.

Waaijenborg, S., Verselewel De Witt Hamer, P. C., and Zwinderman, A. H. (2008). Quantifying the association between gene expressions and dna-markers by penalized canonical correlation analysis. *Statistical applications in genetics and molecular biology*, 7(1):Article 3.

Wilms, I. and Croux, C. (2015). Sparse canonical correlation analysis from a predictive point of view. *Biometrical Journal*, 57(5):834–851.

Witten, D. and Tibshirani, R. (2009). Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical applications in genetics and molecular biology*, 8(1):Article 28.

Witten, D., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534.

Wold, H. (1968). *Nonlinear Estimation by Iterative Least Square Procedures.* Wiley, New York, NY.

Worsley, K. J. (1997). An overview and some new developments in the statistical analysis of pet and fmri data. *Human Brain Mapping*, 5(4):254–258.

Ye, J. (1998). On measuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association*, 93(441):120–131.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *Ann. Statist.*, 38(2):894–942.

Zhou, H. (2017). *Matlab TensorReg Toolbox.* Version 1.0.

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

Zhou, H., Li, L., and Zhu, H. (2013). Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552. PMID: 24791032.

Zhou, J. and He, X. (2008). Dimension reduction based on constrained canonical correlation and variable filtering. *Ann. Statist.*, 36(4):1649–1668.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

Appendix

Proposition 1. *The higher-order singular value decomposition of a rank-$(R_1, \ldots, R_D)$ tensor $\mathcal{X}$ given in Algorithm 1 is an exact decomposition.*

Proof. Let $\mathcal{X}$ denote the original tensor and $\mathcal{X}^*$ denote the tensor obtained from the TKD of $\mathcal{X}$. We will show $\mathcal{X}^* = \mathcal{X}$.

Two facts about the n-mode product are needed:

1. $\mathcal{X} \times_m A \times_n B = \mathcal{X} \times_n B \times_m A$ for $m \neq n$.

2. $\mathcal{X} \times_n A \times_n B = \mathcal{X} \times_n (BA)$.

Then from the definition of the TKD, $\mathcal{X}^* = \mathcal{G} \times_1 B_1 \cdots \times_D B_D$, where the B_d's are the left singular vectors corresponding to the R_d nonzero singular values of $X_{(d)}$ and $\mathcal{G} = \mathcal{X} \times_1 B_1^T \cdots \times_D B_D^T$ (see Algorithm 1). Then $\mathcal{X}^*$ can be written as

$$\begin{aligned} \mathcal{X}^* &= \mathcal{G} \times_1 B_1 \cdots \times_D B_D \\ &= \mathcal{X} \times_1 B_1^T \cdots \times_D B_D^T \times_1 B_1 \cdots \times_D B_D. \end{aligned}$$

Applying facts 1 and 2, we can rearrange as

$$\begin{aligned}
\mathcal{X}^* &= \mathcal{X} \times_1 B_1^T \cdots \times_D B_D^T \times_1 B_1 \cdots \times_D B_D \\
&= \mathcal{X} \times_1 B_1^T \times_1 B_1 \cdots \times_D B_D^T \times_2 B_2 \cdots \times_D B_D \quad \text{(fact 1)} \\
&= \mathcal{X} \times_1 B_1 B_1^T \cdots \times_D B_D^T \times_2 B_2 \cdots \times_D B_D \quad \text{(fact 2).}
\end{aligned}$$

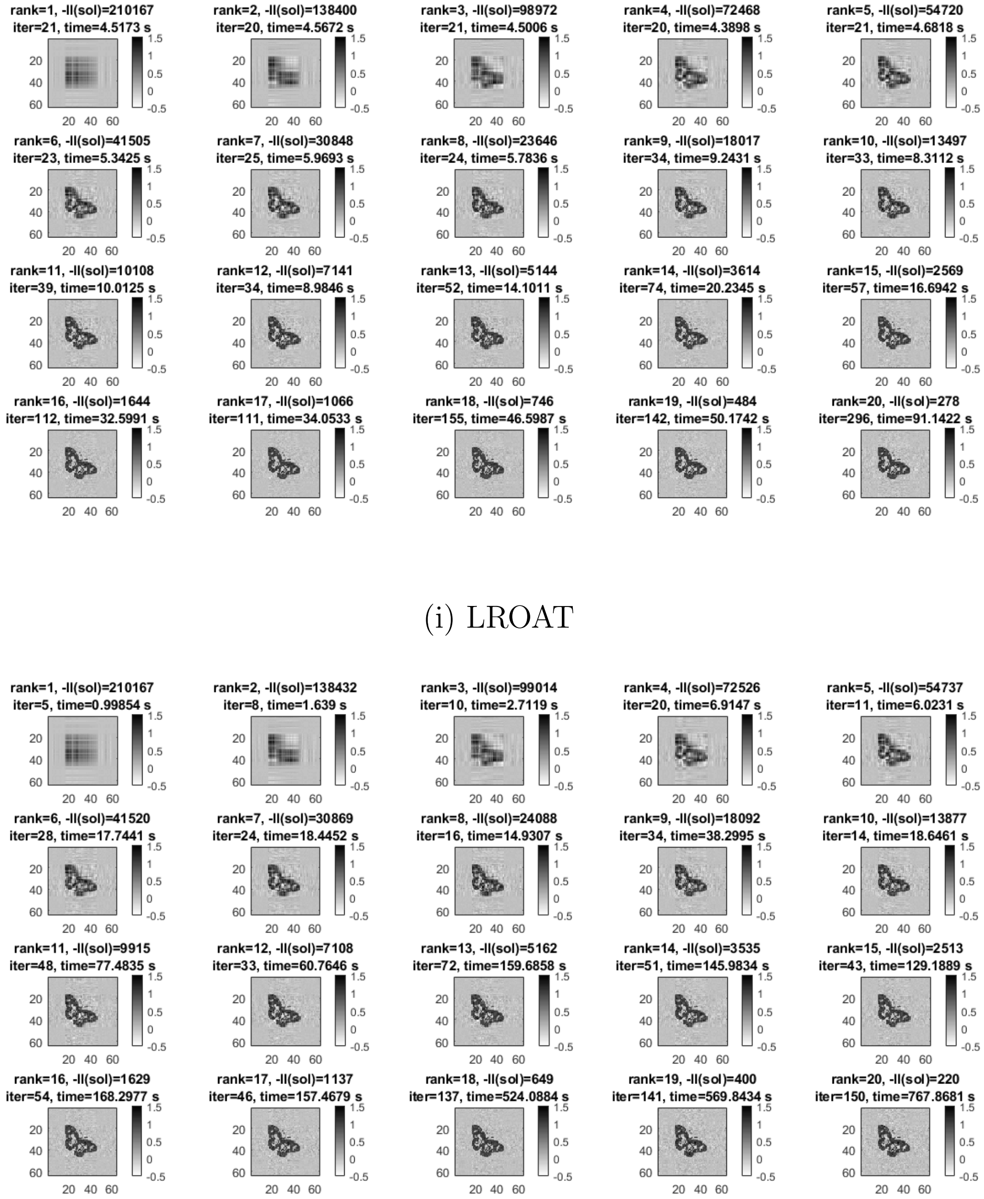
From the definition of the n-mode product

$$\mathcal{X} \times_1 B_1 B_1^T \iff B_1 B_1^T X_{(1)}.$$

But since B_1 are the left singular vectors from $SVD(X_{(1)})$, $B_1 B_1^T$ is a projection matrix to the column space of $X_{(1)}$. Thus,

$$B_1 B_1^T X_{(1)} = X_{(1)} \iff \mathcal{X} \times_1 B_1 B_1^T = \mathcal{X}.$$

The same arguments can be made for the other modes to show that $\mathcal{X}^* = \mathcal{X}$. $\square$



(i) LROAT

(ii) CPD

Figure 5.1: Estimates from the LROAT and CPD models for Experiment 1, Scenario 2.

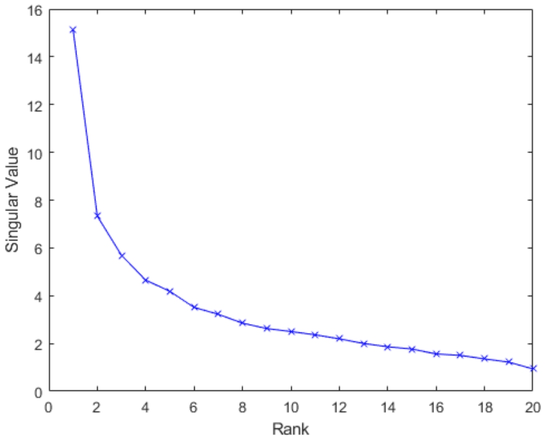


Figure 5.2: Scree plots of the singular values from the LROAT fit for Experiment 1, Scenario 2.

