Statistical Inference on Multiple Graphs

by

Shangsi Wang

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

December, 2017

Abstract

Given multiple graphs, an important question is how to perform statistical inference on them. This question becomes more significant in the recent era with the explosion of graph data and the increasing complexity of data analysis. Successfully addressing this question will have a large impact on various scientific fields including neuroscience, social network analysis, and internet mapping. Graphs are naturally complex objects with intrinsic topological structure which imposes significant challenges to traditional statistical inference. Therefore, graph pre-processing, feature extraction, and dimension reduction are essential to obtain good subsequent inference performance.

In this dissertation, I develop pre-processing, feature extraction, and dimension reduction methods for data taking the form of multiple graphs. The methods are motivated by classical statistical approaches including analysis of variance, feature screening, and principal component analysis. Some methods can be applied under both supervised and unsupervised settings; others are designed only for problems involving labels of interest. I analyze the theoretical properties of these methods

ABSTRACT

jointly with subsequent inference performance under suitable random graph models. Simulations, which include graph clustering, classification, and regression are provided to demonstrate the properties of the proposed methods. I further apply the methods developed here to real data sets such as human brain networks acquired through neuroimaging techniques.

The main contribution of this dissertation is the presentation of a set of methods in analyzing multiple graphs. These methods are supported with theory and numerical experiments. I further demonstrate the utility of the methods by exploring real data sets and discovering statistical patterns.

Primary Reader: Dr. Carey E. Priebe

Secondary Reader: Dr. Joshua T. Vogelstein

Tertiary Reader: Dr. Cencheng Shen

Acknowledgments

First and foremost, I would like to express my deep gratitude to my advisor, Dr. Carey E. Priebe, for his endless effort spent in advising me. He introduced me to this fascinating world of statistics and encouraged me to explore unknown paths in the field. Through years, he has provided me with insightful guidance and generous financial assistance. Dr. Priebe has made my Ph.D. life at Johns Hopkins a productive and enjoyable experience. For everything you have done for me, I truly appreciate all of it.

I would also like to give special thanks to Dr. Joshua T. Vogelstein for his guidance and help in research. He shed light on problems when I was crawling in the dark. Dr. Joshua has greatly expanded my knowledge in practical statistics and neuroscience. He has helped me improving my skills in computation, writing, and presentation significantly. I thank him for the invaluable meetings we have.

Through my Ph.D. life, I have collaborated with many successful researchers. It is my great honor to work with them. I also have learned from many great minds. I would like to thank them for letting me pick their brains. I have received help from

many wonderful professionals as well. I would love to thank them for organizing my life. To my friends who shared this adventure with me, I thank you for making this experience full of fun and memorable.

Finally, my sincere thanks to my parents for always being there for me.

Dedication

This thesis is dedicated to my parents, Xiaozhi Zhang and Xiulong Wang. I would not be the person I am today without their infinite support and unconditional love.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

In many fields of science, graphs naturally appear to model complex relationship between objects. In neuroimaging, graphs are used to model connectivity between neurons or regions of brain [1]. In chemical engineering, graphs are used to represent structure of interactions between molecules [2]. In social networks, graphs are used to capture interactions between users [3]. Graphs are often high-dimensional objects with complicated topological structure, which makes many classical machine learning algorithms not immediately applicable. This dissertation concerns graph embedding, screening, and pre-processing when given multiple graphs, and investigates their applications in graph clustering, classification, and regression.

Classical graph embedding techniques such as Adjacency Spectral Embedding (ASE) [4] and Laplacian Eigenmap (LE) [5] were proposed to embed a single graph observation. Given a set of graphs $\{G_i\}_{i=1}^m$, ASE and LE need to embed individual

adjacency matrix or Laplacian matrix of G_i separately, and there is no easy way to combine multiple embeddings. We propose a joint embedding method which considers the set of graphs simultaneously. The joint embedding takes a matrix factorization approach to extract features for multiple graphs. The joint embedding manages to simultaneously identify a set of rank one matrices and project adjacency matrices into the linear subspace spanned by this set of matrices. We discuss this method in detail in Chapter 2 which is based on our paper [6].

In the face of high-dimensional data, it is difficult to apply traditional machine learning methods directly due to computational complexity and instability. To overcome such challenges, Fan and Lv [7] propose the sure independence screening to identify a subset of important predictors and showed that ranking variables using Pearson correlation possesses a sure screening property in linear regression models. We develop a vertex screening method to locate a small signal graph of interest within a big graph. The vertex screening algorithm utilizes distance-based correlation to screen the vertices and recovers the set of signal vertices with high probability. We study the vertex screening jointly with graph classification in Chapter 3.

Chapter 4 considers the graph pre-processing problem. Collecting and processing some data sets requires massive institutional efforts, and it is often the case that data collectors do not have a single explicit inference task [8, 9]. In this case, optimally addressing experimental design decisions can yield significant savings in both the financial and human costs and also improve accuracy of analytical results [10–12]. To

this end, we propose and develope a formal framework of discriminability to guide data collection and processing. The framework utilizes subject labels to compare different experiment design and data pre-processing options. We extensively investigate the utility of this framework in the brain imaging studies and successfully find the optimal pipeline to convert the raw magnetic resonance imaging signal to a graph of neural connections of the human brain.

Chapter 2

Joint Embedding of Graphs

2.1 Introduction

The graphs are naturally high-dimensional objects with complicated topological structure, which makes graph clustering and classification a challenge to traditional machine learning algorithms. Therefore, feature extraction and dimension reduction techniques are helpful in the applications of learning graph data. In this chapter, we present an algorithm to jointly embed multiple graphs into low-dimensional space, which is primarily based on paper [6]. We demonstrate through theory and experiments that the joint embedding algorithm produces features which lead to state of the art performance for subsequent inference tasks on graphs.

There exist a few unsupervised approaches to extract features from graphs. First, classical Principal Component Analysis can be applied by treating each edge of a

graph as a raw feature [13]. This approach produces features which are linear combinations of edges, but it ignores the topological structure of graphs and the features extracted are not easily interpretable. Second, features can be extracted by computing summary topological and label statistics from graphs [14, 15]. These statistics commonly include number of edges, number of triangles, average clustering coefficient, maximum effective eccentricity, etc. In general, it is hard to know what intrinsic statistics to compute *a priori* and computing some statistics can be computationally expensive. Third, many frequent subgraph mining algorithms are developed [16]. For example, the fast frequent subgraph mining algorithm can identify all connected subgraphs that occur in a large fraction of graphs in a graph data set [17]. Finally, spectral feature selection can also be applied to graphs. It treats each graph as a node and constructs an object graph based on a similarity measure. Features are computed through the spectral decomposition of this object graph [18].

Adjacency Spectral Embedding (ASE) and Laplacian Eigenmap (LE) are proposed to embed a single graph observation [4, 5]. The inference task considered in these papers is learning of the block structure of the graph or clustering vertices. Given a set of graphs $\{G_i = (V_i, E_i)\}_{i=1}^m$, ASE and LE need to embed an adjacency matrix or Laplacian matrix of G_i individually, and there is no easy way to combine multiple embeddings. The joint embedding method considers the set of graphs together. It takes a matrix factorization approach to extract features for multiple graphs. The algorithm manages to simultaneously identify a set of rank one matrices and project

adjacency matrices into the linear subspace spanned by this set of matrices. The joint embedding can be understood as a generalization of ASE for multiple graphs. We demonstrate through simulation experiments that the joint embedding algorithm extracts features which lead to good performance for a variety of inference tasks. In the next section, we review some random graph models and present a model for generating multiple random graphs. In Section 2.3, we define the joint embedding of graphs and present an algorithm to compute it. In Section 2.4, we perform some theoretical analyses of our joint embedding algorithm. The theoretical results and real data experiments are explored in Section 2.5. We conclude the chapter with a brief discussion of implications and possible future work.

2.2 Random Graph Model

Before discussing the joint embedding algorithm, we first review a few classical random graph models. These models are often used to model a single graph observation. Then, we introduce a random graph model which can be used to model multiple graphs.

2.2.1 Stochastic Block Model

In this section, we present the Stochastic Block Model, which is first introduced in [19]. It is a family of random graph models such that the set of n vertices are

assigned randomly to K blocks and the edge probability between two vertices are determined by their block membership. It is a very popular models on graphs which is frequently used to capture community structure of networks [20, 21].

Definition Stochastic Block Model (SBM). Let π be a prior probability vector for block membership which lies in the unit $K-1$-simplex. Denote by $\tau = (\tau_1, \tau_2, ..., \tau_n) \in [K]^n$ the block membership vector, where τ is a multinomial sequence with probability vector π. Denote by $\mathbf{B} \in [0,1]^{K \times K}$ the block connectivity probability matrix. Suppose $\mathbf{A}$ is a random adjacency matrix given by,

$$\mathbb{P}(\mathbf{A}|\tau, \mathbf{B}) = \prod_{u<v} \mathbf{B}_{\tau_u,\tau_v}^{\mathbf{A}_{uv}} (1 - \mathbf{B}_{\tau_u,\tau_v})^{1-\mathbf{A}_{uv}}$$

Then, $\mathbf{A}$ is an adjacency matrix of a K-block stochastic block model graph, and the notation is $\mathbf{A} \sim SBM(\pi, \mathbf{B})$. Sometimes, τ may also be treated as the parameter of interest, in this case the notation becomes $\mathbf{A} \sim SBM(\tau, \mathbf{B})$.

2.2.2 Homogeneous and Inhomogeneous Erdos Renyi model

In this section, we introduce the homogeneous and inhomogeneous Erdos-Renyi random graph model [22].

Definition Inhomogeneous Erdos-Renyi model (IER). A random adjacency matrix $\mathbf{A}$ is said to follow an inhomogeneous Erdos-Renyi random graph model with edge

probability matrix $\mathbf{P} \in [0,1]^{n \times n}$, if the edge probability between vertex u and v is $\mathbf{P}[u,v]$ and independent of other edges. The notation is $\mathbf{A} \sim IER(\mathbf{P})$, and the likelihood of $\mathbf{A}$ under this model is

$$\mathcal{L}(\mathbf{A};\mathbf{P}) = \prod_{u<v} (\mathbf{P}[u,v])^{\mathbf{A}[u,v]} (1-\mathbf{P}[u,v])^{1-\mathbf{A}[u,v]}.$$

When $\mathbf{P}$ is a constant matrix with all its entries equal to p, then we say $\mathbf{A}$ follow a homogeneus Erdos Renyi model with edge probability p. Homogeneus Erdos Renyi model can be understood as a SBM with 1 block. Similarly, inhomogeneous Erdos Renyi model can be understood as a SBM with n block.

2.2.3 Random Dot Product Graph

In this section, we present the Random Dot Product Graph (RDPG) which is proposed in [23]. It is a special case of the Latent Position Model [24]. Specifically, each vertex has a latent position and the edge probability between two vertices is the inner product their latent positions. The RDPG is a convenient model which is designed to capture more complex structures than SBM. The RDPG can be further generalized to Latent Position Graph by replacing the inner product by a kernel [25]. The formal definition of RDPG is as following:

Definition Random Dot Product Graph (RDPG). Let F be a distribution on a set $\mathcal{X} \in \mathbb{R}^d$ satisfying $x^T y \in [0,1]$ for all $x, y \in \mathcal{X}$. Let $\mathbf{X} = [x_1^T, x_2^T, ..., x_n^T] \in \mathcal{X}^n$. The notation is $(\mathbf{X}, \mathbf{A}) \sim RDPG(F)$, if the x_i are independent and identically distributed

according to F, and conditioned on $\mathbf{X}$, the $\mathbf{A}_{uv}$ are independent Bernoulli random variables,

$$\mathbf{A}_{uv} \sim Bernoulli(x_u^T x_v).$$

Alternatively,

$$\mathbb{P}(\mathbf{A}|\mathbf{X}) = \prod_{u<v} (x_u^T x_v)^{\mathbf{A}_{uv}} (1 - x_u^T x_v)^{1-\mathbf{A}_{uv}}.$$

Also, define $\mathbf{P} := \mathbf{X}\mathbf{X}^T$ to be edge probability matrix. When the latent positions $\mathbf{X}$ is regarded as parameter, the notation becomes $\mathbf{A} \sim RDPG(\mathbf{X})$.

2.2.4 Multiple Random Eigen Graphs

In this section, we present the Multiple Random Eigen Graphs (MREG) model which is proposed in [6]. It can be used to model multiple networks and is defined as following:

Definition Multiple Random Eigen Graphs (MREG). Let $\{h_k\}_{k=1}^d$ be a set of norm-1 vectors in $\mathbb{R}^n$, and F be a distribution on a set $\mathcal{X} \in \mathbb{R}^d$, satisfying $\sum_{k=1}^{d} \lambda[k] h_k h_k^T \in [0,1]^{n \times n}$ for all $\lambda \in \mathcal{X}$, where $\lambda[k]$ is the kth entry of vector λ. The m pairs $\{(\lambda_i, \mathbf{A}_i)\}_{i=1}^m$ follow a d-dimensional multiple random eigen graphs model, and the notation is

$$\{(\lambda_i, \mathbf{A}_i)\}_{i=1}^m \sim MREG(F, h_1, ..., h_d),$$

if $\{\lambda_i\}_{i=1}^m$ is independent and identically distributed according to distribution F, and

conditioned on λ_i, the entries of A_i are independent Bernoulli random variables,

$$\mathbf{A}_i[u,v] \sim Bernoulli(\sum_{k=1}^{d} \lambda_i[k] h_k[u] h_k[v]).$$

$\mathbf{P}_i := \sum_{k=1}^{d} \lambda_i[k] h_k h_k^T$ is defined to be the edge probability matrix for graph i. In cases that $\{\lambda_i\}_{i=1}^m$ are of primary interest, they are treated as parameters, and it is said $\{\mathbf{A}_i\}_{i=1}^m$ follows a m-graph d-dimensional multiple random eigen graphs model with the notation:

$$\{\mathbf{A}_i\}_{i=1}^m \sim MREG(\lambda_1, ..., \lambda_m, h_1, ..., h_d).$$

Compared to the RDPG model, MREG is designed to model multiple graphs. The vectors $\{h_k\}_{k=1}^d$ are shared across graphs; a λ_i is sampled for each graph. On a single graph, they are equivalent if the edge probability matrix is positive semidefinite. In MREG, we allow self loops to happen. This is mainly for theoretical convenience.

The left panel of Figure 2.1 shows the relationships between three random graph models defined above and the Erdos-Renyi (ER) model on 1 graph. The models considered are those conditioned on latent positions, that is τ, $\mathbf{X}$ and λ in SBM, RDPG and MREG respectively are treated as parameters; furthermore, loops are ignored in MREG. If an adjacency matrix $\mathbf{A} \sim SBM(\tau, \mathbf{B})$ and the block connectivity matrix $\mathbf{B}$ is positive semidefinite, $\mathbf{A}$ can also be written as an $RDPG(\mathbf{X})$ with $\mathbf{X}$ having at most K distinct rows. If an adjacency matrix $\mathbf{A} \sim RDPG(\mathbf{X})$, then it is also a 1-graph $MREG(\lambda_1, h_1, ..., h_d)$ with h_k being the normalized kth column of $\mathbf{X}$ and λ_1 being the vector containing the squared norms of columns of $\mathbf{X}$. However,

a 1-graph $MREG(\lambda_1, h_1, ..., h_d)$ is not necessarily an RDPG graph since λ_1 could contain negative entries which may result in an indefinite edge probability matrix.

The right panel of Figure 2.1 shows the relationships between the models on multiple graphs. For ER, SBM and RDPG, the graphs are sampled i.i.d. with the same parameters. MREG has the flexibility to have λ differ across graphs, which leads to a more generalized model for multiple graphs. Actually, it turns out that if d is allowed to be as large as $\frac{n(n+1)}{2}$, MREG can represent any distribution on binary graphs, which includes distributions in which edges are not independent.

2.2.5 Adjacency Spectral Embedding

In this section, we present the Adjacency Spectral Embedding (ASE) algorithm [4]. ASE takes the adjacency matrix and computes a low rank embedding through matrix factorization. It has been demonstrated to have good theoretical properties such as consistency and asymptotic normality [4, 26] under SBM and RDPG. Given an adjacency matrix $\mathbf{A}$ and a dimensionality parameter d, the ASE algorithm is as described in Algorithm 1.

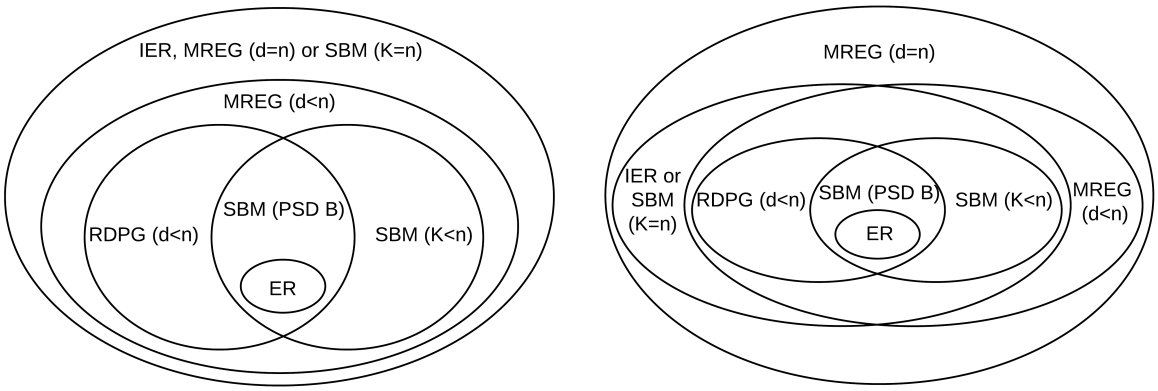

Figure 2.1: Relationship between random graph models on 1 graph and multiple graphs. The left panel shows the relationships between the random graph models on 1 graph. The models considered are those conditioned on latent positions, that is τ, $\mathbf{X}$ and λ in SBM, RDPG and MREG respectively are treated as parameters. ER is a 1-block SBM. If a graph follows SBM with a positive semidefinite edge probability matrix, it also follows the RDPG model. Any SBM and RDPG graph can be represented by a d-dimensional MREG model with d being less than or equal to the number of blocks or the dimension of RDPG. On one graph, inhomogeneous ER (IER), n-dimensional MREG and n-block SBM are equivalent. The right panel shows the relationships between the random graph models on multiple graphs. The models considered are those conditioned on latent positions, and for ER, SBM and RDPG graphs are sampled i.i.d. with the same parameters. In this case, MREG has the flexibility to have λ differ across graphs, which leads to a more generalized model for multiple graphs.

Algorithm 1 Adjacency Spectral Embedding

procedure ESTIMATE LATENT POSITION $\hat{\mathbf{X}}$

Compute spectral decomposition: $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^T$

Let $\mathbf{U}_d$ have the first d eigenvectors and the corresponding eigenvalues are stored in $\mathbf{D}_d$

Output $\hat{\mathbf{X}} = \mathbf{U}_d\mathbf{D}_d^{\frac{1}{2}}$.

end procedure

2.3 Joint Embedding

2.3.1 Joint Embedding of Graphs

The joint embedding method considers a collection of vertex-aligned graphs, and estimates a common embedding space across all graphs and a loading for each graph. Specifically, it simultaneously identifies a subspace spanned by a set of rank one symmetric matrices and projects each adjacency matrix $\mathbf{A}_i$ into the subspace. The coefficients obtained by projecting $\mathbf{A}_i$ are denoted by $\hat{\lambda}_i \in \mathbb{R}^d$, which is called the loading for graph i. To estimate rank one symmetric matrices and loadings for graphs, the algorithm minimizes the sum of squared Frobenius distances between adjacency matrices and their projections as described below.