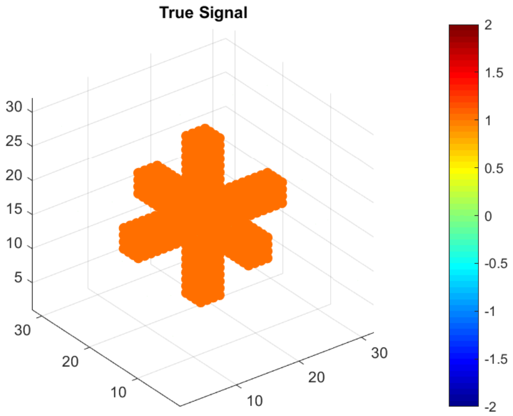
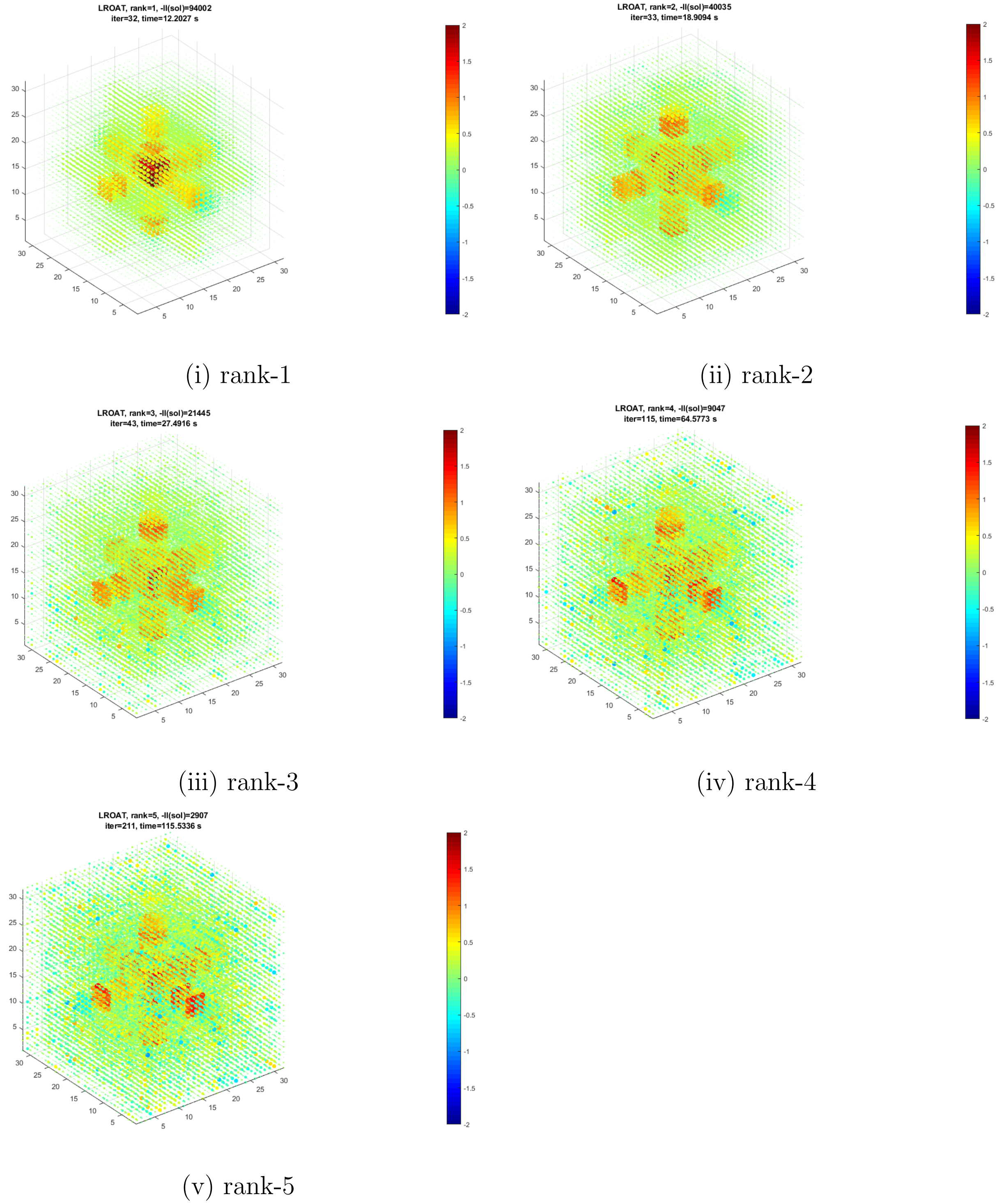


Figure 5.3: True signal for Experiment 2, Scenario 2.



(i) rank-1

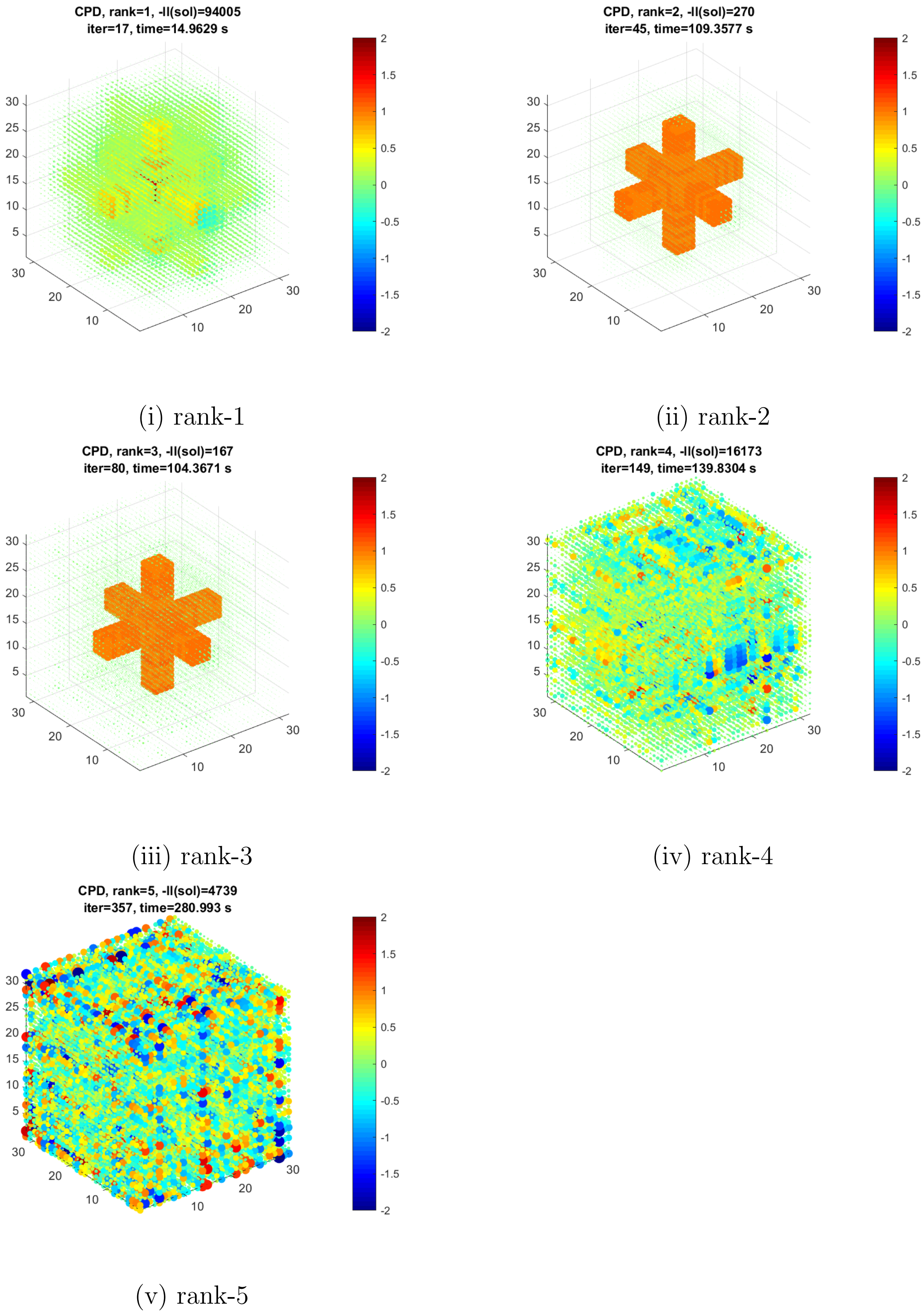
(ii) rank-2

(iii) rank-3

(iv) rank-4

(v) rank-5

Figure 5.4: Estimates from the LROAT model for Experiment 2, Scenario 2.