Definition Joint Embedding of Graphs (JE). Given m graphs $\{G_i\}_{i=1}^m$ with $\mathbf{A}_i$ being the corresponding adjacency matrix, the d-dimensional joint embedding of graphs

$\{G_i\}_{i=1}^m$ is given by

$$(\hat{\lambda}_1, ..., \hat{\lambda}_m, \hat{h}_1, ..., \hat{h}_d) = \underset{\lambda_i, \|h_k\|=1}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \sum_{k=1}^{d} \lambda_i[k] h_k h_k^T\|^2. \tag{2.1}$$

Here, $\|\cdot\|$ denotes the Frobenius norm and $\lambda_i[k]$ is the kth entry of vector λ_i.

To make sure that the model is identifiable and avoid the problem scaling, h_k is required to have norm 1. In addition, $\{h_k h_k^T\}_{k=1}^d$ must be linearly independent to avoid identifiability issue in estimating λ_i; however, $\{h_k\}_{k=1}^d$ needs not to be linearly independent or orthogonal. To ease the notations, let us introduce two matrices $\mathbf{\Lambda} \in \mathbb{R}^{m \times d}$ and $\mathbf{H} \in \mathbb{R}^{n \times d}$, where λ_i is the ith row of Λ and h_k is the kth row of $\mathbf{H}$; that is, $\mathbf{\Lambda} = [\lambda_1^T, ..., \lambda_m^T]$ and $\mathbf{H} = [h_1, ..., h_d]$. The equation (2.1) can be rewritten using $\mathbf{\Lambda}$ and $\mathbf{H}$ as

$$(\hat{\mathbf{\Lambda}}, \hat{\mathbf{H}}) = \underset{\mathbf{\Lambda}, \|h_k\|=1}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \sum_{k=1}^{d} \mathbf{\Lambda}_{ik} h_k h_k^T\|^2.$$

Denote the function on the left hand side of the equation by $f(\mathbf{\Lambda}, \mathbf{H})$ which is explicitly a function of λ_is and h_ks. There are several alternative ways to formulate the problem. If vector λ_i is converted into a diagonal matrix $\mathbf{D}_i \in \mathbb{R}^{d \times d}$ by putting entries of λ_i on the diagonal of $\mathbf{D}_i$, then solving equation (2.1) is equivalent to solving

$$\begin{aligned} \underset{\mathbf{D}_i, \|h_k\|=1}{\operatorname{argmin}} \quad & \sum_{i=1}^{m} \|\mathbf{A}_i - \mathbf{H}\mathbf{D}_i\mathbf{H}^T\|^2 \\ \text{subject to} \quad & \mathbf{D}_i \text{ being diagonal.} \end{aligned}$$

Equation (2.1) can be also viewed as a tensor factorization problem. If $\{\mathbf{A}_i\}_{i=1}^m$ are stacked in a 3-D array $\mathbb{A} \in \mathbb{R}^{m \times n \times n}$, then solving equation (2.1) is also equivalent to

$$\underset{\mathbf{\Lambda}, \|h_k\|=1}{\operatorname{argmin}} \|\mathbb{A} - \sum_{k=1}^{d} \mathbf{\Lambda}_{*k} \otimes h_k \otimes h_k\|^2,$$

where $\otimes$ denotes the tensor product and $\mathbf{\Lambda}_{*k}$ is the kth column of $\mathbf{\Lambda}$. It is well known in the tensor factorization community that the solution to Equation 2.1 may not necessarily exist for $d \geq 2$. This phenomenon is first found by Bini *et al.* [27], and Silva and Lim gives a characterization all such tensors in the order-3 rank-2 case [28]. Although there may not exist a global minimum, finding the local solution in a compact region still provide significant insights to the data. We design an algorithm which is guaranteed to converge, and provide analysis under the $d = 1$ case.

The joint embedding algorithm assumes the graphs are vertex-aligned, unweighted, and undirected. The vertex-aligned graphs are common in applications such as neuroimaging [29]. In case that the graphs are not aligned, graph matching should be performed before the joint embedding [30, 31]. The mis-alignments of some vertices will have adverse effects in estimating corresponding latent positions in $\mathbf{H}$; however, a small number of mis-aligned vertices should not have a big impact in estimating $\mathbf{\Lambda}$. If the graphs have weighted edges, the joint embedding can still be applied. Also, the MREG model can be easily extended to weighted graphs by replacing the Bernoulli distribution with other proper distributions. In fact, in the experiment of section 5.3, the graphs are weighted, where the edge weights are the log of fiber counts across regions of brains. In case of directed graph, to apply the joint embedding, one

can symmetrize the graph by removing the direction of edges. Alternatively, $h_k h_k^T$ in equation (2.1) can be replaced by $h_k g_k^T$, with h_k and g_k representing the in and out latent positions respectively. With this modification, equation (2.1) becomes the tensor factorization problem [32].

The optimization problem in equation (2.1) is similar to Principal Component Analysis (PCA) in the sense of minimizing squared reconstruction error to recover loadings and components [13]. However, there are extra symmetries and rank constraints on the components. Specifically, if $h_k h_k^T$ is replaced by a matrix $\mathbf{S}_k$ in equation (2.1)

$$(\hat{\lambda}_1, ..., \hat{\lambda}_m, \hat{\mathbf{S}}_1, ..., \hat{\mathbf{S}}_d) = \underset{\lambda_i, \mathbf{S}}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \sum_{k=1}^{d} \lambda_i[k]\mathbf{S}_k\|^2,$$

the problem can be solved by applying PCA on vectorized adjacency matrices. In this case, there is a $\mathbf{S}_k$ to estimate for each latent dimension which has $\frac{n(n+1)}{2}$ parameters. Compared to PCA, the joint embedding estimates a rank one matrix $h_k h_k^T$ for each latent dimension which has n parameters, and h_k can be treated as latent positions for vertices, but the joint embedding yields a larger approximation error due to the extra constraints. Similar optimization problems have also been considered in the simultaneous diagonalization literature [33, 34]. The difference is that the joint embedding is estimating an n-by-d matrix $\mathbf{H}$ by minimizing reconstruction error instead of finding a n-by-n non-singular matrix by trying to simultaneously diagonalize all matrices. The problem in equation (2.1) has considerably fewer parameters to optimize, which makes it more stable and applicable with n being moderately large. In

case of embedding only one graph, the joint embedding is equivalent to the Adjacency Spectral Embedding solved by singular value decomposition [4]. Next, we describe an algorithm to optimize the objective function $f(\mathbf{\Lambda}, \mathbf{H})$.

2.3.2 Alternating Descent Algorithm

The joint embedding of $\{G_i\}_{i=1}^m$ is estimated by solving the optimization problem in equation (2.1). There are a few methods proposed to solve similar problems. Carroll and Chang [35] propose to use an alternating minimization method that ignores symmetry. The hope is that the algorithm will converge to a symmetric solution itself due to symmetry in data. Gradient approaches have also been considered for similar problems [36, 37]. We develop an alternating descent algorithm to minimize $f(\mathbf{\Lambda}, \mathbf{H})$ that combines ideas from both approaches [38]. The algorithm can also be understood as a block coordinate descent method with $\mathbf{\Lambda}$ and $\mathbf{H}$ being the two blocks [39, 40]. The algorithm iteratively updates one of $\mathbf{\Lambda}$ and $\mathbf{H}$ while treating the other parameter as fixed. Optimizing $\mathbf{\Lambda}$ when fixing H is straight forward, since it is essentially a least squares problem. However, optimizing $\mathbf{H}$ when fixing $\mathbf{\Lambda}$ is hard due to the fact that the problem is non-convex and there is no closed form solution available. In this case, the joint embedding algorithm utilizes gradient information and take an Armijo backtracking line search strategy to update $\mathbf{H}$ [41].

Instead of optimizing all columns $\mathbf{\Lambda}$ and $\mathbf{H}$ simultaneously, we consider a greedy algorithm which solves the optimization problem by only considering one column of

$\mathbf{\Lambda}$ and $\mathbf{H}$ at a time. Specifically, the algorithm fixes all estimates for the first $k_0 - 1$ columns of $\mathbf{\Lambda}$ and $\mathbf{H}$ at iteration k_0, and then the objective function is minimized by searching through only the k_0th column of $\mathbf{\Lambda}$ and $\mathbf{H}$. That is,

$$(\hat{\mathbf{\Lambda}}_{*k_0}, \hat{h}_{k_0}) = \underset{\mathbf{\Lambda}_{*k_0}, \|h_{k_0}\|=1}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \sum_{k=1}^{k_0-1} \hat{\mathbf{\Lambda}}_{ik} \hat{h}_k \hat{h}_k^T - \mathbf{\Lambda}_{ik_0} h_{k_0} h_{k_0}^T\|^2. \tag{2.2}$$

Let $f(\mathbf{\Lambda}_{*k_0}, h_{k_0})$ denote the sum on the left hand side of the equation. To compute a d-dimensional joint embedding $(\hat{\mathbf{\Lambda}}, \hat{\mathbf{H}})$, the algorithm iteratively solves the one dimensional optimization problem above by letting k_0 vary from 1 to d.

There are a few advantages in iteratively solving one dimensional problems. First, there are fewer parameters to fit at each iteration, since the algorithm are only allowed to vary $\mathbf{\Lambda}_{*k_0}$ and h_{k_0} at iteration k_0. This makes initialization and optimization steps much easier compared to optimizing all columns of $\mathbf{H}$ simultaneously. Second, it implicitly enforces an ordering on the columns of $\mathbf{H}$. This ordering allows us to select the top few columns of $\mathbf{\Lambda}$ and $\mathbf{H}$ in cases where model selection is needed after the joint embedding. Third, it allows incremental computation. If d and d' dimensional joint embeddings are both computed, the first $\min(d, d')$ columns of $\hat{\mathbf{\Lambda}}$ and $\hat{\mathbf{H}}$ will be the same. Fourth, the solution is guaranteed to exist when solving iteratively [28]. Finally, based on numerical experiments, the difference between optimizing iteratively and optimizing all the parameters when d is small is negligible; however, the iterative algorithm yields a slightly smaller objective function when d is large.

The disadvantage of optimizing each column separately is that the algorithm is more likely to end up at a local minimum when the objective function is structured not in favor of embedding iteratively. In practice, this problem can be mitigated by running the joint embedding algorithm several times with random initializations.

To find $\mathbf{\Lambda}_{*k_0}$ and h_{k_0} in equation (2.2), the algorithm needs to evaluate two derivatives: $\frac{\partial f}{\partial h_{k_0}}$ and $\frac{\partial f}{\partial \mathbf{\Lambda}_{ik_0}}$. Denote by $\mathbf{R}_{ik_0}$ the residual matrix after iteration $k_0 - 1$ which is $\mathbf{A}_i - \sum_{k=1}^{k_0-1} \hat{\mathbf{\Lambda}}_{ik} \hat{h}_k \hat{h}_k^T$. The gradient of the objective function with respect to h_{k_0} is given by

$$\frac{\partial f}{\partial h_{k_0}} = -4 \sum_{i=1}^{m} \mathbf{\Lambda}_{ik_0} (\mathbf{R}_{ik} - \mathbf{\Lambda}_{ik_0} h_{k_0} h_{k_0}^T) h_{k_0}. \tag{2.3}$$

The derivative of the objective function with respect to $\mathbf{\Lambda}_{ik_0}$ is given by

$$\frac{\partial f}{\partial \mathbf{\Lambda}_{ik_0}} = -2 \langle \mathbf{R}_{ik} - \mathbf{\Lambda}_{ik_0} h_{k_0} h_{k_0}^T, h_{k_0} h_{k_0}^T \rangle.$$

Setting the derivative to 0 yields

$$\hat{\mathbf{\Lambda}}_{ik_0} = \langle \mathbf{R}_{ik}, h_{k_0} h_{k_0}^T \rangle, \tag{2.4}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product.

The joint embedding algorithm alternates between updating $\hat{\mathbf{\Lambda}}_{ik_0}$ and $\hat{h}_{k_0}$ according to equation (2.3) and (2.4). Algorithm 2 describes the general procedure to compute the d-dimensional joint embedding of graphs $\{G_i\}_{i=1}^m$. The algorithm outputs two matrices: $\hat{\mathbf{\Lambda}}$ and $\hat{\mathbf{H}}$. The rows of $\hat{\mathbf{\Lambda}}$ denoted by $\{\hat{\lambda}_i\}_{i=1}^m$ can be treated as estimates of $\{\lambda_i\}_{i=1}^m$ in MREG and features for graphs. Columns of $\hat{\mathbf{H}}$ denoted by

$\{\hat{h}_k\}_{k=1}^d$ are estimates of $\{h_k\}_{k=1}^d$. If a new graph G is observed with adjacency matrix $\mathbf{A}$, $\mathbf{A}$ can be projected into the linear space spanned by $\{\hat{h}_k\hat{h}_k^T\}_{k=1}^d$ to obtain features for the graph.

In case of $\mathbf{A}_i$ being large, the updating equations (2.3) and (2.4) are not practical due to $h_k h_k^T$ and $\mathbf{R}_{ik}$ being large and dense. However, they can be rearranged to avoid explicit computation of $h_k h_k^T$ and $\mathbf{R}_{ik}$. The equation (2.3) becomes

$$\begin{aligned}
\frac{\partial f}{\partial h_{k_0}} &= -4\sum_{i=1}^{m} \mathbf{\Lambda}_{ik_0}(\mathbf{R}_{ik} - \mathbf{\Lambda}_{ik_0} h_{k_0} h_{k_0}^T) h_{k_0} \\
&= -4\sum_{i=1}^{m} \mathbf{\Lambda}_{ik_0}\mathbf{R}_{ik} h_{k_0} + 4\sum_{i=1}^{m} \mathbf{\Lambda}_{ik_0}^2 h_{k_0} \\
&= -4\sum_{i=1}^{m} \mathbf{\Lambda}_{ik_0}(\mathbf{A}_i - \sum_{k=1}^{k_0-1} \mathbf{\Lambda}_{ik} h_k h_k^T) h_{k_0} + 4\sum_{i=1}^{m} \Lambda_{ik_0}^2 h_{k_0} \\
&= -4\sum_{i=1}^{m} \mathbf{\Lambda}_{ik_0}\mathbf{A}_i h_{k_0} + 4\sum_{i=1}^{m} \mathbf{\Lambda}_{ik_0} \sum_{k=1}^{k_0-1} \mathbf{\Lambda}_{ik}(h_k^T h_{k_0}) h_k + 4\sum_{i=1}^{m} \mathbf{\Lambda}_{ik_0}^2 h_{k_0}.
\end{aligned}$$

Similarly, the equation (2.3) can be rewritten as

$$\begin{aligned}
\hat{\mathbf{\Lambda}}_{ik_0} &= \langle \mathbf{R}_{ik}, h_{k_0} h_{k_0}^T \rangle \\
&= h_{k_0}^T \mathbf{R}_{ik} h_{k_0} \\
&= h_{k_0}^T (\mathbf{A}_i - \sum_{k=1}^{k_0-1} \mathbf{\Lambda}_{ik} h_k h_k^T) h_{k_0} \\
&= h_{k_0}^T \mathbf{A}_i h_{k_0} - \sum_{k=1}^{k_0-1} \mathbf{\Lambda}_{ik}(h_{k_0}^T h_k)^2.
\end{aligned}$$

Based on the rearranged equations, efficiently evaluating matrix vector product $\mathbf{A}_i h_{k_0}$ is needed to calculate the derivatives. This can be completed for a variety of matrices, in particular, sparse matrices [42].

The Algorithm 2 is guaranteed to converge to a stationary point. Specifically, at the termination of iteration k_0, $\frac{\partial f}{\partial h_{k_0}} \approx 0$ and $\frac{\partial f}{\partial \mathbf{\Lambda}_{ik_0}} \approx 0$. First, $\frac{\partial f}{\partial \mathbf{\Lambda}_{ik_0}} \approx 0$ is ensured due to exact updating by equation (2.4). Second notice that updating according to equation (2.3) and (2.4) always decreases the objective function. Due to the fact that h_{k_0} lies on the unit sphere and the objective is twice continuous differentiable, $\frac{\partial f}{\partial h_{k_0}}$ is Lipschitz continuous. This along with Armijo backtracking line search guarantees a "sufficient" decrease $c\|\frac{\partial f}{\partial h_{k_0}}\|^2$ in objective function each time when the algorithm updates h_{k_0} [41], where c is a constant independent of h_{k_0}. Since the objective function is bounded below by 0, this implies convergence of gradient, that is $\frac{\partial f}{\partial h_{k_0}} \to 0$.

In general, the objective function may have multiple stationary points due to non-convexity. Therefore, the joint embedding algorithm is sensitive to initializations. Actually, like many of the problems in tensor factorization, finding the global minimum in joint embedding is NP-Hard [43]. When time permits, we recommend running the joint embedding several times with random initializations. In Section 5.1, we study the effects of different initialization approaches through a numerical simulation experiment. For other simulation and real experiments, we initialize $\hat{\mathbf{\Lambda}}_{ik_0}$ and $\hat{h}_{k_0}$ through SVD of the average residual matrix $\sum \mathbf{R}_{ik_0}/m$. The optimization algorithm described above may not be the fastest approach to solving the problem; however, numerical optimization is not the focus of this thesis. Based on results from numerical applications, our approach works well in estimating parameters and extracting features for subsequent statistical inference. Next, we discuss some variations of the

joint embedding algorithm.

Algorithm 2 Joint Embedding

procedure FIND JOINT EMBEDDING $\hat{\mathbf{\Lambda}}, \hat{\mathbf{H}}$ OF $\{\mathbf{A}_i\}_{i=1}^m$
  Set residuals: $\mathbf{R}_{i1} = \mathbf{A}_i$
  for $k = 1 : d$ **do**
    Initialize h_k and $\mathbf{\Lambda}_{*k}$
    while not convergent **do**
      Fixing $\mathbf{\Lambda}_{*k}$, update h_k by gradient descent (2.3)
      Project h_k back to the unit sphere
      Fixing h_k, update $\mathbf{\Lambda}_{*k}$ by (2.4)
      Compute objective $\sum_{i=1}^{m} \|\mathbf{R}_{ik} - \mathbf{\Lambda}_{ik} h_k h_k^T\|^2$
    end while
    Update residuals: $\mathbf{R}_{i(k+1)} = \mathbf{R}_{ik} - \mathbf{\Lambda}_{ik} h_k h_k^T$
  end for
  Output $\hat{\mathbf{\Lambda}} = [\mathbf{\Lambda}_{*1}, ..., \mathbf{\Lambda}_{*d}]$ and $\hat{\mathbf{H}} = [h_1, ..., h_d]$
end procedure

2.3.3 Variations

The joint embedding algorithm described in the previous section can be modified to accommodate several different settings.

Variation 1. When all graphs come from the same distribution, we can force

estimated loadings $\hat{\lambda}_i$ to be equal across all graphs. This is useful when the primary inference task is to extract features for vertices. Since all graphs share the same loadings, with slightly abusing notations, let $\mathbf{\Lambda}$ be a vector in $\mathbb{R}^d$ and the optimization problem becomes

$$(\hat{\mathbf{\Lambda}}, \hat{\mathbf{H}}) = \underset{\mathbf{\Lambda}, \|h_k\|=1}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \sum_{k=1}^{d} \mathbf{\Lambda}_k h_k h_k^T\|^2,$$

which is equivalent to

$$(\hat{\mathbf{\Lambda}}, \hat{\mathbf{H}}) = \underset{\mathbf{\Lambda}, \|h_k\|=1}{\operatorname{argmin}} \|\frac{1}{m} \sum_{i=1}^{m} \mathbf{A}_i - \sum_{k=1}^{d} \mathbf{\Lambda}_k h_k h_k^T\|^2.$$

Therefore, the optimization problem can be solved exactly by finding the singular value decomposition of the average adjacency matrix $\frac{1}{m} \sum_{i=1}^{m} \mathbf{A}_i$.

Variation 2. When there is a discrete label $y_i \in \mathbb{Y}$ associated with the graph G_i available, we may require all loadings $\hat{\lambda}_i$ to be equal within class. Let $\mathbf{\Lambda} \in \mathbb{R}^{|\mathbb{Y}| \times d}$, the optimization problem becomes

$$(\hat{\mathbf{\Lambda}}, \hat{\mathbf{H}}) = \underset{\mathbf{\Lambda}, \|h_k\|=1}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \sum_{k=1}^{d} \mathbf{\Lambda}_{y_i k} h_k h_k^T\|^2.$$

In this case, when updating $\mathbf{\Lambda}$ as in equation (2.4), the algorithm should average $\mathbf{\Lambda}_{yk}$ within the same class, that is

$$\hat{\mathbf{\Lambda}}_{yk} = \frac{\sum_{i=1}^{m} \mathbb{I}\{y_i = y\} \langle \mathbf{R}_{ik}, h_{k_0} h_{k_0}^T \rangle}{\sum_{i=1}^{m} \mathbb{I}\{y_i = y\}}.$$

Variation 3. In some applications, we may require all $\mathbf{\Lambda}_{ik}$ to be greater than 0, as

in non-negative matrix factorization. One advantage of this constraint is that graph G_i may be automatically clustered based on the largest entry of $\hat{\lambda}_i$. In this case, the optimization problem is

$$(\hat{\mathbf{\Lambda}}, \hat{\mathbf{H}}) = \underset{\mathbf{\Lambda} \geq 0, \|h_k\|=1}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \sum_{k=1}^{d} \mathbf{\Lambda}_{ik} h_k h_k^T\|^2.$$

To guarantee nonnegativity, the algorithm should use nonnegative least squares in updating $\mathbf{\Lambda}$ [44]. Furthermore, a constraint on the number of non-zero elements in ith row of $\mathbf{\Lambda}$ can be added as in K-SVD [45], and a basis pursuit algorithm could be used to update $\mathbf{\Lambda}$ [46, 47]. Next, we discuss some theoretical properties of the MREG model and joint embedding when treated as a parameter estimation procedure for the model.

2.4 Theoretical Results

2.4.1 Multiple Random Eigen Graphs

We first show that MREG is a very general model on graphs. In fact, it can represent any distribution on graphs as Theorem 2.4.1 implies.

Theorem 2.4.1 *Given any distribution $\mathcal{F}$ on graphs and a random adjacency matrix $\mathbf{A} \sim \mathcal{F}$, there exists a dimension d, a distribution F on $\mathbb{R}^d$, and a set of vectors $\{h_k\}_{k=1}^d$, such that $\mathbf{A} \sim MREG(F, h_1, ..., h_d)$.*

Theorem 2.4.1 suggests that MREG is really a semi-parametric model, which can capture any distribution on graphs. One can model any set of graphs by MREG with the guarantee that the true distribution is in the model with d being large enough. However, in practice, a smaller d may lead to better inference performance due to reduction in the dimensionality. In the next section, we consider the joint embedding algorithm which can be understood as a parameter estimation procedure for MREG.

2.4.2 Joint Embedding Estimator

In this section, we consider a simple setting where graphs follow a 1-dimensional MREG model, that is $\{(\lambda_i, \mathbf{A}_i)\}_{i=1}^m \sim MREG(F, h_1)$. The 1-dimensional joint embedding is well defined in this case, that is $\hat{\lambda}_i$ and $\hat{h}_1$ defined in Equation 2.1 is guaranteed to exist. Under this MREG model, the joint embedding of graphs can be understood as estimators for parameters of the model. Specifically, $\hat{\lambda}_i$ and $\hat{h}_1$ are estimates of λ_i and h. We prove two theorems concerning the asymptotic behavior of estimator $\hat{h}_1$ produced by joint embedding.

Let $\hat{h}_1^m$ denote the estimates based on m graphs and define functions ρ, D_m and D as below:

$$\rho(\mathbf{A}_i, h) = \|\mathbf{A}_i - \langle \mathbf{A}_i, hh^T \rangle hh^T\|^2,$$

$$D_m(h, h_1) = \frac{1}{m} \sum_{i=1}^{m} \rho(\mathbf{A}_i, h),$$

$$D(h, h_1) = \mathbb{E}(\rho(\mathbf{A}_i, h)).$$

One can understand D_m and D as sample and population approximation errors respectively. By equation (2.1),

$$\hat{h}_1^m = \underset{\|h\|=1}{\operatorname{argmin}} \underset{\lambda_i}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \lambda_i h h^T\|.$$

By equation (2.4),

$$\langle \mathbf{A}_i, hh^T \rangle = \underset{\lambda_i}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \lambda_i h h^T\|.$$

Therefore,

$$\hat{h}_1^m = \underset{\|h\|=1}{\operatorname{argmin}} D_m(h, h_1).$$

The first theorem states that $\hat{h}_1^m$ converges almost surely to the global minimum of $D(h, h_1)$, given that the global minimum is unique. Alternatively, the theorem implies the sample minimizer converges to the population minimizer.

Theorem 2.4.2 *If $D(h, h_1)$ has a unique global minimum at h', then $\hat{h}_1^m$ converges almost surely to h' as m goes to infinity. That is,*

$$\hat{h}_1^m \xrightarrow{a.s.} h'.$$

Theorem 2.4.2 requires h' to be the unique global minimizer of $D(h, h_1)$. However, the global minimizer is definitely not unique due to the symmetry up to sign flip of h, that is $D(h, h_1) = D(-h, h_1)$ for any h. This problem can be addressed by forcing an orientation of $\hat{h}_1^m$ or stating that the convergence is up to a sign flip. It is also possible that there are multiple global minimizers of $D(h, h_1)$ which are not sign flips of each other. In this case, Theorem 2.4.2 does not apply. We are currently only certain that

when all graphs are from the Erdos-Renyi random graph model, the global minimizer is unique up to a sign flip. The next theorem concerns the asymptotic bias of h'. It gives a bound on the difference between the population minimizer h' and the truth h_1.

Theorem 2.4.3 *If h' is a minimizer of $D(h, h_1)$, then*

$$\|h' - h_1\| \leq \frac{2\mathbb{E}(\lambda_i)}{\mathbb{E}(\lambda_i^2)(h_1^T h')^2}.$$

To see an application of Theorem 2.4.3, let us consider the case in which all graphs are Erdos-Renyi graphs with 100 vertices and edge probability of 0.5. Under this setting, Theorem 2.4.3 implies $\|h' - h_1\| \in [0, 0.04] \cup [1.28, 1.52]$. The second interval is disturbing. It is due to the fact that when $h_1^T h'$ is small, the bound is useless. We provide some insights as to why the second interval is there and how we can get rid of it with additional assumptions. In the proof of Theorem 2.4.3, we show that the global optimizer h' satisfies

$$h' = \underset{\|h\|=1}{\operatorname{argmax}}\ \mathbb{E}(\langle \mathbf{A}_i, hh^T \rangle^2).$$

Taking a closer look at $E(\langle \mathbf{A}_i, hh^T \rangle^2)$,

$$\begin{aligned}\mathbb{E}(\langle \mathbf{A}_i, hh^T \rangle^2) &= \mathbb{E}(\langle \mathbf{P}_i, hh^T \rangle^2) + \mathbb{E}(\langle \mathbf{A}_i - \mathbf{P}_i, hh^T \rangle^2) \\ &= \mathbb{E}(\lambda_i^2)(h_1^T h)^4 + \mathbb{E}((h^T(\mathbf{A}_i - \mathbf{P}_i)h)^2).\end{aligned}$$

Therefore,

$$h' = \underset{\|h\|=1}{\operatorname{argmax}}\ \mathbb{E}(\lambda_i^2)(h_1^T h)^4 + \mathbb{E}((h^T(\mathbf{A}_i - \mathbf{P}_i)h)^2).$$

We can see that $\mathbb{E}(\lambda_i^2)(h_1^T h)^4$ is maximized when $h = h_1$; however, the noise term $\mathbb{E}((h^T(\mathbf{A}_i - \mathbf{P}_i)h)^2)$ is generally not maximized at $h = h_1$. If n is large, we can apply a concentration inequality to $(h^T(\mathbf{A}_i - \mathbf{P}_i)h)^2$ and have an upper bound on $\mathbb{E}((h^T(\mathbf{A}_i - \mathbf{P}_i)h)^2)$. If we further assume A_i is not too sparse, that is $\mathbb{E}(\lambda_i^2)$ grows with n fast enough, then the sum of these two terms is dominated by the first term. This provides a way to have a lower bound on $h_1^T h'$. We may then replace the denominator of the bound in Theorem 2.4.3 by the lower bound. In general, if n is small, the noise term may cause h' to differ from h_1 by a significant amount. In this chapter, we focus on the case that n is fixed. The case that n goes to infinity for Random Dot Product Graph is considered in [26].

The two theorems above concern only the estimation of h_1, but not λ_i. Based on equation (2.4), the joint embedding estimates λ_i by

$$\hat{\lambda}_i^m = \langle \mathbf{A}_i, \hat{h}_1^m \hat{h}_1^{mT} \rangle.$$

When m goes to infinity, we can apply Theorem 2.4.2,

$$\hat{\lambda}_i^m = \langle \mathbf{A}_i, \hat{h}_1^m \hat{h}_1^{mT} \rangle \xrightarrow{a.s.} \langle \mathbf{A}_i, h' h'^T \rangle = h'^T \mathbf{A}_i h'.$$

Then, applying the bound on $\|h' - h_1\|$ derived in Theorem 2.4.3 and utilizing the fact that $h^T \mathbf{A}_i h$ is continuous in h, we can obtain an upper bound on $|\hat{\lambda}_i^m - h_1^T \mathbf{A}_i h_1|$. When $\mathbf{A}_i$ is large, $h_1^T \mathbf{A}_i h_1$ is concentrated around λ_i with high probability. As a consequence, with high probability $|\hat{\lambda}_i^m - \lambda_1|$ is small. In the next section, we demonstrate properties and utilities of the joint embedding algorithm through experiments.

2.5 Numerical Results

Before going into details of experiments, we want to discuss how to select the dimensionality d of the joint embedding. Estimating d is an important model selection question that has been studied for years under various settings [48]. Model selection is not the focus of this thesis, but we still face this problem in numerical experiments. In the simulation experiments of this section, we assume d is known to us and simply set the dimensionality estimate $\hat{d}$ equal to d. In the real data experiment, we recommend two approaches to determine $\hat{d}$. Both approaches require first running the d'-dimensional joint embedding algorithm, where d' is sufficiently large. We then plot the objective function versus dimension, and determine $\hat{d}$ to be where the objective starts to flatten out. Alternatively, we can plot $\{\hat{\mathbf{\Lambda}}_{ik}\}_{i=1}^{m}$ for $k = 1, ..., d'$, and select $\hat{d}$ when the loadings start to look like noise with 0 mean. These two approaches should yield a similar dimensionality estimate of $\hat{d}$.

2.5.1 Simulation: Joint Embedding Under a Simple Model

In the first experiment, we present a simple numerical example to demonstrate some properties of the joint embedding procedure as the number of graphs grows. We repeatedly generate graphs with 20 vertices from 3-dimensional MREG, where

$\lambda_i[1] \sim \text{Uniform}(8, 16)$, $\lambda_i[2] \sim \text{Uniform}(0, 2)$ and $\lambda_i[3] \sim \text{Uniform}(0, 1)$, with

$$h_1 = [1, 1, 1, ..., 1]/\sqrt{20}$$

$$h_2 = [1, -1, 1, -1, 1, -1, ..., -1]/\sqrt{20}$$

$$h_3 = [1, 1, -1, -1, 1, 1, -1, -1, ..., -1]/\sqrt{20}.$$

We keep doubling the number of graphs m from 2^4 to 2^{12}. At each value of m, we compute the 3-dimensional joint embedding of graphs. Let the estimated parameters based on m graphs be denoted by $\hat{\lambda}_i^m$ and $\hat{h}_k^m$. Two quantities based on $\hat{h}_k^m$ are calculated. The first is the norm difference between the current h_k estimates and the previous estimates, namely $\|\hat{h}_k^m - \hat{h}_k^{m/2}\|$. This provides numerical evidence for the convergence of our principled estimation procedure. The second quantity is $\|\hat{h}_k^m - h_k\|$. This investigates whether $\hat{h}_k$ is an unbiased estimator for h_k. The procedure described above is repeated 20 times. Figure 2.2 presents the result. Based on the plot, the norm of differences $\|\hat{h}_k^m - \hat{h}_k^{m/2}\|$ seem to converge to 0 as m increases. This suggests the convergence of $\hat{h}_1^m$. Second, we notice that the bias $\|\hat{h}_2^m - h_2\|$ and $\|\hat{h}_3^m - h_3\|$ do not converge to 0; instead, it stops decreasing at around 0.1 and 0.2 respectively. This suggests that $\hat{h}_k$ is an asymptotically biased estimator for h_k. Actually, this is as to be expected: when there are infinitely many nuisance parameters present, Neyman and Scott demonstrate that maximum likelihood estimator is inconsistent [49]. In our case, there are infinitely many λ_i as m grows; therefore, we do not expect the joint embedding to provide an asymptotic consistent estimate of h_k.

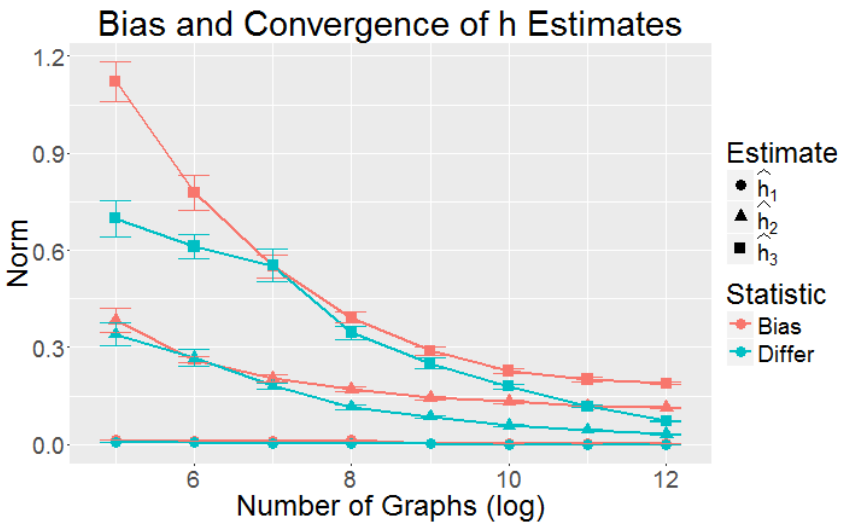

Figure 2.2: Mean bias ($\|\hat{h}_k^m - h_k\|$) and mean difference between estimates ($\|\hat{h}_k^m - \hat{h}_k^{m/2}\|$) across 20 simulations are shown. The standard errors are also given by error bars. The graphs are generated from a 3-dimensional MREG model as described in section 5.1. $\hat{h}_k^m$ has small asymptotic bias; however, it seems to converge as m increases.

In applications such as clustering or classifying multiple graphs, we may be not interested in $\hat{h}_k$. $\hat{\lambda}_i$ is of primary interest, which provides information specifically about the graphs G_i. Here, we consider two approaches to estimate $\lambda_i[1]$. The first approach is estimating $\lambda_i[1]$ through joint embedding, that is

$$\hat{\lambda}_i[1] = \langle \mathbf{A}_i, \hat{h}_1^m \hat{h}_1^{mT} \rangle.$$

The second approach estimates λ_i by assuming h_1 is known. In this case, equation (2.4) gives

$$\hat{\lambda}_i[1] = \langle \mathbf{A}_i, h_1 h_1^T \rangle.$$

Initialization	Objective	Running time (sec)
SVD	375.22(1.21)	8.3(1.0)
1 Random	383.29(1.60)	96.5(5.3)
Best of 10 Random	379.63(1.39)	9.2(1.4)
Truth	374.69(1.22)	7.8(1.0)

Table 2.1: Objective function and running time of four initialization approaches. SVD and truth initializations are significantly better than random initializations.

$\hat{\lambda}_i[1]$ calculated this way can be thought as the "oracle" estimate. Figure 2.3 shows the differences in estimates provided by two approaches. Not surprisingly, the differences are small due to the fact that $\hat{h}_1^m$ and h_1 are close.

Next, we investigate the effects of four different initialization approaches. The first approach utilizes SVD on average residual matrix to initialize h_k at each iteration. The second approach randomly samples independent Gaussian variable for each entry of h_k. The third approach takes the best initialization among 10 random initializations. The fourth approach initializes h_k using the truth. To compare these approaches, we generate 16 graphs from the MREG model and jointly embed them with four different initializations. Then, another 16 graphs are generated and the objective function on these graphs are evaluated using $\hat{\mathbf{H}}$ estimated by joint embedding. This procedure is repeated 100 times. Mean objective function and total running time with standard error of these four approaches are shown in Table 2.1. Based on

Wilcoxon signed-rank test, SVD and truth initializations are significantly better than random initializations. For the rest experiments, the initialization is completed by SVD.

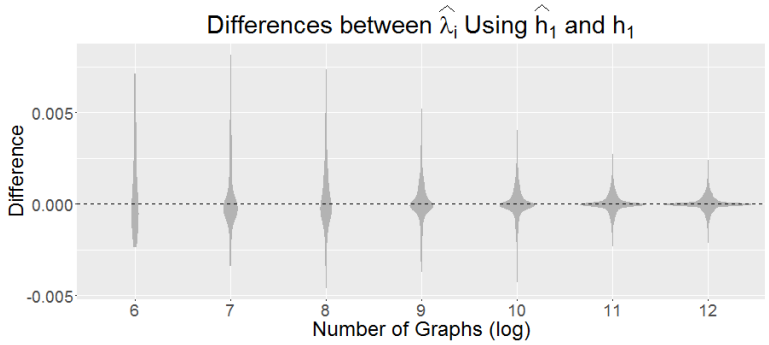

Figure 2.3: Distribution of differences between $\hat{\lambda}_i[1]$ estimated using $\hat{h}_1^m$ and h_1. The graphs are generated from the 3-dimensional MREG model as described in section 5.1. The differences are small due to the fact that $\hat{h}_1^m$ and h_1 are close.

2.5.2 Simulation: Classify Graphs

In this experiment, we consider the inference task of classifying graphs. We have m pairs $\{(\mathbf{A}_i, y_i)\}_{i=1}^m$ of observations. Each pair consists of an adjacency matrix $\mathbf{A}_i \in \{0,1\}^{n\times n}$ and a label $y_i \in [K]$. Furthermore, all pairs are assumed to be independent and identically distributed according to an unknown distribution $\mathbb{F}_{\mathbf{A},y}$, that is

$$(\mathbf{A}_1, y_1), (\mathbf{A}_2, y_2), ..., (\mathbf{A}_m, y_m) \overset{i.i.d.}{\sim} \mathbb{F}_{A,y}.$$

The goal is to find a classifier g which is a function $g : \{0,1\}^{n\times n} \to [K]$ that has a small classification error $L_g = \mathbb{P}(g(\mathbf{A}) \neq y)$.

We consider a binary classification problem where y takes value 1 or 2. 200 graphs with 100 vertices are independently generated. The graphs are sampled from a 2-dimensional MREG model. Let h_1 and h_2 be two vectors in $\mathbb{R}^{100}$, and

$$h_1 = [0.1, ..., 0.1]^T \text{ , and } h_2 = [-0.1, ..., -0.1, 0.1, ..., 0.1]^T.$$

Here, h_2 has -0.1 as its first 50 entries and 0.1 as its last 50 entries. Graphs are generated according to the MREG model,

$$\{(\lambda_i, \mathbf{A}_i)\}_{i=1}^{200} \sim MREG(F, h_1, h_2), \tag{2.5}$$

where F is a mixture of two point masses with equal probability,

$$F = \frac{1}{2}\mathbb{I}\{\lambda = [25, 5]\} + \frac{1}{2}\mathbb{I}\{\lambda = [22.5, 2.5]\}.$$

We let the class label y_i indicate which point mass λ_i is sampled from, that is $y_i = 1$ if $\lambda_i = [25, 5]$ and $y_i = 2$ if $\lambda_i = [22.5, 2.5]$. In terms of SBM, this graph generation scheme is equivalent to

$$A_i|y_i = 1 \sim SBM((1, ..., 1, 2, ..., 2), \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix})$$

$$A_i|y_i = 2 \sim SBM((1, ..., 1, 2, ..., 2), \begin{bmatrix} 0.25 & 0.2 \\ 0.2 & 0.25 \end{bmatrix}).$$

To classify graphs, we first jointly embed 200 graphs into 2 dimensions. The loadings are shown in Figure 2.4. We can see two classes are separated after being jointly embedded. Then, a 1-nearest neighbor classifier (1-NN) [50] is constructed based on loadings $\{\hat{\lambda}_i\}_{i=1}^{m}$.