(i) rank-1

(ii) rank-2

(iii) rank-3

(iv) rank-4

(v) rank-5

Figure 5.5: Estimates from the CPD model for Experiment 2, Scenario 2.

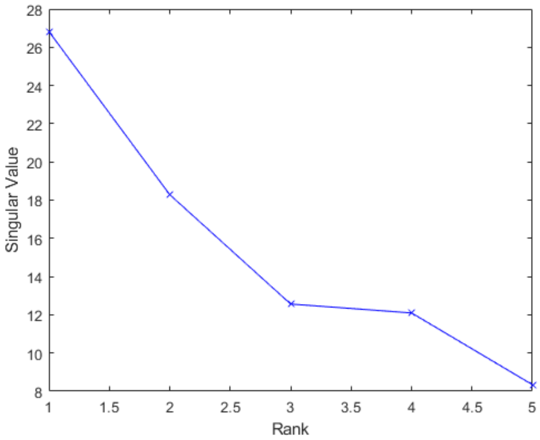


Figure 5.6: Scree plots of the singular values from the LROAT fit for Experiment 2, Scenario 2.

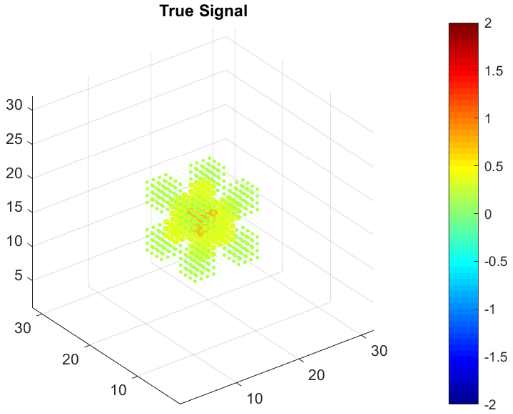
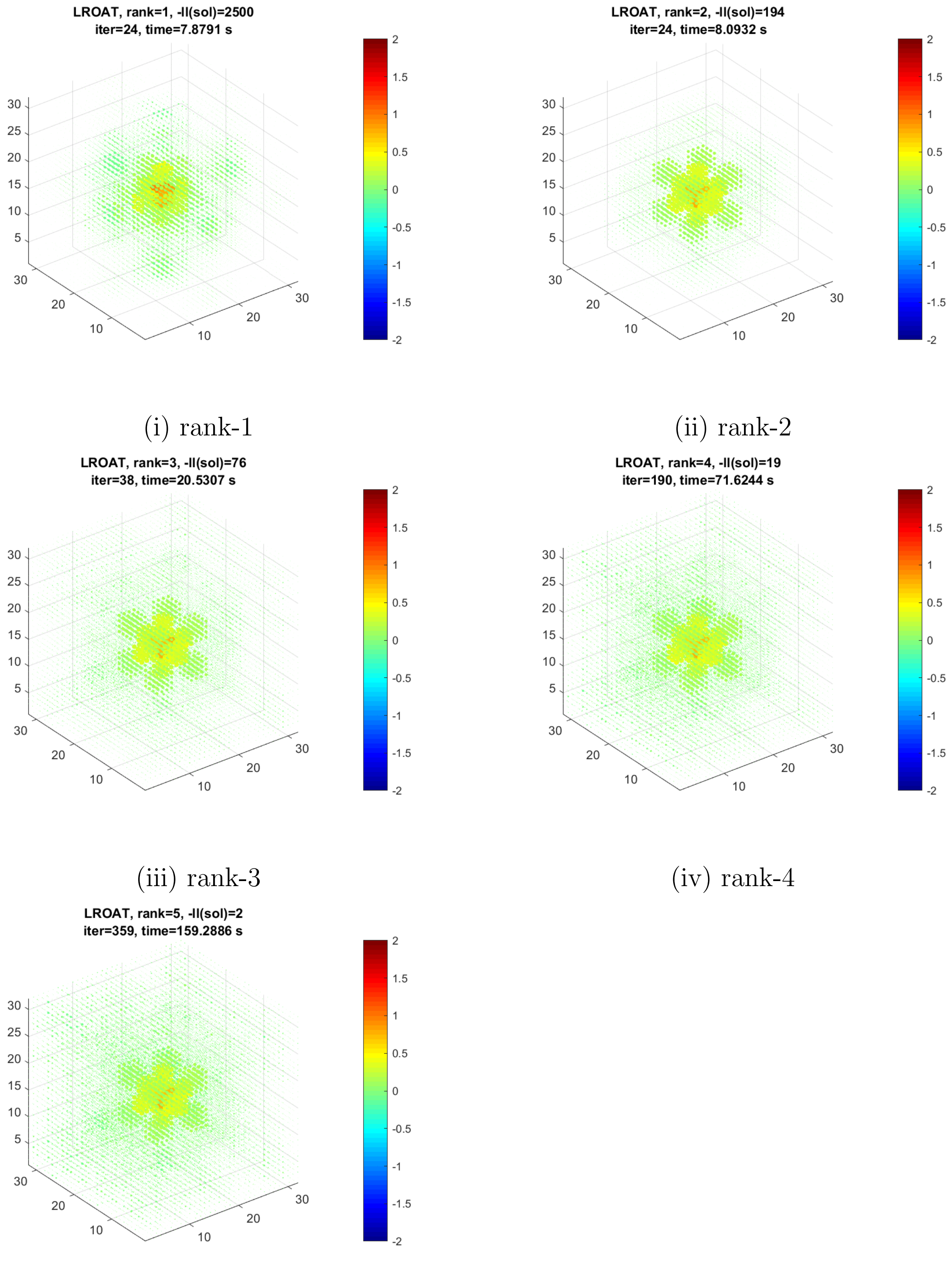


Figure 5.7: True signal for Experiment 2, Scenario 3.



(i) rank-1

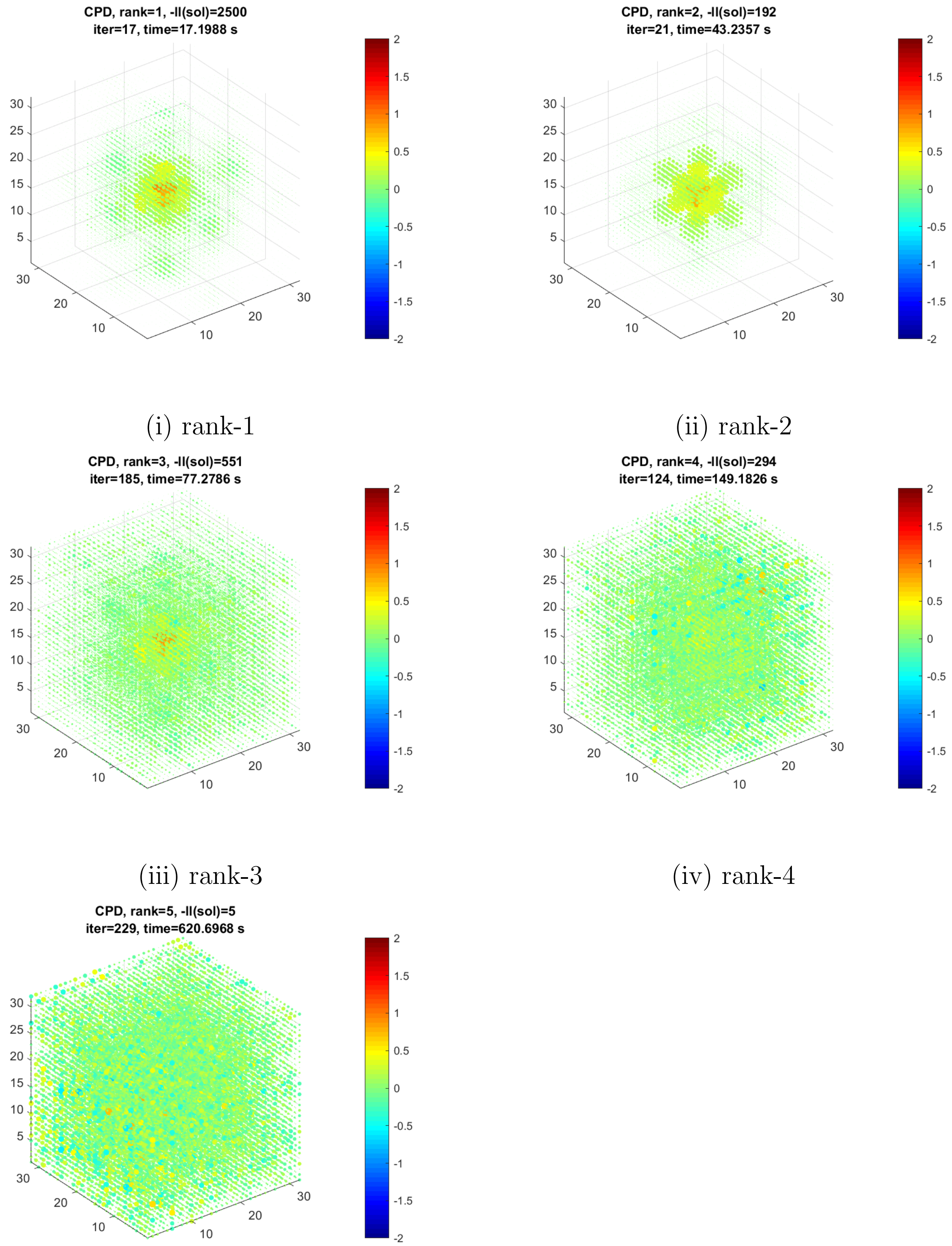
(ii) rank-2

(iii) rank-3

(iv) rank-4

(v) rank-5

Figure 5.8: Estimates from the LROAT model for Experiment 2, Scenario 3.



(i) rank-1

(ii) rank-2

(iii) rank-3

(iv) rank-4

(v) rank-5

Figure 5.9: Estimates from the CPD model for Experiment 2, Scenario 3.

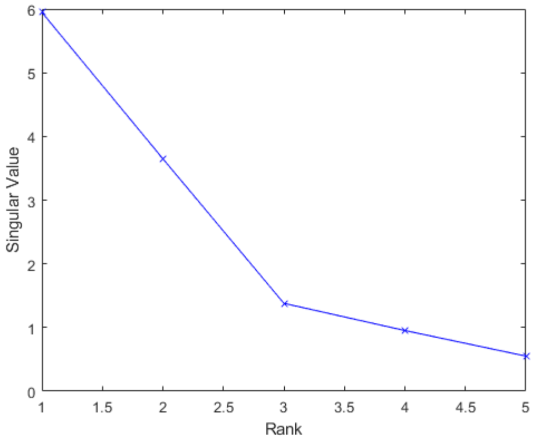


Figure 5.10: Scree plots of the singular values from the LROAT fit for Experiment 2, Scenario 3.

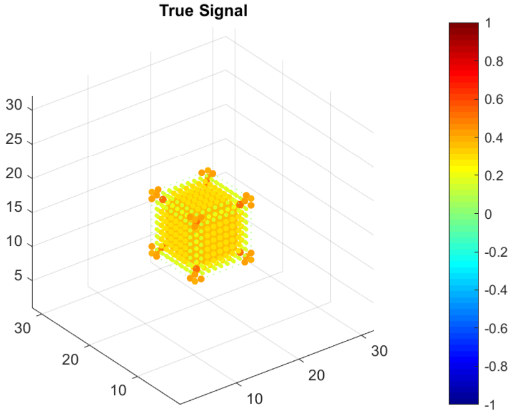
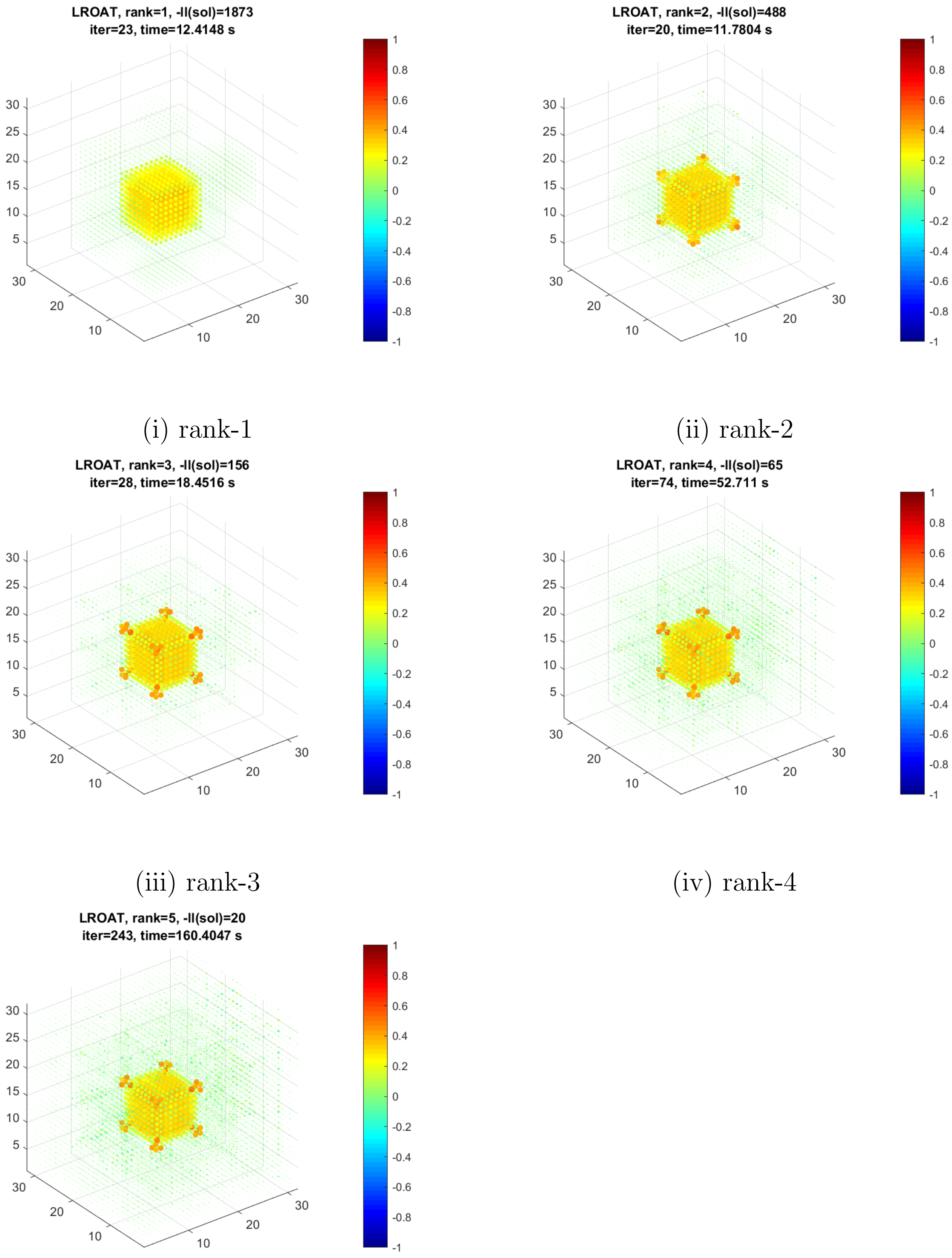


Figure 5.11: True signal for Experiment 2, Scenario 4.