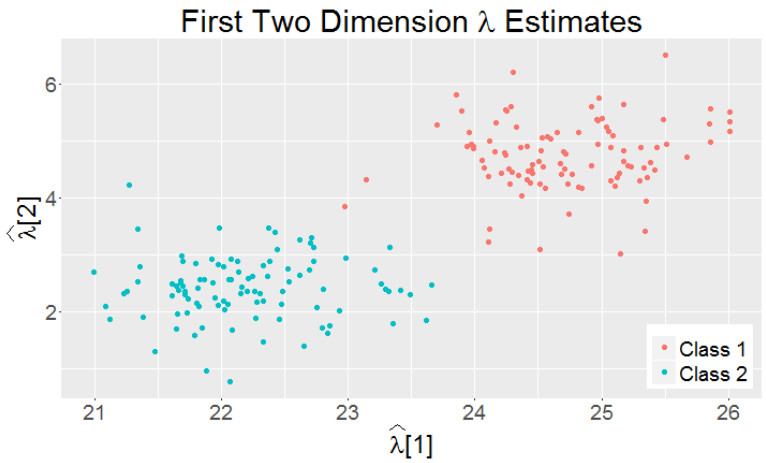

Figure 2.4: Scatter plot of loadings computed by jointly embedding 200 graphs. The graphs are generated from the 2-dimensional MREG model as described in equation (2.5). The loadings of two classes are separated after being jointly embedded.

We compare classification performances of using the joint embedding to extract features to five other feature extraction approaches discussed at the beginning of this chapter: Adjacency Spectral Embedding, Laplacian Eigenmap, Graph Statistics, Graph Spectral Statistics, and PCA. For Adjacency Spectral Embedding (ASE), each adjacency matrix is augmented with mean edge probability on the diagonal and embed into 2 dimensions. Then, pairwise procrustes distance between are computed based on embeddings. For Laplacian Eigenmap (LE), we first embed each normalized

Laplacian matrix and then compute the procrustes distance. For Graph Statistics (GS), we compute topological statistics of graphs considered in [15], which include size, maximum degree, maximum average degree, scan statistic, number of triangles, clustering coefficient and average path length. These features are then normalized to have mean 0 variance 1. For Graph Spectral Statistics (GSS), we compute the eigenvalues of adjacency matrices and treat them as features [51]. For PCA, we vectorize the adjacency matrices and compute the first two principal components through SVD. After the feature extraction step, we apply a 1-Nearest Neighbor rule to classify graphs. We let the number of graphs m increase from 4 to 200. For each value of m, we repeat the simulation 100 times. Figure 2.5 shows the result. ASE, LE, GS and GSS do not take advantage of increasing sample size in the feature extraction step. PCA has poor performance when the sample sizes is small. Joint embedding could take advantage of increasing sample size and outperforms other approaches when given more than 10 graphs.

2.5.3 Real Data: Predict Composite Creativity Index

In this experiment, we study predicting individual composite creativity index (CCI) through brain connectomes obtained by Multimodal Magnetic Resonance Imaging [52]. Neuroimaging and creativity have been jointly investigated previously. Most

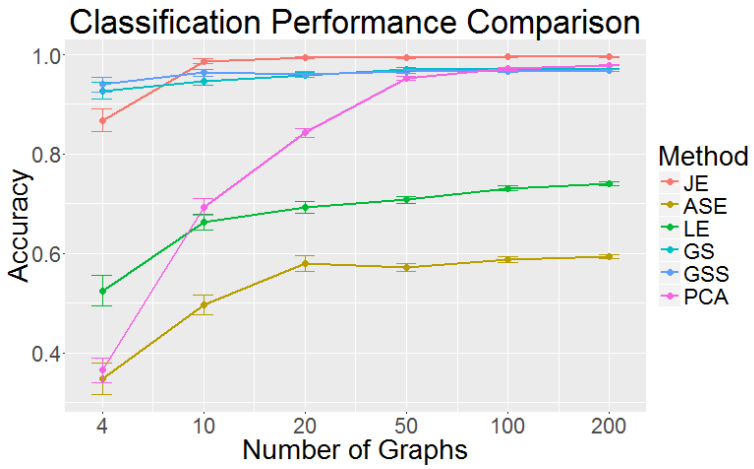

Figure 2.5: Mean classification accuracy of joint embedding, Adjacency Spectral Embedding, Laplacian Eigenmap, Graph Statistics, Graph Spectral Statistics, and PCA with their standard errors are shown. The graphs are generated from a 2-dimensional MREG model as described in the equation (2.5). The features are first extracted using methods described above; subsequently, we apply a 1-NN to classify graphs. For each value of m, the simulation is repeated 100 times. ASE, LE, GS and GSS do not take advantage of increasing sample size in the feature extraction step. PCA has poor performance when the sample sizes is small. Joint embedding takes advantage of increasing sample size and outperforms other approaches when given more than 10 graphs.

studies utilize a statistical testing method and find CCI significantly related or inversely related to the activity of some regions of the brain. For a review, please see Arden *et al.* [53]. We embrace a different approach by directly building a prediction model for CCI. First, we jointly embed brain graphs of all subjects. Then, we con-

struct a linear regression model by treating the estimated loadings as explanatory variables and CCI as the response variable.

In total, 113 healthy, young adult subjects were scanned using a Siemens TrioTim scanner. 3D-MPRAGE and DTI in 35 directions of the subjects were acquired [54]. The images were then registered by Desikan-Killiany Atlas [55], and a graph of 70 vertices is constructed. The process of transforming MRI to graphs was completed by NeuroData's MRI Graphs pipeline [56]. The graphs derived have weighted edges. One example of a graph is shown in Figure 2.6. For each subject, a divergent thinking measure was scored by independent judges using the Consensual Assessment Technique [57], from which the CCI is derived.

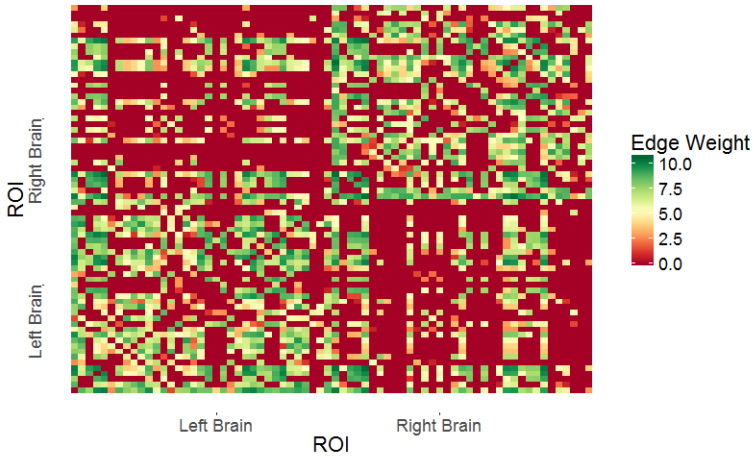

Figure 2.6: The plot shows the adjacency matrix of brain graph derived from a typical subject. There is much more neural connectivity within each hemisphere.

To predict the CCI, we first jointly embed 113 graphs with $d = 10$, and then fit a

linear model by regressing CCI on $\hat{\lambda}_i$, that is

$$CCI_i \sim \beta_0 + \hat{\lambda}_i^T \beta + \epsilon_i.$$

We consider two linear regression models. One using only $\hat{\lambda}_i[1]$ as the explanatory variable, and another one using $\hat{\lambda}_i$ as the explanatory variables. If only the first dimensional loadings are used, the top panel of Figure 2.8 shows the result. There is a significant positive linear relationship between CCI and the first dimensional loadings. The first dimensional loadings generally capture the overall connectivity of graphs. In this case, the correlation between the first dimensional loadings and the sum of edge weights is around 0.98. This model implies that the individual tends to be more creative when there is more brain connectivity. The R-square of this model is 0.07248, and the model is statistically significantly better when compared to the null model with a p-value 0.0039, according to the F-test. This model suggests that individual is more creative if the brain is more connected.

If CCI is regressed on the 10 dimensional loadings, a summary of the linear model is provided below and a scatter plot of fitted CCI versus true CCI is provided in the bottom panel of Figure 2.8.

```
> model<-lm(cci~Lambda+1)
> summary(model)
Coefficients: Estimate   Pr(>|t|)
(Intercept)   1.275e+02    0.000275 ***
```

```
Lambda1      2.421e-04    0.997981
Lambda2     -2.326e-01    0.070110  .
Lambda3     -3.716e-02    0.822592
Lambda4      8.049e-02    0.687628
Lambda5     -2.925e-01    0.421858
Lambda6     -4.285e-01    0.009088 **
Lambda7     -1.745e-01    0.590533
Lambda8     -3.465e-01    0.240093
Lambda9     -8.970e-01    0.007999 **
Lambda10    -8.955e-01    0.052839  .
Residual standard error: 9.437
on 102 degrees of freedom
Multiple R-squared:  0.2325,
Adjusted R-squared:  0.1572
F-statistic:  3.09 on 10 and 102 DF,
p-value: 0.001795
```

The R-square is 0.2325 and the model is statistically significantly better than the null model with a p-value 0.0018 according to the F-test. It is also significantly better than the model with only $\hat{\lambda}_i[1]$. Although there is still a positive relationship between CCI and the first dimensional loadings, it is no longer significant due to the inclusion of more explanatory variables. In this model, there is a significant negative relationship

between CCI and $\hat{\lambda}_i[6]$ based on the t-test. The scatter plot of CCI against $\hat{\lambda}_i[6]$ is given in the middle panel of Figure 2.8. We look into the rank one matrix $\hat{h}_6^T\hat{h}_6$, which is shown in Figure 2.7. It has positive connectivity within each hemisphere of the brain, but negative connectivity across hemispheres. This suggests that compared to within-hemisphere connectivity, across-hemisphere connectivity tends to have a more positive impact on human creativity.

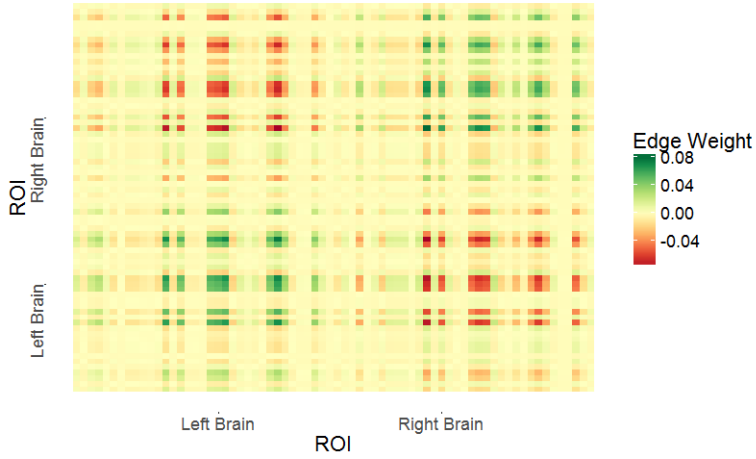

Figure 2.7: The plot shows the rank one matrix $\hat{h}_6^T\hat{h}_6$, which has positive connectivity within each hemisphere, but negative connectivity across hemispheres.

We also applied PCA to the vectorized adjacency matrices as described in (2.3.1), and then we regress CCI on PCA loadings. The log P-value of regression models are shown in Figure 2.9, with log(0.05) represented by the dash line. When the number of dimensions is small, our joint embedding and PCA yield similar results; however, the performance of PCA degrades quickly as the number of dimensions increase. When regressed on 10 dimensional loadings, the $p - value$ of joint embedding and PCA is

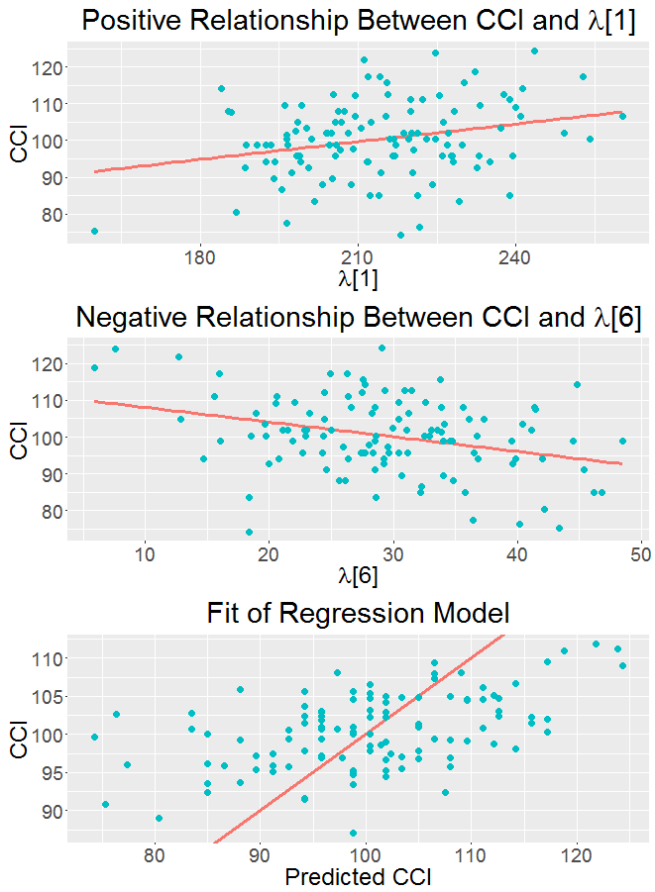

Figure 2.8: The top panel shows the scatter plot of CCI against $\hat{\lambda}_i[1]$ with the regression line. There is a positive relationship between CCI and first dimensional loadings. The middle panel shows the scatter plot of CCI against $\hat{\lambda}_i[6]$ with regression line. There is a negative relationship between CCI and sixth dimensional loadings. The bottom panel shows the predicted CCI versus true CCI with the identity line.

0.0018 and 0.16 respectively. The reason that joint embedding outperforms PCA in this setup is as explained before: the joint embedding only need to estimate a rank one graph for each dimension, which has 70 parameters in this application; however, the PCA needs has $2415 = \frac{70(70-1)}{2}$ parameters to fit for each dimension.

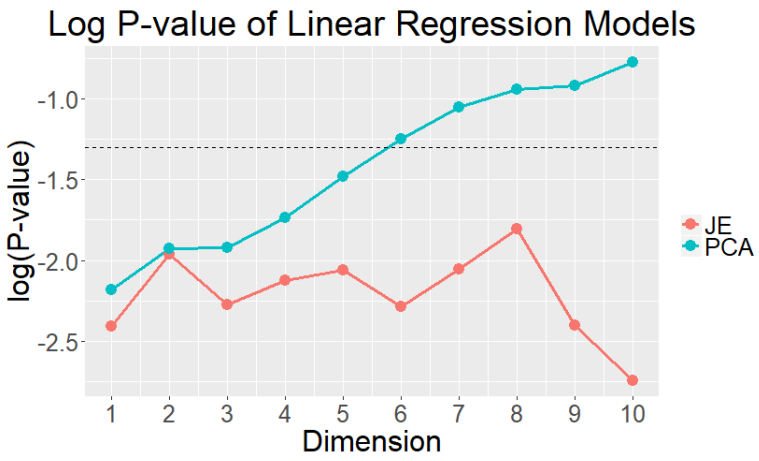

Figure 2.9: The log p-value of linear regression models using joint embedding and PCA are shown. The dash line is drawn at log(0.05). When the number of dimensions is small, the joint embedding and PCA yield similar results; however, the performance of PCA degrades quickly as the number of dimensions increase.

2.5.4 Real Data: Cluster Wikipedia Webpages

In the previous experiments, we focus on feature extraction for graphs through the joint embedding. Here, we consider a different task, that is spectral clustering through the joint embedding. In general, spectral clustering first computes (generalized) eigenvalues and eigenvectors of adjacency matrix or Laplacian matrix, then clustering the latent positions of vertices into groups [4, 5]. The cluster identities of latent positions become the cluster identities of vertices of the original graph. When applied to one graph, the joint embedding is equivalent to Adjacency Spectral Embedding (ASE), which is one of the spectral clustering algorithms. When given multiple graphs, the joint embedding can estimate latent positions for graph i as

$[\hat{\lambda}_i[1]^{\frac{1}{2}}\hat{h}_1, \hat{\lambda}_i[2]^{\frac{1}{2}}\hat{h}_2, ..., \hat{\lambda}_i[d]^{\frac{1}{2}}\hat{h}_d]$ or equivalently $\hat{\mathbf{H}}\hat{\mathbf{D}}_i^{\frac{1}{2}}$. Then, clustering algorithm can be applied to the latent positions.

We apply the spectral clustering approach to Wikipedia graphs [58]. The vertices of these graphs represent Wikipedia article pages. The two vertices are connected by an edge if either of the associated pages hyperlinks to the other. Two graphs are constructed based on English webpages and French webpages. The full graph has 1382 vertices which represents articles within 2-neighborhood of "Algebraic Geometry". Based on the content of the associated articles, they are grouped by hand into 6 categories: People, Places, Dates, Things, Math Terms, and Categories.

We consider a subset vertices from 3 categories: People, Things, Math Terms. After taking the induced subgraph of these vertices and removing isolated vertices, there are $n = 704$ vertices left. Specifically, 326, 181, and 197 vertices are from People, Things and Math Terms respectively. We consider 4 approaches to embed the graphs to obtain latent positions: ASE on the English graph (ASE+EN), ASE on the French Graph (ASE+FR), joint embedding on the English graph (JE+EN), and joint embedding on the French Graph (JE+FR). The dimension d is set to 3 for all approaches, and the latent positions are scaled to have norm 1 for degree correction. Then, we apply 3-means to the latent positions [59].

The latent positions of English graph estimated based on the joint embedding is provided in Figure 2.10. The latent positions of Math Terms are separated from the other two clusters. However, the latent positions of People and Things are mixed.

Table 2.2 shows the clustering results measured by adjusted rand index and the associated standard error [60, 61]. The standard error of adjusted rand index is estimated through repeatedly clustering bootstrapped latent positions. All methods yield clustering results which are significantly better than random. The English graph demonstrates clearer community structure than the French graph. The joint embedding produces latent positions which leads to better clustering performance compared to ASE. However, the difference between the joint embedding and ASE on English graph is not statistically significant based on the standard error estimated by bootstrap. Nevertheless, compared to ASE, joint embedding is able to improve clustering performance on French graph significantly, and produces at least comparable result on English graph, given the fact French graph is considerably worse than English graph. We expect joint embedding to be even better when given more graphs.

2.6 Discussion

In summary, we developed a joint embedding method that can simultaneously embed multiple graphs into low dimensional space. The joint embedding can be utilized to estimate features for inference problems on multiple vertex matched graphs. Learning on multiple graphs has significant applications in diverse fields and our results have both theoretical and practical implications for the problem. As the real data experiment illustrates, the joint embedding is a practically viable inference

Method	ASE+EN	ASE+FR	JE+EN	JE+FR
ARI	0.1456	0.1169	**0.1586**	0.1562
S.E.	0.0129	0.0139	0.0146	0.0136

Table 2.2: Clustering Performance on Wikipedia Graphs. The adjusted rand index (ARI) and the associated standard error (S.E.) of 4 spectral clustering approaches are shown. The best result is bolded. The standard error of adjusted rand index is estimated through repeatedly clustering bootstrapped latent positions. The joint embedding estimates latent positions which lead to better clustering performance than ASE.

procedure. We also proposed a Multiple Random Eigen Graphs model. It can be understood as a generalization of the Random Dot Product Graph model or the Stochastic Block Model for multiple random graphs. We analyzed the performance of joint embedding on this model under simple settings. We demonstrated that the joint embedding method provides estimates with bounded error. Our approach is intimately related to other matrix and tensor factorization approaches such as singular value decomposition and CP decomposition. Indeed, the joint embedding and these algorithms all try to estimate a low dimensional representation of high dimensional objects through minimizing a reconstruction error. We are currently investigating the utility of joint embedding with more or less regularizations on parameters and under different set ups. We are optimistic that our method provides a viable tool for

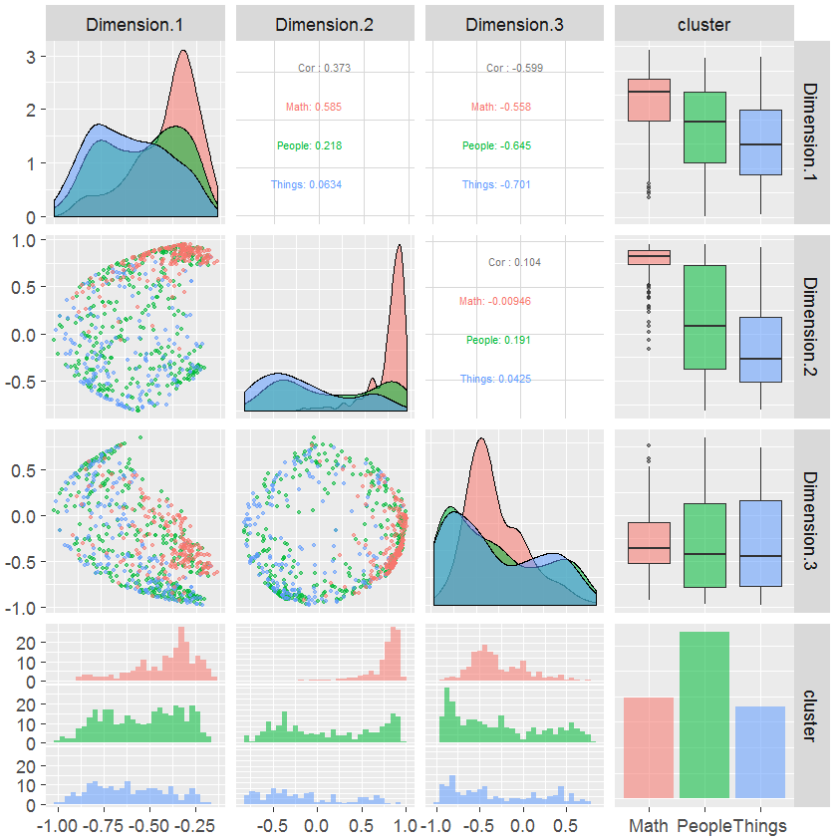

Figure 2.10: Latent positions of English Graph estimated by the joint embedding are shown. The first three plots on the diagonal are density estimates of latent positions for each dimension and category, and the last plot shows the number of points from each category. The first three plots of the last row show the histogram of latent positions for each dimension and category, and the first three plots of the last column are the corresponding box plot. The pairs plots of latent positions are given in the first three plots below the diagonal, the corresponding correlations are given above the diagonal. The latent positions of Math Terms are separated from the other two clusters. However, the latent positions of People and Things are mixed.

analyzing multiple graphs and can contribute to a deeper understanding of the joint structure of networks.

2.7 Proofs

Proof of Theorem 2.4.1 Denote the probability of observing a particular adjacency matrix $\mathbf{A}_i$ under distribution $\mathcal{F}$ by p_i. It suffices to show that there is a set of parameters for MREG such that observing $\mathbf{A}_i$ under MREG is also p_i.

For undirected graphs with loops on n vertices, there are $\binom{n+1}{2}$ possible edges. Let $\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_{2^{\binom{n+1}{2}}}$ be all the possible adjacency matrices. Since real symmetric matrix of size n has $\binom{n+1}{2}$ free entries which lies in a linear space, if there exists $\binom{n+1}{2}$ linearly independent rank one symmetric matrices, they form a basis for this space. It turns out that the rank one symmetric matrices generated by vectors $\{e_i\}_{i=1}^n \cup \{e_i+e_j\}_{i<j}$ are linearly independent, where $\{e_i\}_{i=1}^n$ is the standard basis for n-dimensional Euclidean space.

Next, we construct parameters for the MREG. Let d be $\binom{n+1}{2}$ and

$$\{h_k\}_{k=1}^d = \{e_i\}_{i=1}^n \cup \{\frac{e_i + e_j}{\sqrt{2}}\}_{i<j}.$$

Since $\{h_k h_k^T\}_{k=1}^d$ forms a basis for real symmetric matrices, for each adjacency matrix $\mathbf{A}_i$, there exists a vector λ_i, such that

$$\mathbf{A}_i = \sum_k \lambda_i[k] h_k h_k^T.$$

Let F be a finite mixture distribution on points $\{\lambda_i\}_{i=1}^{2^{\binom{n+1}{2}}}$, that is

$$F = \sum p_i \mathbb{I}\{\lambda = \lambda_i\}.$$

Under this MREG model, for any adjacency matrix $\mathbf{A}_i$

$$\mathbb{P}(\mathbf{A} = \mathbf{A}_i) = P(\lambda = \lambda_i) = p_i.$$

This concludes that the distribution $\mathcal{F}$ and $MREG(F, h_1, ..., h_d)$ are equal.

Proof of Theorem 2.4.2 First, we show that $|D_n(h, h_1) - D(h, h_1)|$ converges uniformly to 0. To begin with, notice three facts:

(1) the set $\{h : \|h\| = 1\}$ is compact;

(2) for all h, the function $\rho(\cdot, h)$ is continuous

(3) for all h, the function $\rho(\cdot, h)$ is bounded by n^2.

Therefore, by the uniform law of large numbers [62], we have

$$\sup_h |D_m(h, h_1) - D(h, h_1)| \xrightarrow{a.s.} 0.$$

To prove the claim of the theorem, we use a technique similar to that employed by Bickel and Doksum [63]. By definition, we must have $D_m(\hat{h}_1^m, h) \leq D_m(h', h)$ and $D(h', h) \leq D(\hat{h}_1^m, h)$. From these two inequalities,

$$D_m(h', h) - D(h', h) \geq D_m(\hat{h}_1^m, h) - D(h', h) \geq D_m(\hat{h}_1^m, h) - D(\hat{h}_1^m, h).$$

Therefore,

$$|D_m(\hat{h}_1^m, h) - D(h', h)| \leq \max(|D_m(h', h) - D(h', h)|, |D_m(\hat{h}_1^m, h) - D(\hat{h}_1^m, h)|).$$

This implies

$$|D_m(\hat{h}_1^m, h) - D(h', h)| \leq \sup_h |D_m(h, h_1) - D(h, h_1)|.$$

Hence, $|D_m(\hat{h}_1^m, h) - D(h', h)|$ must converge almost surely to 0, that is

$$|D_m(\hat{h}_1^m, h) - D(h', h)| \overset{a.s.}{\to} 0.$$

If $\hat{h}_1^m$ does not converge almost surely to h', then $\|\hat{h}_1^m - h'\| \geq \epsilon$ for some ϵ and infinitely many values of m. Since h' is the unique global minimum, $|D(\hat{h}_1^m, h) - D(h', h)| > \epsilon'$ for infinitely many values of m and some ϵ'. This contradicts with the previous equation. Therefore, $\hat{h}_1^m$ must converge almost surely to h'.

Proof of Theorem 2.4.3 The proof of theorem relies on two lemmas. The first lemma shows that h' is the eigenvector corresponding to the largest eigenvalue of $\mathbb{E}(\langle \mathbf{A}_i, h'h'^T \rangle \mathbf{A}_i)$. The second lemma bounds the Frobenius norm difference between $\mathbb{E}(\langle \mathbf{A}_i, h'h'^T \rangle \mathbf{A}_i)$ and $\mathbb{E}(\lambda_i^2)(h_1^T h')^2 h_1 h_1^T$. Then, we apply Davis-Kahan theorem [64] to establish the claim of theorem.

Lemma 2.7.1 *The vector h' is the eigenvector corresponding to the largest eigenvalue of $E(\langle \mathbf{A}_i, h'h'^T \rangle \mathbf{A}_i)$.*

We notice that

$$\begin{aligned}\min_{\|h\|=1} D(h, h_1) &= \min_{\|h\|=1} \mathbb{E}(\|\mathbf{A}_i - \langle \mathbf{A}_i, hh^T\rangle hh^T\|^2) \\ &= \min_{\|h\|=1} \mathbb{E}(\langle \mathbf{A}_i, \mathbf{A}_i\rangle - \langle \mathbf{A}_i, hh^T\rangle^2) \\ &= \mathbb{E}(\langle \mathbf{A}_i, \mathbf{A}_i\rangle) - \max_{\|h\|=1} \mathbb{E}(\langle \mathbf{A}_i, hh^T\rangle^2).\end{aligned}$$

Therefore,

$$h' = \operatorname*{argmin}_{\|h\|=1} D(h, h_1) = \operatorname*{argmax}_{\|h\|=1} \mathbb{E}(\langle \mathbf{A}_i, hh^T\rangle^2). \tag{2.6}$$

Taking the derivative of $\mathbb{E}(\langle \mathbf{A}_i, hh^T\rangle^2) + c(h^T h - 1)$ with respect to h,

$$\begin{aligned}\frac{\partial \mathbb{E}(\langle \mathbf{A}_i, hh^T\rangle^2) + c(h^T h - 1)}{\partial h} &= \mathbb{E}(\frac{\partial \langle \mathbf{A}_i, hh^T\rangle^2}{\partial h}) + 2ch \\ &= 4\mathbb{E}(\langle \mathbf{A}_i, hh^T\rangle \mathbf{A}_i)h + 2ch.\end{aligned}$$

Setting this expression to 0 yields,

$$\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle \mathbf{A}_i)h' = -\frac{1}{2}ch'.$$

Using the fact that $\|h'\| = 1$, we can solve for c:

$$c = -2h'^T \mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle \mathbf{A}_i)h' = -2\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle^2).$$

Then, substituting for c,

$$\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle \mathbf{A}_i)h' = \mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle^2)h'. \tag{2.7}$$

Therefore, we see that h' is an eigenvector of $\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle \mathbf{A}_i)$ and the corresponding eigenvalue is $\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle^2)$. Furthermore, $\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle^2)$ must be the eigenvalue

with the largest magnitude. For if not, then there exists an h'' with norm 1 such that

$$|h''^T\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle \mathbf{A}_i)h''| = |\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle\langle \mathbf{A}_i, h''h''^T\rangle)| > \mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle^2);$$

however, by Cauchy-Schwarz inequality we must have

$$\mathbb{E}(\langle \mathbf{A}_i, h''h''^T\rangle^2)\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle^2) > |\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle\langle \mathbf{A}_i, h''h''^T\rangle)|^2.$$

This implies $\mathbb{E}(\langle \mathbf{A}_i, h''h''^T\rangle^2) > \mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle^2)$, which contradicts equation (2.6). Thus, we conclude that h' is the eigenvector corresponding to the largest eigenvalue of $\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle \mathbf{A}_i)$.

Next, we consider the norm difference between $\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle \mathbf{A}_i)$ and $\mathbb{E}(\lambda_i^2)(h_1^T h')^2 h_1 h_1^T$.

Lemma 2.7.2

$$\|\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle \mathbf{A}_i) - \mathbb{E}(\lambda_i^2)(h_1^T h')^2 h_1 h_1^T\| \leq 2E(\lambda_i).$$

We compute $\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle \mathbf{A}_i)$ by conditioning on $\mathbf{P}_i$.

$$\begin{aligned}
&\mathbb{E}(\langle \mathbf{A}_i, h'h'^T\rangle \mathbf{A}_i|\mathbf{P}_i) \\
&= \mathbb{E}(\langle \mathbf{A}_i - \mathbf{P}_i, h'h'^T\rangle(\mathbf{A}_i - \mathbf{P}_i)|\mathbf{P}_i) + \mathbb{E}(\langle \mathbf{A}_i - \mathbf{P}_i, h'h'^T\rangle \mathbf{P}_i)|\mathbf{P}_i) \\
&\quad + \mathbb{E}(\langle \mathbf{P}_i, h'h'^T\rangle(\mathbf{A}_i - \mathbf{P}_i)|\mathbf{P}_i) + \mathbb{E}(\langle \mathbf{P}_i, h'h'^T\rangle \mathbf{P}_i|\mathbf{P}_i) \\
&= \mathbb{E}(\langle \mathbf{A}_i - \mathbf{P}_i, h'h'^T\rangle(\mathbf{A}_i - \mathbf{P}_i)|\mathbf{P}_i) + \lambda_i(h_1^T h')^2 \mathbf{P}_i \\
&= 2h'h'^T * \mathbf{P}_i * (\mathbf{J} - \mathbf{P}_i) - DIAG(h_1 h_1^T * \mathbf{P}_i * (\mathbf{J} - \mathbf{P}_i)) + \lambda_i(h_1^T h')^2 \mathbf{P}_i.
\end{aligned}$$

Here, DIAG() means only keep the diagonal of the matrix; $*$ means the Hadamard

product, and $\mathbf{J}$ is a matrix of all ones. Using the fact that $\mathbf{P}_i = \lambda_i h_1 h_1^T$, we have

$$\begin{aligned}
&\mathbb{E}(\langle \mathbf{A}_i, h'h'^T \rangle \mathbf{A}_i) - E(\lambda_i^2)(h_1^T h')^2 h_1 h_1^T \\
&= \mathbb{E}(\mathbb{E}(\langle \mathbf{A}_i, h'h'^T \rangle \mathbf{A}_i | \mathbf{P}_i) - \lambda_i (h_1^T h')^2 \mathbf{P}_i) \\
&= \mathbb{E}(2h'h'^T * \mathbf{P}_i * (\mathbf{J} - \mathbf{P}_i) - DIAG(h'h'^T * \mathbf{P}_i * (\mathbf{J} - \mathbf{P}_i))).
\end{aligned}$$

If we consider the norm difference between $\mathbb{E}(\langle \mathbf{A}_i, h'h'^T \rangle \mathbf{A}_i)$ and $\mathbb{E}(\lambda_i^2)(h_1^T h')^2 h_1 h_1^T$, we have

$$\begin{aligned}
&\|\mathbb{E}(\langle \mathbf{A}_i, h'h'^T \rangle \mathbf{A}_i) - \mathbb{E}(\lambda_i^2)(h_1^T h')^2 h_1 h_1^T\| \\
&= \|\mathbb{E}(2h'h'^T * \mathbf{P}_i * (\mathbf{J} - \mathbf{P}_i) - DIAG(h'h'^T * \mathbf{P}_i * (\mathbf{J} - \mathbf{P}_i)))\| \\
&\leq \mathbb{E}(\|2h'h'^T * \mathbf{P}_i * (\mathbf{J} - \mathbf{P}_i) - DIAG(h'h'^T * \mathbf{P}_i * (\mathbf{J} - \mathbf{P}_i))\|) \\
&\leq \mathbb{E}(\|2h'h'^T * \mathbf{P}_i * (\mathbf{J} - \mathbf{P}_i)\|) \\
&\leq \mathbb{E}(\|2h'h'^T * \mathbf{P}_i\|) \\
&\leq 2\mathbb{E}(\lambda_i)\|h'h'^T * h_1 h_1^T\| \\
&= 2\mathbb{E}(\lambda_i).
\end{aligned}$$

This finishes the proof of the lemma.

Notice that the only non-zero eigenvector of $\mathbb{E}(\lambda_i^2)(h_1^T h')^2 h_1 h_1^T$ is h_1 and the corresponding eigenvalue is $\mathbb{E}(\lambda_i^2)(h_1^T h')^2$. We apply the Davis-Kahan theorem [64] to the eigenvector corresponding to the largest eigenvalue of matrices $\mathbb{E}(\langle \mathbf{A}_i, h'h'^T \rangle \mathbf{A}_i)$ and $\mathbb{E}(\lambda_i^2)(h_1^T h')^2 h_1 h_1^T$, yielding

$$\|h' - h_1\| \leq \frac{2\mathbb{E}(\lambda_i)}{\mathbb{E}(\lambda_i^2)(h_1^T h')^2}.$$

Chapter 3

Signal Subgraph Estimation via Vertex Screening

3.1 Introduction

Graph classification and regression are crucial to analyze data sets in various fields such as neuroscience, internet mapping, and social networks [1, 3, 65]. Given a set of graphs $\{G_i\}_{i=1}^m$ along with a set of corresponding covariates $\{Y_i\}_{i=1}^m$, we would like to predict the covariate Y_i based on the graph G_i. However, G can be extremely large in practice, e.g., the social networks and raw neuroimages can have millions of vertices [66], which is a great challenge computationally without first reducing the size of the graph. Therefore, it is imperative to come up with an accurate and efficient method for signal subgraph estimation. Signal subgraph extraction tries to locate a

subgraph of G that contains all the useful information about Y, which can be helpful to improve the subsequent inference performance. However, estimating the signal subgraph is very challenging for large graphs, because a graph with n vertices could have 2^n different induced subgraphs.

When the number of features is large, dimension reduction and feature selection is generally difficult and expensive, which is a challenge to many modern real data sets. To overcome this challenge, Fan and Lv [7] proposed the feature screening algorithm and showed that ranking variables via the Pearson's correlation possesses a sure screening property under linear regression models. Screening through marginal likelihood are later considered for generalized linear models [67, 68]. Motivated by their approaches, we develop a vertex screening procedure to estimate the signal subgraph.

To screen the vertices effectively requires a sufficient measure of "correlation". Although Pearson's product moment correlation has been a popular choice, it only captures linear association and thus is not a good candidate for identifying general dependencies. The recently proposed distance correlation (Dcorr) [69–71] is able to detect all types of dependencies between two random variables consistently. The later proposed multiscale generalize correlation (MGC) [72–74] is a localized version of Dcorr, which shares the consistency property with improved finite-sample testing power against nonlinear dependencies. Consistent screening under a model-free setting via distance correlation was proposed and investigated in [75, 76].

We therefore combine distance-based correlation and screening to yield an effective vertex screening method to estimate the signal subgraph, which works efficiently and tackles all inherent challenges. The methodology consists of three steps: (i) feature computation, (ii) calculating the distance-based correlation, and (iii) thresholding. The first step computes a feature for each vertex based the graph. The second step calculates a distance-based correlation measure between the feature of each vertex and the label of interest Y over all graphs. The last step thresholds the correlations and only keeps the vertices with large correlations. We further developed an iterative vertex screening algorithm, in which the three steps are applied recursively to improve the performance without sacrificing the running time. In the next section, we introduce the signal graph estimation problem and presend the vertex screening algorithm. Finally, we conclude the chapter with a discussion about possible future extensions.

3.2 Preliminaries

3.2.1 Signal Subgraph Estimation Problem

Given m graphs $\{G_i, i = 1, \ldots, m\}$ with a shared vertex set $V = [n]$, let $A_i \in \mathbb{R}^{n \times n}$ be the adjacency matrix of G_i for each i, which can be weighted or un-weighted, directed or un-directed. Additionally, there is a covariate of interest $\{Y_i \in \mathbb{R}, i = 1, \ldots, m\}$ associated to each graph. A common example is a neuroimaging study,

where the human brain image of a subject is used to produce a graph G_i, and for each subject there is an associated variable Y_i representing a group label, covariate or additional phenotype (e.g. behavior, genotype or sex). Information on the brain graph and phenotype pair is collected for m subjects. In this paper, we focus on the case that Y_i is a scalar label, but the screening algorithm is readily applicable to any multivariate Y_i. The classical statistical pattern recognition set up is that $\{(G_i, Y_i)\}_{i=1}^m$ are independent and identically distributed pairs according to some distribution $F_{G,Y}$ [77], that is

$$(G_1, Y_1), (G_2, Y_2), (G_3, Y_3), ..., (G_m, Y_m) \overset{i.i.d.}{\sim} F_{G,Y}$$

for some true but unknown joint distribution.