(i) rank-1

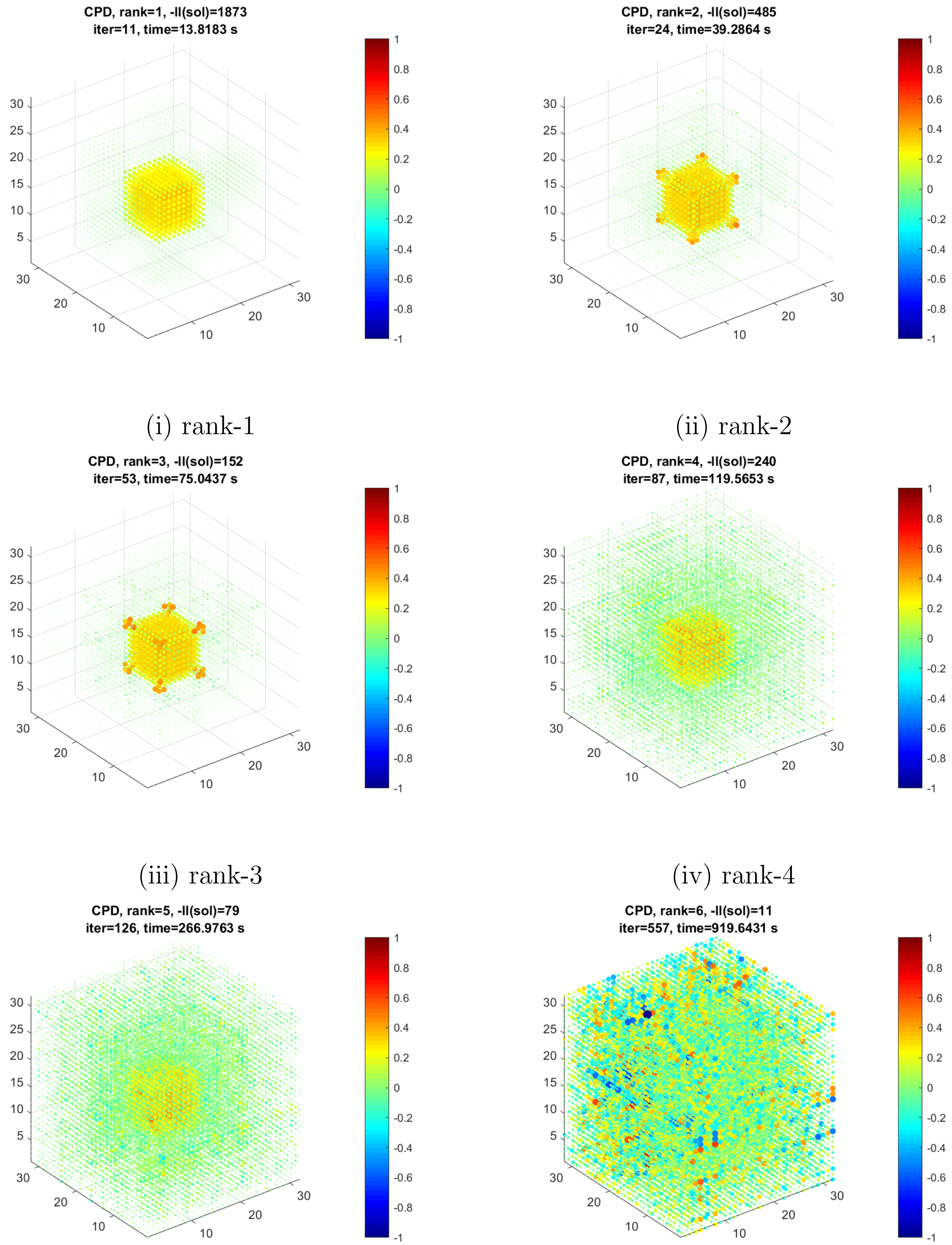
(ii) rank-2

(iii) rank-3

(iv) rank-4

(v) rank-5

Figure 5.12: Estimates from the LROAT model for Experiment 2, Scenario 4.



(i) rank-1

(ii) rank-2

(iii) rank-3

(iv) rank-4

(v) rank-5

(vi) rank-6

Figure 5.13: Estimates from the CPD model for Experiment 2, Scenario 4.

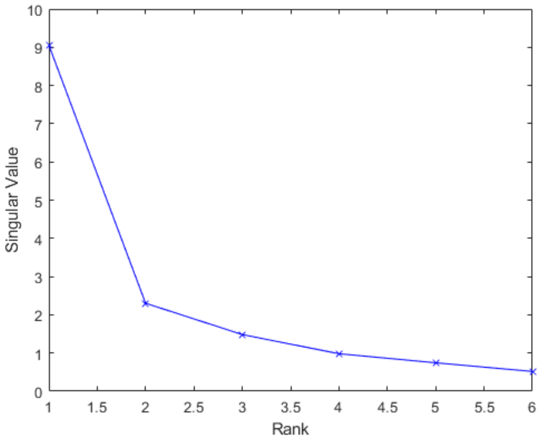


Figure 5.14: Scree plots of the singular values from the LROAT fit for Experiment 2, Scenario 4.

Table 5.1: Mean (SD) rank of signal variables in the X and Y datasets according to two importance metrics. Results based on 100 simulations from the multivariate normal distribution. Lower values of the rank are better (except for the Null scenario).

		X variables		Y variables	
Scenario	**Method**	Metric 1†	Metric 2‡	Metric 1†	Metric 2‡
AR(0.9), Identity	PMD (Witten et al. 2009)	24.88 (19)	24.14 (18)	17.58 (13)	16.57 (13)
$\rho = 0.9$	SELP-I (Safo et a. 2018)	25.94 (20)	24.67 (19)	17.09 (14)	16.43 (13)
90% sparsity	spLRMA	25.95 (18)	24.42 (17)	17.74 (13)	16.42 (12)
AR(0.9), Identity	PMD (Witten et al. 2009)	32.16 (25)	31.23 (24)	21.33 (18)	20.28 (18)
$\rho = 0.5$	SELP-I (Safo et a. 2018)	32.38 (27)	31.4 (25)	20.4 (19)	19.58 (18)
90% sparsity	spLRMA	24.29 (18)	21.37 (18)	15.86 (11)	13.57 (9)
AR(0.7), Identity	PMD (Witten et al. 2009)	42.36 (17)	41.33 (16)	29.02 (12)	28.14 (12)
$\rho = 0.9$	SELP-I (Safo et a. 2018)	43.13 (17)	41.41 (16)	29.17 (12)	27.83 (11)
90% sparsity	spLRMA	34.99 (13)	35.71 (13)	22.14 (10)	22.88 (10)
AR(0.7), Identity	PMD (Witten et al. 2009)	48.59 (19)	47.33 (19)	31.89 (14)	30.64 (14)
$\rho = 0.7$	SELP-I (Safo et a. 2018)	49.55 (19)	47.11 (18)	31.89 (14)	30.4 (13)
90% sparsity	spLRMA	33.66 (15)	35.52 (14)	19.98 (9)	21.37 (9)
AR(0.7), Identity	PMD (Witten et al. 2009)	54.73 (22)	53.29 (22)	37.13 (18)	35.46 (17)
$\rho = 0.5$	SELP-I (Safo et a. 2018)	55.57 (22)	52.72 (21)	37.03 (17)	34.51 (16)
90% sparsity	spLRMA	31.26 (16)	34.69 (15)	17.8 (8)	19.72 (8)
AR(0.7), Identity	PMD (Witten et al. 2009)	61.12 (27)	59.56 (27)	43.2 (22)	41.44 (22)
$\rho = 0.3$	SELP-I (Safo et a. 2018)	62.62 (26)	58.79 (25)	42.74 (22)	39.75 (22)
90% sparsity	spLRMA	30.43 (16)	34.81 (16)	16.82 (8)	19.6 (9)
AR(0.2), Identity	PMD (Witten et al. 2009)	84.32 (18)	83.18 (18)	66.78 (17)	65.99 (17)
$\rho = 0.9$	SELP-I (Safo et a. 2018)	85.29 (18)	82.99 (19)	66.75 (17)	65.8 (17)
90% sparsity	spLRMA	82.23 (16)	81.15 (17)	63.84 (14)	63.49 (15)
CS(0.2), Identity	PMD (Witten et al. 2009)	46.74 (28)	46.11 (28)	34.69 (20)	34.27 (20)
$\rho = 0.9$	SELP-I (Safo et a. 2018)	46.24 (29)	44.82 (28)	33.93 (21)	33.21 (21)
90% sparsity	spLRMA	31.03 (25)	31.04 (26)	21.4 (15)	21.48 (16)
CS(0.2), Identity	PMD (Witten et al. 2009)	53.03 (35)	52.88 (35)	42.55 (26)	42.61 (26)
$\rho = 0.5$	SELP-I (Safo et a. 2018)	52.6 (37)	51.23 (36)	41.27 (27)	41 (27)
90% sparsity	spLRMA	33.2 (27)	32.71 (29)	24.84 (19)	24.66 (20)
Identity, Identity	PMD (Witten et al. 2009)	94.37 (14)	93.7 (14)	74.51 (12)	73.84 (12)
$\rho = 0.9$	SELP-I (Safo et a. 2018)	94.27 (14)	93.36 (15)	75.21 (12)	74.3 (12)
90% sparsity	spLRMA	92.69 (14)	90.86 (14)	73.23 (12)	72.03 (12)
Identity, Identity	PMD (Witten et al. 2009)	84.94 (27)	83.6 (28)	66.88 (22)	65.7 (22)
$\rho = 0.9$	SELP-I (Safo et a. 2018)	86.38 (27)	84.98 (27)	67.38 (22)	65.1 (21)
95% sparsity	spLRMA	82.26 (20)	79.62 (20)	63.04 (18)	61.6 (19)
AR(0.7), AR(0.6)	PMD (Witten et al. 2009)	53.7 (31)	53.59 (31)	39.84 (25)	39.22 (25)
$\rho = 0.9$	SELP-I (Safo et a. 2018)	53.14 (29)	52.7 (29)	38.8 (25)	38.03 (25)
90% sparsity	spLRMA	52.6 (22)	52.95 (24)	36.3 (20)	35.04 (20)
CS(0.7), Identity	PMD (Witten et al. 2009)	34.44 (42)	33.16 (41)	45.43 (38)	42.97 (38)
Null	SELP-I (Safo et a. 2018)	40.79 (46)	37.16 (43)	42.73 (41)	39.11 (38)
($\rho = 0$)	spLRMA	19.82 (23)	17.8 (22)	15.34 (16)	13.28 (14)

†1–No. of times a variables was selected

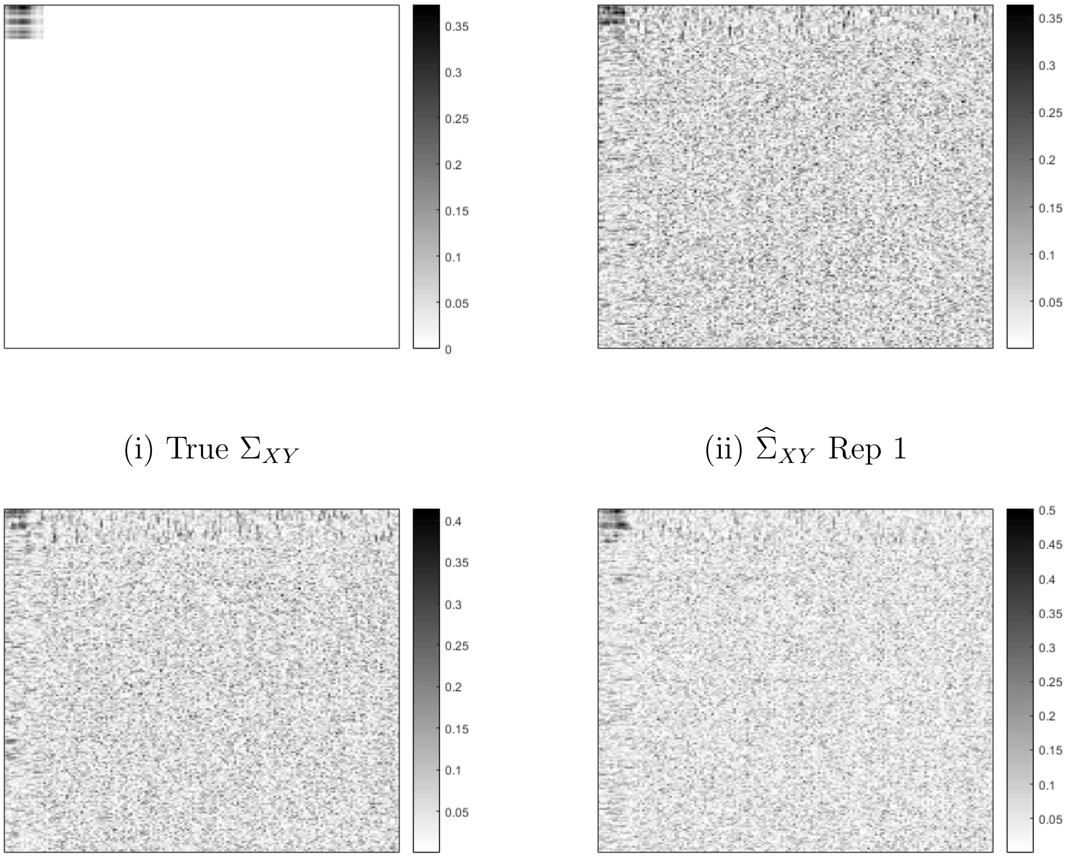
‡2–Magnitude of canonical vector loading

Table 5.2: Mean (SD) rank of signal variables in the X and Y datasets according to two importance metrics. Results based on 100 simulations from the multivariate $t(df = 15)$ distribution. Lower values of the rank are better.

		X variables		Y variables	
Scenario	**Method**	Metric 1[†]	Metric 2[‡]	Metric 1[†]	Metric 2[‡]
AR(0.7), Identity	PMD (Witten et al. 2009)	50.85 (25)	49.93 (24)	37.41 (20)	36.66 (20)
$\rho = 0.9$	SELP-I (Safo et a. 2018)	51.07 (24)	49.63 (23)	36.76 (20)	35.51 (19)
90% sparsity	spLRMA	47.32 (24)	47.12 (23)	31.66 (16)	31.67 (16)
AR(0.7), Identity	PMD (Witten et al. 2009)	72.18 (26)	70.56 (26)	52.43 (23)	51.5 (23)
$\rho = 0.5$	SELP-I (Safo et a. 2018)	72.05 (26)	69.8 (26)	51.58 (23)	50.24 (23)
90% sparsity	spLRMA	57.38 (26)	58.18 (25)	38.55 (21)	39.83 (21)
CS(0.2), Identity	PMD (Witten et al. 2009)	68.99 (34)	68.43 (34)	58.77 (24)	58.37 (24)
$\rho = 0.9$	SELP-I (Safo et a. 2018)	69.29 (35)	68.06 (34)	59.55 (25)	58.43 (24)
90% sparsity	spLRMA	63.75 (27)	61.85 (29)	54.36 (19)	53.39 (21)
CS(0.2), Identity	PMD (Witten et al. 2009)	80.23 (31)	80.52 (30)	68.43 (23)	68.46 (22)
$\rho = 0.5$	SELP-I (Safo et a. 2018)	80.25 (31)	79.67 (31)	68.81 (24)	68.19 (23)
90% sparsity	spLRMA	74.64 (25)	73.78 (27)	63.07 (19)	62.99 (20)