It is often the case that the covariate Y only depends on a small part of G. In addition, merely predicting Y is insufficient in some applications. It is desirable to know which vertices or subgraph is associated to Y. Therefore, it is natural to search for a signal subgraph such that Y is independent of other parts of the graph. This motivates our definition of signal vertices and signal subgraph.

Definition For any subset of vertices $U \subset V = [n]$, denote the induced subgraph of U by $G[U]$, and denote the subgraph removing all edges in $G[U]$ as $G \backslash G[U]$.

The set of ***signal vertices*** S is defined to be the minimal subset of vertices U, such that $G \backslash G[U]$ is independent of Y, that is

$$S = \arg\min_U |U| \text{ , subject to } G \backslash G[U] \perp Y,$$

where the notation $\perp$ means independence, or $F_{G\setminus G[U],Y} = F_{G\setminus G[U]}F_Y$. The induced graph $G[S]$ on the signal vertices is called the ***signal subgraph***.

If the graph G is independent of Y, there is no signal in the graph and the signal subgraph is empty in this case. If all vertices in G are incident on at least one edge which is dependent on Y, the signal subgraph is the whole graph G. Moreover, there can be multiple subsets attaining the minimum, so for ease of presentation we assume there exists a unique signal subgraph $G[S]$.

In practice, m graph-covariate pairs $\{(A_i, Y_i)\}_{i=1}^m$ are observed, we want to estimate the signal subgraph $G[S]$. The subsequent statistical inference can benefit from the bias-variance trade-off or statistical parsimony, if vertices with weak or no signal can be screened out effectively.

3.2.2 Bayes Plug-in Classifier

We introduce a binary classification problem that is predicting the label $Y \in \{0, 1\}$ using graph G, which serves as the foundation for Section 3.4 and later simulations. The network model under consideration is the inhomogeneous Erdos-Renyi (IER) random graph model [22], which is defined in the last chapter. The class label is built into this model as follows: suppose the graph follow IER model conditioned on Y, that is

$$A|Y = y \sim IER(P^y) \qquad \text{for } y \in \{0, 1\}$$

Under this setting, it is clear that vertex u is a signal vertex if and only if $P^0[u, v] \neq P^1[u, v]$ for some vertex v, that is

$$S = \{u \in V | \exists v \in V, P^0[u, v] \neq P^1[u, v]\}.$$

Given this model, it is known that the optimal classification performance is achieved by the Bayes classifier g^* [77], which is defined

$$g^*(A) = \begin{cases} 1 & \text{if } \pi_0 \mathcal{L}(A; P^0) < \pi_1 \mathcal{L}(A; P^1), \\ 0 & \text{if } \pi_0 \mathcal{L}(A; P^0) \geq \pi_1 \mathcal{L}(A; P^1), \end{cases}$$

where π_0 and π_1 are prior probabilities for each class. In practice, it is natural to consider the Bayes plug-in classifier which estimates the π_y and P^y and plug them into the likelihood. In this case, the maximum likelihood estimates of parameters are

$$\hat{\pi}_y = \frac{\sum_i \mathbb{I}\{Y_i = y\}}{m},$$

$$\hat{P}^y = \frac{\sum_i \mathbb{I}\{Y_i = y\} A_i}{\sum_i \mathbb{I}\{Y_i = y\}}.$$

Using these estimates, we can construct the Bayes plug-in classifier g_V based on the whole graph, that is

$$g_V(A) = \begin{cases} 1 & \text{if } \hat{\pi}_0 \mathcal{L}(A; \hat{P}^0) < \hat{\pi}_1 \mathcal{L}(A; \hat{P}^1), \\ 0 & \text{if } \hat{\pi}_0 \mathcal{L}(A; \hat{P}^0) \geq \hat{\pi}_1 \mathcal{L}(A; \hat{P}^1). \end{cases}$$

When we have an estimate of the signal subgraph $G[\hat{S}]$, we could also consider Bayes plug-in classifier $g_{\hat{S}}$ based on the estimated signal subgraph, that is

$$g_{\hat{S}}(A) = \begin{cases} 1 & \text{if } \hat{\pi}_0 \mathcal{L}(A[\hat{S}]; \hat{P}^0[\hat{S}]) < \hat{\pi}_1 \mathcal{L}(A[\hat{S}]; \hat{P}^1[\hat{S}]), \\ 0 & \text{if } \hat{\pi}_0 \mathcal{L}(A[\hat{S}]; \hat{P}^0[\hat{S}]) \geq \hat{\pi}_1 \mathcal{L}(A[\hat{S}]; \hat{P}^1[\hat{S}]), \end{cases}$$

where

$$\mathcal{L}(A[\hat{S}]; \hat{P}^y[\hat{S}]) = \prod_{u,v \in \hat{S}} A[u,v]^{\hat{P}^y[u,v]} (1 - A[u,v])^{(1-\hat{P}^y[u,v])}.$$

Similarly, we use g_S to denote the Bayes plug-in classifier based on the true signal subgraph.

To evaluate the classification performance, we consider the $0-1$ loss or classification error L. For a classifier g, the loss $L(g)$ is defined by

$$L(g) = \mathbb{P}(g(A) \neq Y).$$

In Section 3.4, we investigate how $L(g_V)$ and $L(g_{\hat{S}})$ behave, i.e., the classification error for the full graph versus the classification error for the estimated signal subgraph.

3.2.3 Distance Correlation and Multiscale Generalized Correlation

The distance correlation (Dcorr) [69, 70] and multiscale generalize correlation (MGC) [72, 73] measure dependency between two random variables $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$. Here, we review their definitions. First, the distance covariance $Dcov(X, Y)$

is given by

$$Dcov(X,Y) = \frac{1}{c_p c_q} \int\int \frac{|\phi_{X,Y}(s,t) - \phi_X(s)\phi_Y(t)|^2}{\|s\|^{1+p}\|t\|^{1+q}} dtds, \tag{3.1}$$

where $\phi_{X,Y}$, ϕ_X and ϕ_Y are characteristic functions of (X,Y), X and Y respectively, and c_p, c_q are constants. When X and Y have finite second moment, it can be shown that the distance correlation can be alternatively defined by

$$Dcov(X,Y) = \mathbb{E}(\|X - X'\|\|Y - Y'\|) + \mathbb{E}(\|X - X'\|)\mathbb{E}(\|Y - Y'\|)$$
$$- 2\mathbb{E}(\|X - X'\|\|Y - Y''\|),$$

where $(X,Y), (X',Y'), (X'',Y'')$ are independent and identically distributed as F_{XY}. Distance covariance between two random variables X and Y is always non-negative, and it equals 0 if and only if X and Y are independent. The distance correlation $Dcorr(X,Y)$ between X and Y is

$$Dcorr(X,Y) = \frac{Dcov(X,Y)}{\sqrt{Dcov(X,X)Dcov(Y,Y)}},$$

which lies in $[0,1]$.

When applying distance correlation to sample data, the sample distance correlation is defined by properly centering Euclidean distance matrices, followed by taking a Hadamard product. The sample Dcorr converges to the population Dcorr as sample size increases to infinity, therefore we concentrate on analyzing the population Dcorr in the theoretical proofs.

The more recent multiscale generalized correlation (MGC) is a local optimal version of distance correlation: when evaluating the integral in Equation (3.1), the char-

acteristic function is truncated to a neighborhood, which can be shown to yield a larger statistic and better test power under a wide variety of high-dimensional and non-linear dependence cases. A detailed discussion of MGC is in [73], which essentially shares the same theoretical properties as distance correlation.

Given pairs of samples $\{X_i, Y_i\}_{i=1}^m$, the sample distance covariance can be computed in the following steps. First, pairwise Euclidean distance matrices C and D are computed, that is

$$C_{ij} = \|X_i - X_j\| \text{ and } D_{ij} = \|Y_i - Y_j\|.$$

Then the two distance are double centered.

$$c_{ij} = C_{ij} - C_{i\cdot} - C_{\cdot j} + C_{\cdot\cdot},$$

where $C_{i\cdot}$, $C_{\cdot j}$ and $C_{\cdot\cdot}$ are the mean of ith row, jth column and the whole matrix respectively. Similarly,

$$d_{ij} = D_{ij} - D_{i\cdot} - D_{\cdot j} + D_{\cdot\cdot}.$$

The sample distance covariance $Dcov(\{(X_i, Y_i)\}_{i=1}^m)$ is defined by

$$Dcov(\{(X_i, Y_i)\}_{i=1}^m) = \frac{1}{n^2} \sum_{i,j} c_{ij} d_{ij}.$$

The sample distance correlation $Dcorr(\{(X_i, Y_i)\}_{i=1}^m)$ can be computed by

$$\frac{Dcov(\{(X_i, Y_i)\}_{i=1}^m)}{\sqrt{Dcov(\{(X_i, X_i)\}_{i=1}^m) Dcov(\{(Y_i, Y_i)\}_{i=1}^m)}}.$$

To calculate sample MGC, we start with computing local distance correlation at scale k and l. Let $R(C_{\cdot j}, i)$ be the rank of X_i relative to X_j, that is, $R(C_{\cdot j}, i) = k$ if

X_i is the k^{th} closest point (or neighbor) to X_j, as determined by ranking the $n-1$ distances to X_j. Define $R(D_{i\cdot}, j)$ equivalently for the Ys, but ranking relative to the rows rather than the columns. For any neighborhood size k around each X_i and any neighborhood size l around each Y_j, we set distance outside the neighborhood to 0:

$$c_{ij}^k = \begin{cases} C_{ij}, & \text{if } R(A_{\cdot j}, i) \leq k, \\ 0, & \text{otherwise;} \end{cases} \qquad d_{ij}^l = \begin{cases} D_{ij}, & \text{if } R(B_{i\cdot}, j) \leq l, \\ 0, & \text{otherwise;} \end{cases} \tag{3.2}$$

and then let $c_{ij}^k = c_{ij}^k - \bar{c}^k$, where $\bar{c}^k$ is the mean of $\{c_{ij}^k\}$, and similarly for d_{ij}^l. Then, the sample local distance correlation $Lcorr^{kl}$ at scale k and l can be calculated

$$Lcorr^{kl} = \frac{\sum_{i,j} c_{ij}^k d_{ij}^l}{\sqrt{\sum_{i,j}(c_{ij}^k)^2 \sum_{i,j}(d_{ij}^l)^2}}.$$

The sample MGC is defined to be the maximum among local distance correlation over all possible scales [73], that is

$$MGC(\{(X_i, Y_i)\}_{i=1}^m) = \max_{k,l} Lcorr^{kl}.$$

3.3 Vertex Screening

The vertex screening procedure provides an estimate of signal subgraph $G[\hat{S}]$ via the following steps: feature extraction, distance-based correlation computation and thresholding. We also develop an iterative vertex screening procedure, which applies the three steps recursively. We will first present the non-iterative vertex screening, followed by the iterative version.

The first step extracts a feature vector for each vertex in a graph. We use the notation $\hat{X}_i[u, \cdot]$ to denote the feature extracted for vertex u in graph i where $i \in [m]$ and $u \in [n]$. A simple example is setting $\hat{X}_i[u, \cdot]$ to be the uth row of adjacency matrix A_i, that is $\hat{X}_i[u, \cdot] = A_i[u, \cdot]$. As a result, $\hat{X}_i[u, \cdot]$ is a vector in $\mathbb{R}^n$ which can be a high dimensional space. Alternatively, summary statistics can be treated as a feature vector. For example, the number of vertices within k-neighborhood of the vertex or eccentricity of the vertex can be used as the feature for the vertex [78, 79]. Spectral methods could also be applied to extract a feature vector which lies in $\mathbb{R}^d$. For example, Adjacency Spectral Embedding [4] and Joint Embedding [6] could recover a low dimension latent position for each vertex. In this paper, we focus on using adjacency vector as the vertex feature for simplicity.

The second step computes sample distance-based correlation between the feature vector $\{\hat{X}_i[u, \cdot]\}_{i=1}^m$ and label $\{Y_i\}_{i=1}^m$ for each vertex $u \in V$. The correlation choice is either distance correlation (Dcorr) or multiscale generalized correlation (MGC). Denote the distance-based correlation by c_u, that is

$$c_u = Dcorr(\{(\hat{X}_i[u, \cdot], Y_i)\}_{i=1}^m), \text{ or}$$

$$c_u = MGC(\{(\hat{X}_i[u, \cdot], Y_i)\}_{i=1}^m).$$

The motivation of Dcorr and MGC is that they can detect any kind of dependency when the sample size is large enough. Generally speaking, we recommend using MGC when m is small but to use Dcorr when m is large. This is because Dcorr runs in $\mathcal{O}(m^2 n)$ while MGC runs in $\mathcal{O}(m^2 nlog(m))$. Then for small m, the computation

difference is negligible while MGC can be more powerful against general dependencies; while for m large (like above 1000 graphs) the power difference is negligible against most dependencies due to the consistency, and Dcorr wins in the running time.

The last step orders $\{c_u\}_{u \in V}$ by their magnitudes, and we threshold the correlations by a critical value c. The vertices surviving the threshold are the estimated signal vertices $\hat{S}$, that is

$$\hat{S} = \{u \in V | c_u > c\}.$$

The estimated signal subgraph and the corresponding adjacency matrix are denoted by $G[\hat{S}]$ and $A[\hat{S}]$ respectively. Algorithm 3 describes the general procedure of vertex screening using adjacency vector as the feature vector.

Algorithm 3 Vertex Screening.

Require: $\{(A_i, Y_i)\}_{i=1}^m$ and $c \in [0, 1]$

1: **for** $u \in V$ **do**

2: $\quad c_u = Dcorr(\{(A_i[u, \cdot], Y_i)\}_{i=1}^m)$

3: **end for**

4: $\hat{S} = \{u \in V | c_u > c\}$.

We observe that vertex feature vector $\hat{X}_i[u, \cdot]$ has dimension n, which is the number of vertices. If the vertex screening is performed on a smaller graph, $\hat{X}_i[u, \cdot]$ has fewer dimension and is more likely to exhibit a stronger signal via a larger distance-based correlation statistic with Y_i for a signal vertex. This observation motivates the iterative version that repeatedly applies Algorithm 3, i.e., at each iteration, only

a small proportion δ of all vertices are removed from the graph. The size of the subgraph is iteratively reduced until size 1 or some pre-determined number. Among all possible subgraphs, pick the subgraph that has the largest Dcorr or MGC statistic with the class label. The details are described by Algorithm 4, where $A_i[V_k]$ denotes the adjacency matrix of V_k induced subgraph of graph i.

Alternatively, other possible methods to select the subgraph include: 1) use cross-validation [48] to select the size of the subgraph with the best leave-one-out prediction error, which can be computationally expensive; 2) order the correlations $\{c_u\}_{u \in V}$ to locate a gap among correlations, and select the vertices larger than this gap [80]; 3) background information available could determine the number of vertices which could have signal. In the experiment section, we will verify the iterative screening method that maximizes the statistics, which works very well and almost always achieves the best leave-one-out prediction error.

Note that the iterative algorithm circumvents choosing the threshold c by designating a δ. For large graphs, empirically it suffices to let δ be 0.5, which achieves an excellent performance with only a $\log(n)$ factor increase in running time. For graphs with a small number of vertices, the running time is not a issue; one may let δ be 0.05 or even reduce the size of subgraph by 1 in each iteration.

Algorithm 4 Iterative Vertex Screening.

Require: $\{(A_i, Y_i)\}_{i=1}^m$, $\delta \in (0, 1)$

Set $k = 1$, and $V_k = V$

while $|V_k| > 1$ **do**

  for $u \in V_k$ **do**

    $c_u = Dcorr(\{(A_i^{V_k}[u, \cdot], Y_i)\}_{i=1}^m)$

  end for

  Set t_k be the δ quantile among $\{c_u, u \in V_k\}$

  Set $V_{k+1} = \{u \in V_k | c_u > t_k\}$

  Set $k = k + 1$

end while

$k^* = \arg\max_k Dcorr(\{(A_i^{V_k}[V_k, \cdot], Y_i)\}_{i=1}^m)$

Output the signal vertices $\hat{S} = V_{k^*}$.

3.4 Theoretical Results

3.4.1 Screening Theory

The next theorem states that if the threshold t is small enough to make sure $|\hat{S}| > |S|$, then $\hat{S}$ equals to S with high probability as the number graphs increases. This theorem is a direct consequence by Li, Zhong and Zhu [76].

Theorem 3.4.1 *If the following condition is satisfied*

$$\min Dcorr(A[u, \cdot], Y) \geq c > 0 \qquad \text{for } u \in S,$$

then $\hat{S}$ contains S with high probability. Specifically, there exist two constants $c_1, c_2 > 0$, for any $0 < \gamma < 1/2$,

$$\mathbb{P}(S \subset \hat{S}) > 1 - O(n \exp(-c_1 m^{1-2\gamma}) + n^2 \exp(-c_2 m^{\gamma})).$$

The theorem states that the estimated signal subgraph contains the true signal subgraph with high probability. Actually, it is also possible to derive a threshold t to ensure $P(S = \hat{S})$ as the number of graphs goes to infinity. For the proof of, please refer to Theorem 1 in [76]. Similar results also hold for MGC as well [73].

3.4.2 Justification on Iterative Screening

Despite the consistency of screening proven above, the finite-sample performance for non-iterative screening can be often improved by iterative screening. The next

theorem justifies this phenomenon, which demonstrates that the signal vertices will have its signal amplified under distance correlation by eliminating the noise vertices.

To simplify the discussion, we assume the feature vector X consists of two sets of entries that is $X = [X^*, Z]$, where $X^* \in \mathbb{R}^p$ and $Z \in \mathbb{R}^r$. Suppose X^* is the true signal and is dependent on Y, while Z is noise and is independent of Y and X^*. The first Lemma claims that the distance covariance between X and Y increases after removing the noise entries Z.

Lemma 3.4.2 *Suppose that $X = [X^*, Z] \in \mathbb{R}^p \times \mathbb{R}^r$, where $X^* \not\perp Y$, $Z \perp Y$, and $Z \perp X^*$. Then,*

$$Dcov(X^*, Y) \geq Dcov(X, Y).$$

If we let r increase by adding more noise entries to X, the next theorem demonstrates that the distance correlation will decrease to 0 as r goes to infinity.

Theorem 3.4.3 *Suppose that $X_r = [X^*, Z_r] \in \mathbb{R}^p \times \mathbb{R}^r$, where $X^* \not\perp Y$, $Z_r \perp Y$, and $Z_r \perp X^*$. Assume $Z_r \in \mathbb{R}^r$ has independent and identically distributed entries, then*

$$\lim_{r \to \infty} Dcorr(X_r, Y) = 0.$$

Therefore, if the screening algorithm iteratively eliminates the noise vertices, the distance correlation between the signal vertex and label will become larger and larger. As a result, iterative screening can provide a more accurate ranking of signal vertices than one-time screening.

3.4.3 Classification Improvement

The result next shows that the estimated signal subgraph indeed helps the classification. Let e denote the number of possible edges on the graph (which is $O(n^2)$ except sparse graphs), and denote the minimum of class priors by α, that is $\alpha = \min\{\pi_0, \pi_1\}$. We first analyze the performance of Bayes plug-in classifier based on the whole graph g_V. The next theorem states that its prediction error $L(g_V)$ converges to the Bayes optimal error $L(g^*)$ as the number of graphs goes to infinity.