[†] 1–No. of times a variables was selected

[‡] 2–Magnitude of canonical vector loading



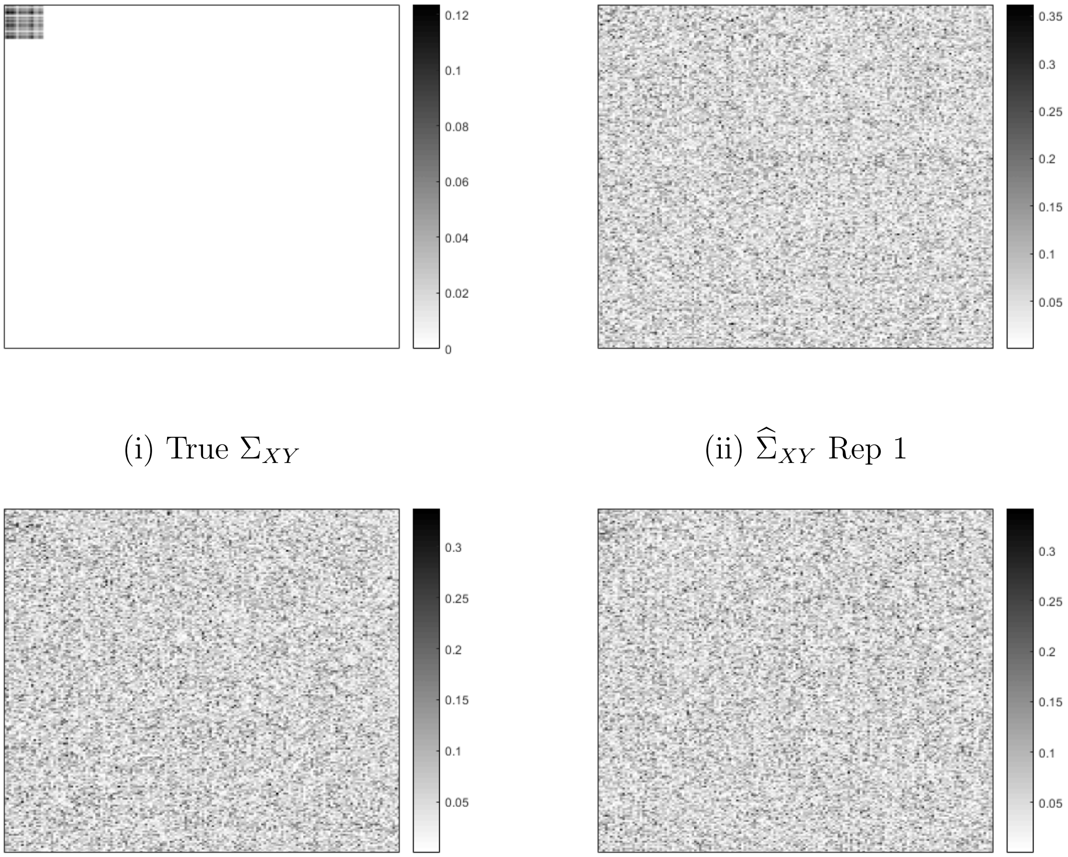
(i) True Σ_{XY}

(ii) $\widehat{\Sigma}_{XY}$ Rep 1

(iii) $\widehat{\Sigma}_{XY}$ Rep 2

(iv) $\widehat{\Sigma}_{XY}$ Rep 3

Figure 5.15: True Σ_{XY} vs. observed $\widehat{\Sigma}_{XY}$ for several datasets generated from scenario with within-set covariances $BD\left[AR(0.7),\ I\right],\ \rho = 0.9$.



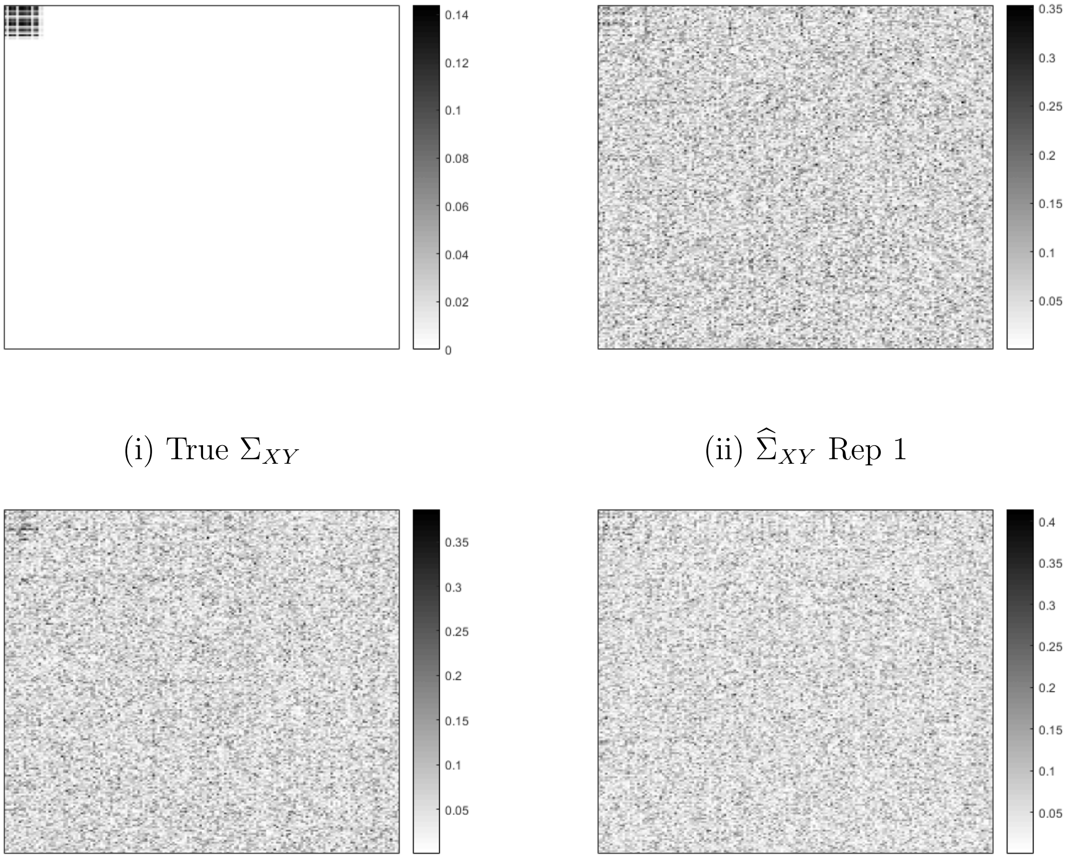
(i) True Σ_{XY}

(ii) $\widehat{\Sigma}_{XY}$ Rep 1

(iii) $\widehat{\Sigma}_{XY}$ Rep 2

(iv) $\widehat{\Sigma}_{XY}$ Rep 3

Figure 5.16: True Σ_{XY} vs. observed $\widehat{\Sigma}_{XY}$ for several datasets generated from scenario with within-set covariances $BD\left[AR(0.2),\ I\right],\ \rho = 0.9$.



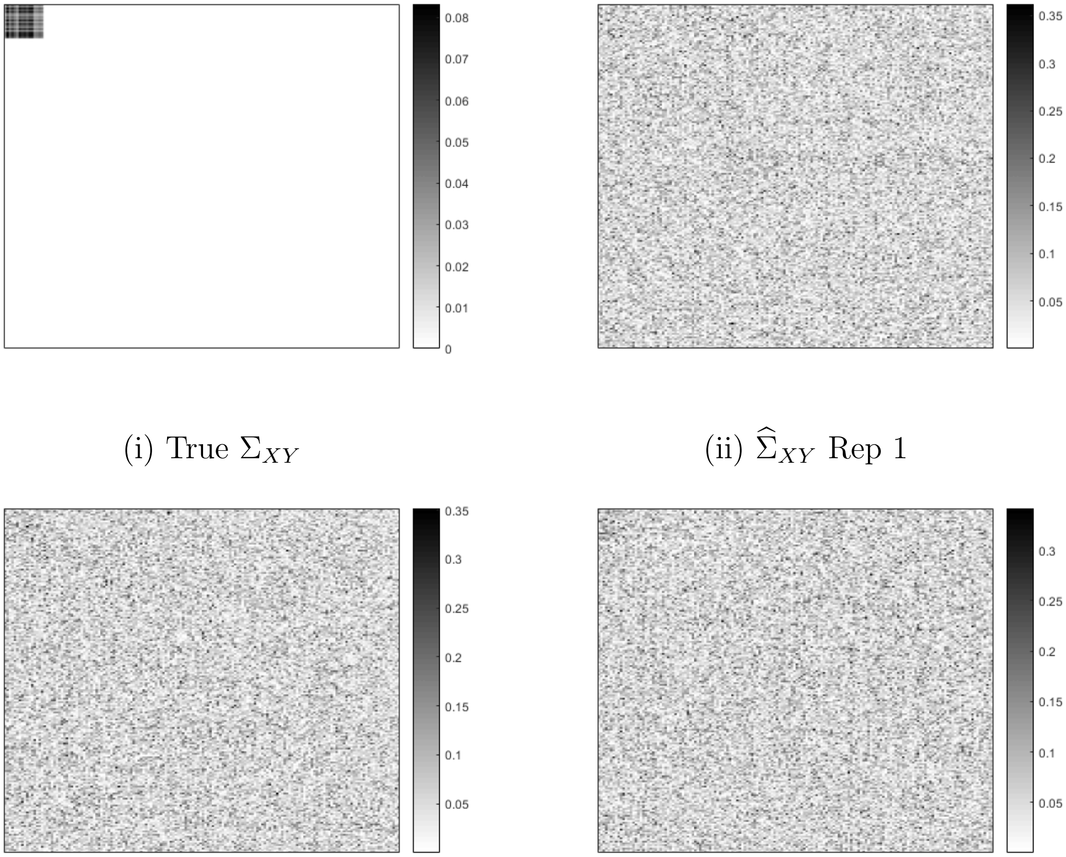
(i) True Σ_{XY}

(ii) $\widehat{\Sigma}_{XY}$ Rep 1

(iii) $\widehat{\Sigma}_{XY}$ Rep 2

(iv) $\widehat{\Sigma}_{XY}$ Rep 3

Figure 5.17: True Σ_{XY} vs. observed $\widehat{\Sigma}_{XY}$ for several datasets generated from scenario with within-set covariances $BD\left[CS(0.2),\ I\right],\ \rho = 0.9$.