Theorem 3.4.4 *With high probability, $L(g_V) - L(g^*)$ is bounded by ϵ, that is*

$$\mathbb{P}(L(g_V) - L(g^*) < \epsilon) \geq 1 - 2(e+1)\exp\left(\frac{-m\alpha\epsilon^2}{(2e+\sqrt{2\alpha})^2}\right).$$

Alternatively, with probability at least $1-\eta$

$$L(g_V) - L(g^*) \leq (2e+\sqrt{2\alpha})\sqrt{\frac{\log(\frac{2(e+1)}{\eta})}{m\alpha}}.$$

An immediate consequence of the theorem above is the following.

Corollary 3.4.5 *For small $\epsilon > 0$,*

$$\mathbb{E}(L(g_V)) \leq L(g^*) + \epsilon + 2(e+1)\exp\left(\frac{-m\alpha\epsilon^2}{(2e+\sqrt{2\alpha})^2}\right).$$

The theorem and corollary above consider predicting Y based on the whole graph. If we first apply vertex screening and then predict Y based on estimated signal subgraph $\hat{S}$ using the Bayes plug-in classifier, we will have the following results by applying Theorem 3.4.1.

Theorem 3.4.6 *With high probability, $L(g_{\hat{S}}) - L(g^*)$ is bounded by ϵ. Specifically, there exist constants c_1, c_2, and c_3, such that*

$$\mathbb{P}(L(g_{\hat{S}}) - L(g^*) < \epsilon) \geq 1 - 2(e_s + 1)\exp\left(\frac{-m\alpha\epsilon^2}{(2e_s + \sqrt{2\alpha})^2}\right) - c_3(n\exp(-c_1 m^{\frac{1}{3}}) + n^2\exp(-c_2 m^{\frac{1}{3}})),$$

where e_s is the number of possible edges in the estimated signal subgraph.

Corollary 3.4.7 *For any $\epsilon > 0$, there exist three constants c_1, c_2 and c_3,*

$$\mathbb{E}(L(g_{\hat{S}})) < L(g^*) + \epsilon + 2(e_s + 1)\exp\left(\frac{-m\alpha\epsilon^2}{(2e_s + \sqrt{2\alpha})^2}\right) + c_3(n\exp(-c_1 m^{\frac{1}{3}}) + n^2\exp(-c_2 m^{\frac{1}{3}})).$$

Comparing Theorem 3.2 and 3.4, we can see that if n, $|S|$ and $|\hat{S}|$ are fixed, $L(g_V)$ and $L(g_{\hat{S}})$ are both converging to $L(g^*)$ with m going to infinity. In fact, prediction based on the whole graph converges at a faster rate to the Bayes optimal. If n, $|S|$ and $|\hat{S}|$ increase faster than $m^{\frac{1}{2}}$, then prediction with or without screening have no error bound guarantees. However, if $|S|$ and $|\hat{S}|$ are fixed and n does not grow faster than $m^{\frac{1}{2}}$, only vertex screening guarantees convergence of prediction error. We state that in the next theorem.

Theorem 3.4.8 *Assume $|S|$ and $|\hat{S}|$ are fixed, and $n \in O(\exp(m^{\frac{1}{6}}))$ and $m \in o(n^2)$, then $L(g_{\hat{S}}) \to L(g^*)$ while $L(g_V)$ does not converge to the Bayes optimal error.*

This justifies the importance of extracting signal subgraph in prediction when the graphs are large.

3.5 Numerical Results

3.5.1 Simulation: Vertex Screening under IER

In this experiment, we investigate the performance of vertex screening under various setting. We generate 100 graphs from 2 classes, that is $A|Y = y \sim IER(P^y)$ with $y \in \{0, 1\}$ and

$$P^y = \begin{bmatrix} p^y \times \mathbf{1}_{20\times 20} & 0.2 \times \mathbf{1}_{20\times 180} \\ 0.2 \times \mathbf{1}_{180\times 20} & 0.3 \times \mathbf{1}_{180\times 180} \end{bmatrix},$$

where $p^0 = 0.3$ and $p^1 = 0.4$. Based on this data generation scheme, each graph has 200 vertices with the first 20 vertices being the signal vertices. Note that it is also equivalent to generating graphs from two Stochastic Block models [4], where the vertices in the first block are signal vertices.

We carry out the one-time screening using Dcorr and MGC, iterative screening (ItDcorr and ItMGC) with δ being 0.5 and 0.05 respectively. As the true signal subgraph size is 20, all the screening methods are required to return the estimated signal subgraph with 20 vertices. For comparison, screening with canonical correlation analysis (CCA) [81] and RV coefficient (RV) [82] are also included. We repeat the data generation and screening 100 times. Figure 3.1 shows the Receiver operating characteristic (ROC) [83] of one repeat. Due to overlap, only results of screening using MGC is shown. Table 3.1 reports the area under the curve (AUC) [84] for all methods along with the running time. We observe that Dcorr and MGC work

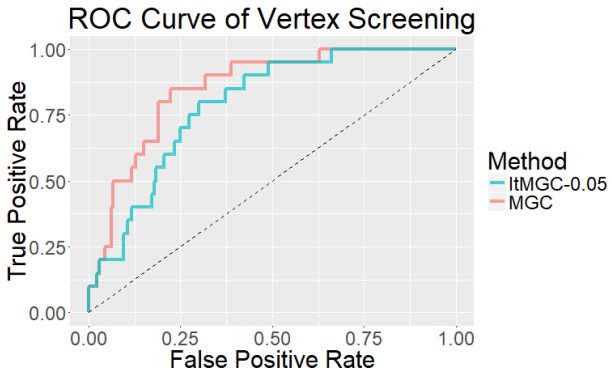

Figure 3.1: Receiver operating characteristic of two vertex screening procedures. The graphs are generated as described in Section 4.1. Iterative vertex screening is better than one-time vertex screening.

much better than CCA. In fact, RV is also worse than Dcorr or MGC, when they are compared using a paired test across 100 repeats. Furthermore, iterative screening at $\delta = 0.5$ improves the performance further with a slight increase of running time, while iterative screening at $\delta = 0.05$ improves marginally at the cost of much higher running time.

3.5.2 Simulation: Graph Classification under IER

In this experiment, we investigate the effects of signal subgraph extraction for later classification. We consider a 3-class classification problem, that is $A|Y = y \sim IER(P^y)$ with $y \in \{0, 1, 2\}$ and

$$P^y = \begin{bmatrix} p^y \times \mathbf{1}_{20\times 20} & 0.2 \times \mathbf{1}_{20\times 180} \\ 0.2 \times \mathbf{1}_{180\times 20} & 0.3 \times \mathbf{1}_{180\times 180} \end{bmatrix},$$

Method	AUC	Time (sec)
ItDcorr-0.05	0.8705 (0.0113)	18.50 (1.35)
ItDcorr-0.50	0.8655 (0.0094)	2.03 (0.17)
ItMGC-0.05	0.8720 (0.0122)	967.42 (17.73)
ItMGC-0.50	0.8625 (0.0106)	120.16 (7.32)
Dcorr	0.8554 (0.0056)	1.23 (0.22)
MGC	0.8555 (0.0057)	38.44 (1.720)
RV	0.8506 (0.0077)	2.12 (0.10)
CCA	0.5353 (0.0080)	0.92 (0.04)

Table 3.1: The mean and standard error of AUC and running time of the eight vertex screening approaches across 100 repeats. Iterative vertex screening has better AUC, but takes longer to run.

where

$$p^y = \begin{cases} 0.3 & \text{if } y = 0, \\ 0.4 & \text{if } y = 1, \\ 0.5 & \text{if } y = 2. \end{cases}$$

Based on this data generation scheme, each graph has 200 vertices with the first 20 vertices being the signal vertices. We consider the classification performance of 5 classifiers; specifically, $L(g^*)$, $L(g_V)$, $L(g_S)$, $L(g_{\hat{S}})$, where $\hat{S}$ is estimated using Dcorr ($\hat{S}$-Dcorr), MGC ($\hat{S}$-MGC), or iterative Dcorr ($\hat{S}$-Iter). Note that here $L(g^*)$ and

$L(g_S)$ are shown for demonstration purpose and are the best possible error rate one can accomplish in theory. Stopping at $|S| = 20$, the classification performance and false positive rate in identifying signal vertices are shown in Figure 3.2. Prediction based on the signal subgraph estimated by screening has a clear advantage over prediction based on the whole graph. Furthermore, screening with MGC is better than screening with Dcorr, and iterative screening is better than non-iterative screening. Since the interpretation are similar as the first experiment, we do not include CCA, or RV performance here. Note that screening is able to recover the signal vertices perfectly when $m > 300$. However, due to estimation error in $\hat{P}^y$, prediction error $L(g_{\hat{S}})$ of plugin classifier based on the subgraph still is not as good as Bayes optimal error $L(g^*)$.

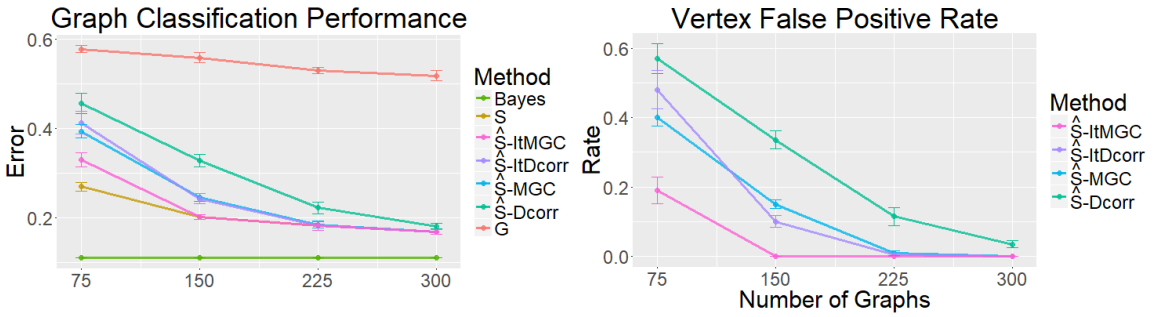

Figure 3.2: The graph classification error of 7 approaches with their standard errors are shown at the top panel. We generate graphs from 3 inhomogeneous Erdos-Renyi model as described in Section 4.2, then apply 7 approaches to classify these graphs: Bayes plug-in on G (G), Bayes plug-in on $G[S]$ (S), Bayes optimal classifier (Bayes), Bayes plug-in on $G[\hat{S}]$ with $\hat{S}$ estimated by Dcorr or MGC ($\hat{S}$-Dcorr, $\hat{S}$-MGC), and Bayes plug-in on $G[\hat{S}]$ with $\hat{S}$ estimated by iterative Dcorr or MGC ($\hat{S}$-ItDcorr, $\hat{S}$-ItMGC). The plot at top shows graph prediction error using these 7 approaches. The plot at bottom shows the false positive rate in identifying signal vertices using 4 signal subgraph estimation approaches. The classifiers based on estimated signal subgraph have significantly better classification performance compared to classifiers based on the whole graph, and are close to Bayes optimal classifier when given 300 graphs. Furthermore, screening with MGC is better than screening with Dcorr and iterative screening is better than non-iterative screening, in terms of both graph classification and signal subgraph estimation.

If we assume $|S|$ is unknown and estimate the size of signal subgraph via maximizing the distance correlation between the subgraph and label, the resulting subgraph

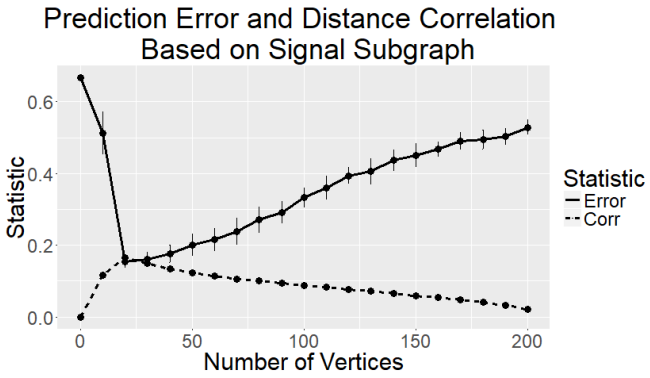

Figure 3.3: The cross validation error and distance correlation with their standard error based on different size of subgraph, produced by the iterative Dcorr screening algorithm. The optimal size of signal subgraph implied by these two statistics are both 20.

at the maximal correlation statistic coincides with the subgraph of the best classification error. Figure 3.3 shows the classification error and distance correlation with their standard error for different size of subgraph using iterative screening. Given 300 graphs, finding the best prediction error or maximizing the distance correlation between the subgraph and label yield the same estimate of the size of signal subgraph. However, calculating the distance correlation between subgraph and label is computationally cheaper than computing the prediction error. This point will be demonstrated in real data experiments as well.

3.5.3 Real Data: Site and Sex Prediction With Human Functional Magnetic Resonance Images

We consider the task of predicting the site and sex based on functional magnetic resonance image (fMRI) graphs [85]. Two datasets used are SWU4 [86] and HNU1 [87], which have 467 and 300 samples respectively. Each sample is an fMRI scan registered to the MNI152 template using the Desikan altas, which has 70 regions [55]. We first merge two data sets and then try to predict the site a sample come from. In addition, we try to predict the sex of subject based the fMRI scan.

There are multiple scans (samples) per subject; as a consequence, we carry out a leave-one-subject-out signal subgraph estimation and prediction procedure. To estimate the signal subgraph for site and sex, we first apply iterative vertex screening with samples from one subject left out. Next, we apply 11-Nearest Neighbor to predict the site and sex of the left out samples. The prediction is based on the estimated signal subgraph. This procedure is repeated for all subjects and we compute the leave-one-subject-out screening and prediction error [48, 50].

The prediction error and distance correlation between the subgraph and label with varying size of signal subgraph are shown in Figure 3.4. Predicting randomly or using no graph at all will have error rate 0.39 and 0.50 for site and sex prediction respectively, which is shown in the Figure 3.4 with the number of vertices at 0.

Sex prediction has prediction error around 0.5 and correlation small. However, the site prediction has achieved high accuracy with classification error less than 0.1 when predicting using a signal subgraph with around $10-30$ vertices. The best performance is achieved by the signal subgraph with 30 vertices.

As in the simulation experiment, we further utilize the minimum prediction error and the maximum correlation between the subgraph and label to estimate the size of the signal subgraph. In addition, we order the correlations between vertices and the label to find a gap between signal vertices and insignal vertices. The estimated size of signal subgraph to predict site using the three methods is 30, 25 and 27 respectively. The estimated size of signal subgraph to predict sex is 45, 10 and 12. The three different methods yield similar error rate, which validates that the stopping criterion in the iterative screening algorithm works well.

We further apply the iterative vertex screening to all samples and pick the top 30 signal vertices with large distance-based correlations. It turns out that these 30 vertices are matched across left and right hemispheres. If we consider the 35 paired regions in Desikan atlas, we can group the pairs according to whether both regions are among the top 30 signal vertices or not. Table 3.2 shows the result. The regions with large distance-based correlations are significantly matched based on Chi-square test with a p-value of 0.0020. The 11 left-right hemisphere matched regions are caudal anterior cingulate, corpus callosum, cuneus, fusiform, lateral occipital, lingual, parsorbitalis, precuneus, rostral anterior cingulate, rostral middle frontal gyrus, and

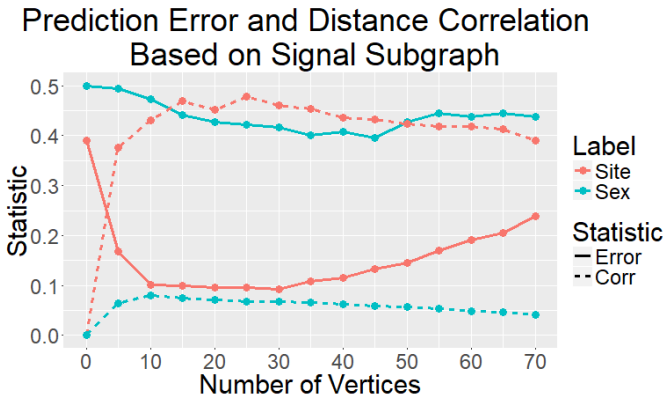

Figure 3.4: Leave-one-subject-out prediction error and distance correlation based on different size of the signal subgraph. Two studies SWU4 and HNU1 are merged into one data set. We carry out a leave one subject out, screening and prediction procedure to predict sex or site of the left-out sample. The prediction error with different size of signal subgraph is represented by the solid lines. When predicting with no graph or predicting all samples randomly, the prediction error is 0.39 and 0.50 for site and sex respectively, which are shown with the number of vertices being 0. The distance correlation between the subgraph and two covariates is represented by dashed lines. Sex prediction performs poorly in this setting with prediction error being around 0.5 and correlation small. The site prediction has high accuracy with the best performance achieved when the subgraph has $10 - 30$ signal vertices.

superior frontal gyrus. They are shown in Figure 3.5.

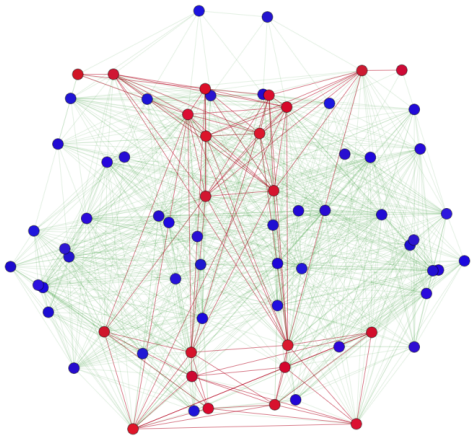

Figure 3.5: Desikan atlas with highlighted brain regions which are significantly dependent on site. The 11 matched brain regions as found in Table 3.2 are highlighted in red. They are spatially adjacent.

Number of Pairs	Right-Large	Right-Small
Left-Large	11	1
Left-Small	7	16

Table 3.2: The number of left-right hemisphere matched regions with large or small distance-based correlations.

3.5.4 Real Data: Sex Difference in Mouse Brain with Magnetic Resonance Diffusion Tensor Imaging

Structural magnetic resonance imaging has provided insight into the genetic basis of mouse brain variability, by examining the relationship between volume covariance

and genotypes [88]. Using high resolution diffusion tensor imaging and tractography we can now examine the underlying bases for structural connectivity patterns [89], in relationship with genotype and sex. 55 mouse brains (of pooled genotypes) were scanned and registered into the space of a minimum deformation template, aligned to Waxholm space [90]. The atlas labels were propagated onto the template, and subsequently onto each individual brain using ANTs [91]. DSI Studio [92] was used to estimate tract based structural connectivity for each brain. Each connectome was represented as a graph with 332 vertices, 166 per hemisphere. Out of 55 mice, 32 of them are male and 23 are female. Again, we carry out a leave-one-out iterative vertex screening to estimate the signal subgraph. Then, the left-out sample is predicted based on the estimated signal subgraph using a 9 nearest neighbor classifier. The prediction result and distance correlation based on various size of signal subgraph are shown in Figure 3.6. Due to the small sample size, the prediction error becomes more volatile. Furthermore, correlation becomes monotone decreasing probably because of over-fitting, since the sample size is small and graph size is large. The iterative screening algorithm yields a signal subgraph of size 10, which is very close to the best possible leave-one-out error at size 20. The top ranked nodes include a thalamic component and the periaqueductal gray, which are important in driving the sexually dimorphic mouse brain development [93] [94].

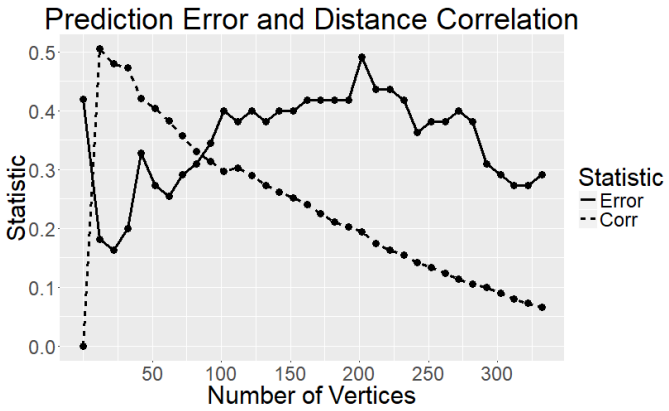

Figure 3.6: Mouse sex prediction and distance correlation based on different size of signal subgraph. Left one out iterative vertex screening and prediction is carried on the mouse brain dataset. Signal subgraph with 10 or 20 vertices yield the best performance.