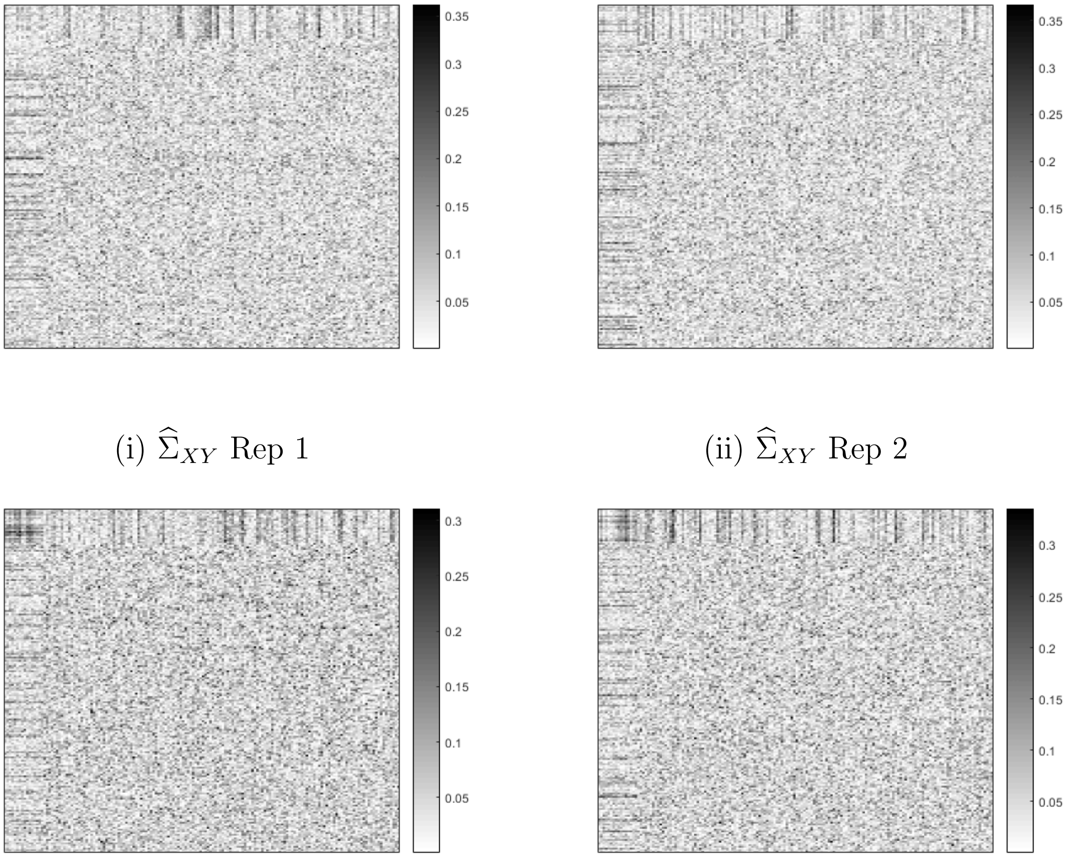
(i) True Σ_{XY}

(ii) $\widehat{\Sigma}_{XY}$ Rep 1

(iii) $\widehat{\Sigma}_{XY}$ Rep 2

(iv) $\widehat{\Sigma}_{XY}$ Rep 3

Figure 5.18: True Σ_{XY} vs. observed $\widehat{\Sigma}_{XY}$ for several datasets generated from scenario with within-set covariances $BD\,[I,\ I]\,,\ \rho = 0.9$.



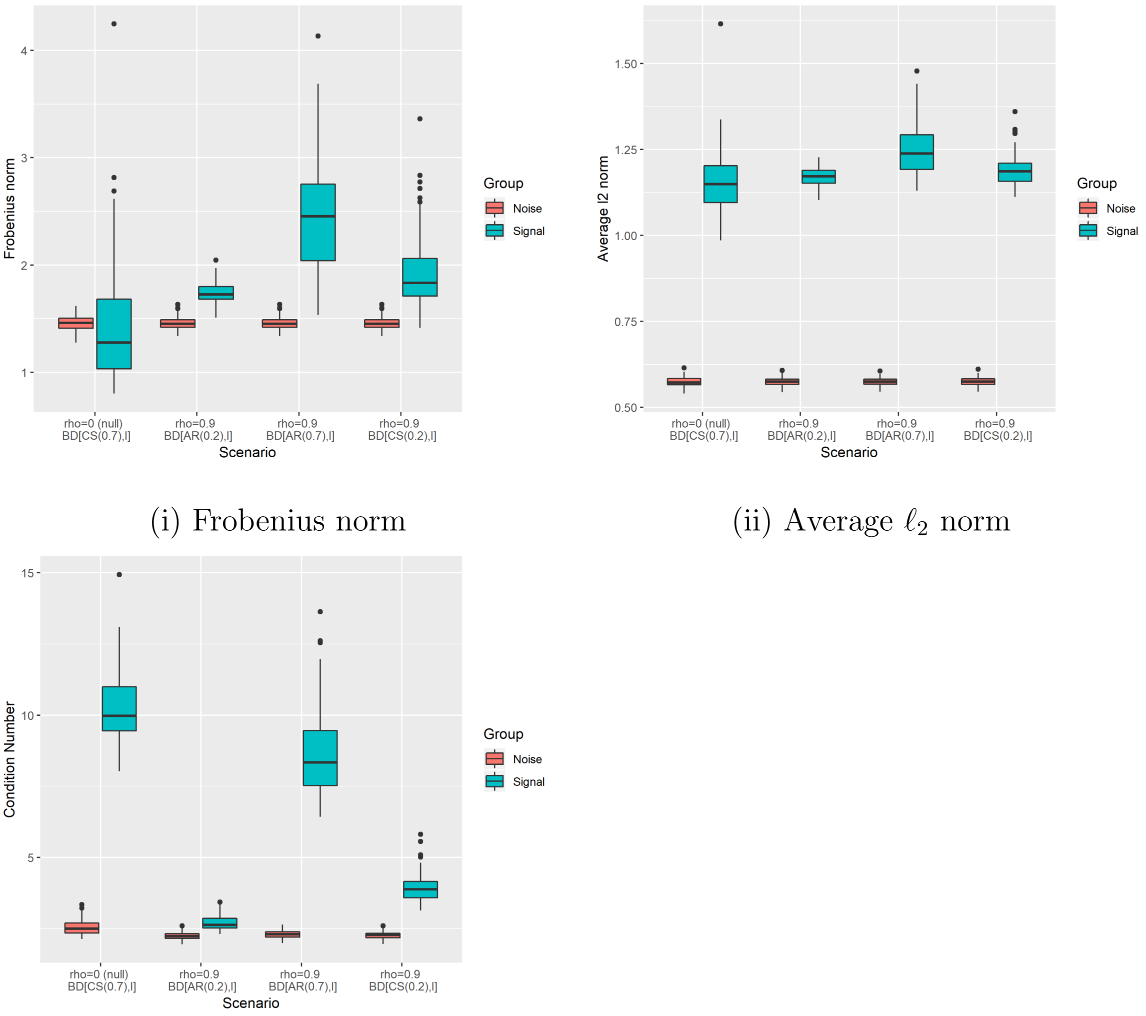
(i) $\widehat{\Sigma}_{XY}$ Rep 1

(ii) $\widehat{\Sigma}_{XY}$ Rep 2

(iii) $\widehat{\Sigma}_{XY}$ Rep 3

(iv) $\widehat{\Sigma}_{XY}$ Rep 4

Figure 5.19: Observed $\widehat{\Sigma}_{XY}$ for several datasets generated from Null Scenario (true $\Sigma_{XY} = 0$, but Σ_{XX} and Σ_{YY} are $BD\,[CS(0.7),\ I]$).



(i) Frobenius norm

(ii) Average ℓ_2 norm

(iii) Condition number

Figure 5.20: Measures associated with the strength of signal and degree of linear dependence of elements/columns of $\widehat{\Sigma}_{XY}$ corresponding to the signal variables (with an equal number of noise variables included for reference). Boxplots are based on 100 realizations of $\widehat{\Sigma}_{XY}$.