3.6 Discussion

In summary, we developed an iterative vertex screening methodology to estimate the signal subgraph of interest. The data experiments and theories offer strong evidence that our screening algorithm estimates the signal subgraph effectively and accurately, which leads to better performance for subsequent inference task. Our approach is intimately related to classical feature screening under linear models [7,67,68]. However, instead of Pearson correlation, we utilize distance correlation [69] and multiscale generalized correlation [72] to measure the dependency between the vertex and response variable. This approach allows the possibility to estimate signal subgraph based on non-scalar response variable or response with non-Euclidean metric. Our method also naturally applies to topological or spectral features of vertices [4, 78],

which have been shown to be effective in analyzing fMRI data [95, 96]. Thus our method provides a general and viable tool for supervised learning problems on graphs.

3.7 Proofs

Proof of Lemma 3.4.2 By definition of distance covariance,

$$\begin{aligned}
&Dcov(X, Y) \\
&= \frac{1}{c_{p+r}c_q} \int_{s,t} \frac{|\phi_{X,Y}(s,t) - \phi_X(s)\phi_Y(t)|^2}{\|s\|^{1+p+r}\|t\|^{1+q}} \\
&= \frac{1}{c_{p+r}c_q} \int_{s_p,s_r,t} \frac{|\phi_{X^*,Y}(s_p,t) - \phi_{X^*}(s_p)\phi_Y(t)|^2|\phi_Z(s_r)|^2}{\|[s_p,s_r]\|^{1+p+r}\|t\|^{1+q}} \\
&\leq \frac{1}{c_{p+r}c_q} \int_{s_p,s_r,t} \frac{|\phi_{X^*,Y}(s_p,t) - \phi_{X^*}(s_p)\phi_Y(t)|^2}{\|[s_p,s_r]\|^{1+p+r}\|t\|^{1+q}} \\
&= Dcov([X^*, \vec{\mathbf{0}})], Y),
\end{aligned}$$

where the inequality holds because $|\phi_Z(s_r)| \leq 1$.

Using the alternative definition of distance covariance, we have

$$\begin{aligned}
&Dcov([X^*, \vec{\mathbf{0}})], Y) \\
&= \mathbb{E}(\|[X^*, \vec{\mathbf{0}})] - [X^{*\prime}, \vec{\mathbf{0}})]\|\|Y - Y'\|) + \mathbb{E}(\|[X^*, \vec{\mathbf{0}})] - [X^{*\prime}, \vec{\mathbf{0}})]\|)E(\|Y - Y'\|) \\
&\quad - 2\mathbb{E}(\|[X^*, \vec{\mathbf{0}})] - [X^{*\prime}, \vec{\mathbf{0}})]\|\|Y - Y''\|) \\
&= \mathbb{E}(\|X^* - X^{*\prime}\|\|Y - Y'\|) + \mathbb{E}(\|\|X^* - X^{*\prime}\|\|)E(\|Y - Y'\|) \\
&\quad - 2\mathbb{E}(\|\|X^* - X^{*\prime}\|\|\|Y - Y''\|) \\
&= Dcov(X^*, Y).
\end{aligned}$$

This concludes $Dcov(X^*, Y) \geq Dcov(X, Y)$.

Proof of Corollary 3.4.3 By definition of distance correlation,

$$Dcorr(X_r, Y) = \frac{Dcov(X_r, Y)}{\sqrt{Dcov(X_r, X_r)Dcov(Y, Y)}}.$$

We first show that $Dcov(X_r, Y)$ converges to 0 as the number of noise dimension r goes to infinity. By definition,

$$\begin{aligned}
Dcov(X_r, Y) &= \mathbb{E}(\|X_r - X_r'\|\|Y - Y'\|)+ \\
&\quad \mathbb{E}(\|X_r - X_r'\|)E(\|Y - Y'\|) - 2\mathbb{E}(\|X_r - X_r'\|\|Y - Y''\|) \\
&= \mathbb{E}(\|X_r - X_r'\|\|Y - Y'\|) - \mathbb{E}(\|X_r - X_r'\|)\mathbb{E}(\|Y - Y'\|)+ \\
&\quad 2\mathbb{E}(\|X_r - X_r'\|)E(\|Y - Y'\|) - 2\mathbb{E}(\|X_r - X_r'\|\|Y - Y''\|) \\
&= Cov(\|X_r - X_r'\|, \|Y - Y'\|) - 2Cov(\|X_r - X_r'\|, \|Y - Y''\|)
\end{aligned}$$

Let us look at $Cov(\|X_r - X_r'\|, \|Y - Y'\|)$.

$$\begin{aligned}
&Cov(\|X_r - X_r'\|, \|Y - Y'\|) \\
&= \mathbb{E}(\|X_r - X_r'\|\|Y - Y'\|) - \mathbb{E}(\|X_r - X_r'\|)\mathbb{E}(\|Y - Y'\|) \\
&\leq \mathbb{E}(\|X_r - X_r'\|\|Y - Y'\|) - \mathbb{E}(\|Z_r - Z_r'\|)\mathbb{E}(\|Y - Y'\|) \\
&= \mathbb{E}(\|X_r - X_r'\|\|Y - Y'\|) - \mathbb{E}(\|Z_r - Z_r'\|\|Y - Y'\|) \\
&= \mathbb{E}(\|X_r - X_r'\|\|Y - Y'\| - \|Z_r - Z_r'\|\|Y - Y'\|)
\end{aligned}$$

Let Z be the first entry of Z_r, and we define

$$\mu = \mathbb{E}((Z - Z')^2),$$

$$\sigma^2 = Var((Z - Z')^2),$$

$$\gamma^2 = Cov((Z - Z')^2, (Z - Z'')^2).$$

Applying the Taylor expansion to $\sqrt{\frac{\|Z_r - Z_r'\|^2}{r}}$ at μ, we have

$$\begin{aligned}\frac{\|Z_r - Z_r'\|}{\sqrt{r}} =& \mu^{\frac{1}{2}} + \frac{1}{2}\mu^{-\frac{1}{2}}(\frac{\|Z_r - Z_r'\|^2}{r} - \mu) \\ & - \frac{1}{8}(\frac{\|Z_r - Z_r'\|^2}{r} - \mu)^2 + O(r^{-\frac{3}{2}}).\end{aligned}$$

Similarly,

$$\begin{aligned}\frac{\|X_r - X_r'\|}{\sqrt{r}} =& \mu^{\frac{1}{2}} + \frac{1}{2}\mu^{-\frac{1}{2}}(\frac{\|X_r - X_r'\|^2}{r} - \mu) \\ & - \frac{1}{8}(\frac{\|X_r - X_r'\|^2}{r} - \mu)^2 + O(r^{-\frac{3}{2}}).\end{aligned}$$

Therefore,

$$\|X_r - X_r'\| - \|Z_r - Z_r'\| = O(r^{-\frac{1}{2}}).$$

As a consequence,

$$\begin{aligned}& Cov(\|X_r - X_r'\|, \|Y - Y'\|) \\ & \leq \mathbb{E}(\|X_r - X_r'\|\|Y - Y'\| - \|Z_r - Z_r'\|\|Y - Y'\|) \\ & = O(r^{-\frac{1}{2}}).\end{aligned}$$

We can also derive a lower bound:

$$
\begin{aligned}
&Cov(\|X_r - X_r'\|, \|Y - Y'\|) \\
&= \mathbb{E}(\|X_r - X_r'\|\|Y - Y'\|) - \mathbb{E}(\|X_r - X_r'\|)\mathbb{E}(\|Y - Y'\|) \\
&\geq \mathbb{E}(\|Z_r - Z_r'\|\|Y - Y'\|) - \mathbb{E}(\|X_r - X_r'\|)\mathbb{E}(\|Y - Y'\|) \\
&= \mathbb{E}(\|Z_r - Z_r'\|)E(\|Y - Y'\|) - \mathbb{E}(\|X_r - X_r'\|)E(\|Y - Y'\|) \\
&= \mathbb{E}(\|Z_r - Z_r'\| - \|X_r - X_r'\|)\mathbb{E}(\|Y - Y'\|) \\
&= O(r^{-\frac{1}{2}})
\end{aligned}
$$

Similarly, we can show that

$$Cov(\|X_r - X_r'\|, \|Y - Y''\|) \to 0.$$

This proves $Dcov(X_r, Y) \to 0$.

Next, we demonstrate that $Dcov(X_r, X_r)$ is non-vanishing. Again, we need to analyze $Cov(\|X_r - X_r'\|, \|X_r - X_r'\|)$ and $Cov(\|X_r - X_r'\|, \|X_r - X_r''\|)$.

$$
\begin{aligned}
&Cov(\|X_r - X_r'\|, \|X_r - X_r'\|) \\
&= \mathbb{E}(\|X_r - X_r'\|^2) - \mathbb{E}^2(\|X_r - X_r'\|) \\
&= \mathbb{E}(\|X^* - X^{*\prime}\|^2) + r\mu - \mathbb{E}^2(\|X_r - X_r'\|) \\
&= \mathbb{E}(\|X^* - X^{*\prime}\|^2) + r\mu - r(\mu^{\frac{1}{2}} + \frac{1}{2}\mu^{-\frac{1}{2}}\frac{\mathbb{E}(\|X^* - X^{*\prime}\|^2)}{r} - \frac{1}{8}\mu^{-\frac{3}{2}}\frac{\sigma^2}{r} + O(r^{-\frac{3}{2}}))^2 \\
&= \frac{1}{4}\mu^{-1}\sigma^2 + O(r^{-1}).
\end{aligned}
$$

Use the similar Taylor expansion technique, we can show

$$Cov(\|X_r - X_r'\|, \|X_r - X_r''\|) = \frac{1}{8}\mu^{-1}\gamma^2 + O(r^{-1}).$$

As long as Z is non-degenerate, $\sigma^2 - \gamma^2 > 0$. This shows

$$\lim_{r\to\infty} Dcov(X_r, X_r) = \frac{1}{4}\mu^{-1}(\sigma^2 - \gamma^2) > 0$$

. Moreover, $Dcov(Y, Y)$ is always a fixed positive number for non-degenerate Y, thus we conclude

$$\lim_{r\to\infty} Dcorr(X_r, Y) = 0.$$

Proof of Theorem 3.4.4 It suffices to show the Bayes plug-in density $\mathcal{L}(A; \hat{P}^y)$ is close to the true density $\mathcal{L}(A; P^y)$ with high probability. We will assume $\pi_y \geq \alpha$ for some fixed $\alpha > 0$. Applying Hoeffding's Equality to $\hat{\pi}_y$ [97],

$$\mathbb{P}(|\hat{\pi}_y - \pi_y| < \epsilon_1) \geq 1 - 2\exp(-2m\epsilon_1^2).$$

By choosing ϵ_1 small enough such that $\hat{\pi}_y > \frac{\alpha}{2}$, and applying Hoeffding's Equality to $\hat{P}^y_{ij}$, it follows that

$$\mathbb{P}(|\hat{P}^y_{ij} - P^y_{ij}| < \epsilon_2) \geq 1 - 2\exp(-m\alpha\epsilon_2^2).$$

If $|\hat{\pi}_y - \pi_y| < \epsilon_1$ and $|\hat{P}^y_{ij} - P^y_{ij}| < \epsilon_2$, for any adjacency matrix A:

$$\begin{aligned}
|\pi_y\mathcal{L}(A; P^y) - \hat{\pi}_y\mathcal{L}(A; \hat{P}^y)| &\leq |\pi_y\mathcal{L}(A; \hat{P}^y) - \hat{\pi}_y\mathcal{L}(A; \hat{P}^y)| + |\pi_y\mathcal{L}(A; P^y) - \pi_y\mathcal{L}(A; \hat{P}^y)| \\
&< \epsilon_1 + |\pi_y\mathcal{L}(A; P^y) - \pi_y\mathcal{L}(A; \hat{P}^y)| \\
&< \epsilon_1 + |\mathcal{L}(A; P^y) - \mathcal{L}(A; \hat{P}^y)| \\
&< \epsilon_1 + e\epsilon_2.
\end{aligned}$$

The last inequality follows from recursively applying the technique used in the first inequality and the fact that $|\hat{P}^y_{ij} - P^y_{ij}| < \epsilon_2$. As a consequence, conditioned on $\hat{\pi}_y$ and $\hat{P}^y$ satisfy the Hoeffding's inequality,

$$\mathbb{E}_A(|\pi_0\mathcal{L}(A; P^0) - \hat{\pi}_0\mathcal{L}(A; \hat{P}^0)| + |\pi_1\mathcal{L}(A; P^1) - \hat{\pi}_1\mathcal{L}(A; \hat{P}^1)|) \leq 2(\epsilon_1 + e\epsilon_2).$$

Setting $2(\epsilon_1 + e\epsilon_2) = \epsilon$ and $2\epsilon_1^2 = \alpha\epsilon_2^2$, we have $\epsilon_2 = \frac{\epsilon}{2e+\sqrt{2\alpha}}$. Then, apply Theorem 2.3 in [77] yields

$$\mathbb{P}(L(g_V) - L(g^*) < \epsilon) \geq 1 - 2(e+1)\exp\left(\frac{-m\alpha\epsilon^2}{(2e+\sqrt{2\alpha})^2}\right).$$

Alternatively, setting $\eta = 2(e+1)\exp(\frac{-m\alpha\epsilon^2}{(2e+\sqrt{2\alpha})^2})$ yields that with probability at least $1-\eta$, it holds that

$$L(g_V) - L(g^*) \leq (2e+\sqrt{2\alpha})\sqrt{\frac{\log(\frac{2(e+1)}{\eta})}{m\alpha}}.$$

Proof of Corollary 3.4.5 Following Theorem 3.2, we have

$$\begin{aligned}
\mathbb{E}(L(g_V)) - L(g^*) &= \mathbb{E}(L(g_V) - L(g^*)) \\
&< \epsilon\mathbb{I}\{L(g_V) - L(g^*) < \epsilon\} + \mathbb{I}\{L(g_V) - L(g^*) \geq \epsilon\} \\
&< \epsilon + 2(e+1)\exp\left(\frac{-m\alpha\epsilon^2}{(2e+\sqrt{2\alpha})^2}\right).
\end{aligned}$$

Chapter 4

Optimal Decisions for Discovery Science via Maximizing Discriminability

4.1 Introduction

In this era of big data, many scientific, government, and corporate groups are collecting and processing massive data sets [8, 9]. To obtain optimal quantitative answers to any inquiry about data requires making two decisions: (i) how should the data be collected?, and (ii) how should the data be processed?. When the downstream inference task is specified, a priori, we can collect and process data to optimize the performance of task [48, 98]. However, recently, across industry, governmental, and

academic settings, certain data sets become benchmark or reference data sets. Such data sets are then used for a wide variety of different inferential problems. Collecting and processing these data sets requires massive institutional investments, and choices related to questions(i) and (ii) above have dramatic effects on all subsequent analyses. Optimally addressing experimental design decisions can yield significant savings in both the financial and human costs, and also improve accuracy of analytical results [10–12]. Therefore, a theoretical framework to enable investigators to select from a set of possible design decisions in the absence of an explicit task or for multiple tasks could reap great rewards.

This framework should provide a measure of consistency of data collection and processing, which is intuitive to understand and easy to implement. It should be non-parametric and robust; therefore, it is ready to be applied under a variety of settings. It should not be computationally expensive and can be applied to large data sets. Furthermore, it should be simple and unified; as a consequence, we can easily compare it across data sets. Lastly, theories and real data experiments should provide solid support to use this measure to guide data collection and processing.

To this end, we have proposed and developed a formal definition of discriminability to guide data collection and processing. Discriminability is a non-parametric statistical property of a joint distribution in a hierarchical model, which can be used to differentiate between classes of objects. We prove that discriminability (which may be more aptly called reliability), provides a lower bound on predictive accuracy for

any downstream inference task, even if we have never seen the covariates to predict in the processing. We then design an estimator of discriminability computed from test-retest data set, demonstrate that it is unbiased, and derive our estimators asymptotic distribution. Furthermore, one sample and two sample tests for discriminability are developed. These tests determines the statistical significance of hypothesis of interest based on the discriminability estimator.

Numerical simulations are conducted to demonstrate the basic property of our discriminability estimator and tests in a variety of settings. Then, we apply our approach to choose amongst a set of choices one must make when designing a neuroimaging study to investigate functional connectomics [99, 100]. We start by finding the most discriminable threshold for converting correlation connectome matrices into binary graphs. Indeed, consistent with our theoretical and simulated results, maximizing the discriminability also maximizes performances on a suite of different downstream inference tasks. We then ask about a series of pre-processing steps: should one motion correct or not, should one perform frequency filtering or not, and should one implement global signal regression or not, etc. We determine the optimal choice for each pre-processing steps, and find the most discriminable pipelines amongst 64 pre-processing pipelines.

Thus, in total, our discriminability analysis is a powerful tool for making decisions about how to collect and process data sets designed for discovery science. In the next section, we discuss previous work on measuring reliability or reproducibility. In

Section 4.3, we present the discriminability and its estimator. In Section 4.4, we demonstrate theoretical properties of discriminability. In Section 4.5, we illustrate the utility of our discriminability framework through experiments with emphasis on processing human brain networks.

4.2 Related Work

There are some successful attempts to quantify reliability or reproducibility in neuroimaging studies [101–109]. We are going to review a subset of them which is related to our approach.

- Intraclass correlation coefficient (ICC) is introduced to measure consistency or reproducibility of scalar quantitative measurements [101]. In neuroimaging, people attempt to extract one or a few summary scalar statistics from each image and then evaluate the ICC of the statistics [104, 105]. They report moderate-to-high test-retest reliability for different statistics. The problem with this approach is that the summary statistics may not be representative. Also, there is no principled approach to average over multiple ICCs.

- Image intraclass correlation coefficient (I2C2) is proposed by Shou et al. to measure reliability [106]. It generalizes classic intraclass coefficient to high dimensional observations. It computes reliability estimates based on the traces of within subject and across subject covariance matrix. It relies on the assumption

that noise is additive and observations lies in the space equipped with Euclidean distance. As a consequence, it is not suitable to apply to more general settings.

- Graphical intraclass correlation coefficient (GICC) is a reproducibility measure proposed by Yue et al. [107]. It is designed specifically for the case when data of interest are binary graphs. It takes a parametric approach by first assuming a probit link function and estimating latent edge feature vectors. Then, it computes GICC based on variation of latent edge feature vectors. In practice, its assumptions is hard to justify and it is computationally expensive to estimate latent features for graphs of moderate size.

- Correspondence curve is introduced to study reproducibility of signals [109]. It first ranks all the signals by a scalar score within each replicates, and then the proportion of signals which ranked among top percentile of both replicates is computed. It generalizes Spearman's rank correlation coefficient and can be used to detect irreproducible signals. In our studies, we are interested in reproducibility of measurements instead of signals and the measurements are vectors or matrices, which makes this approach not immediately applicable.

- Distance components (DISCO) is proposed by Rizzo and Székely as a measure of dispersion [103]. It computes one distance statistic for multiple empirical distributions based on pairwise distances between samples. It can also be used to test the hypothesis that whether multiple sets of samples are drawn from the

same distribution or not. Our approach is similar to DISCO in the sense that we all rely on pairwise distance matrix. However, DISCO is designed for testing which requires a fixed number of subjects and a large amount of measurements from each subject. In our studies, we only have a few measurements from each subject which makes DISCO hard to apply.

- NPAIRS data analysis framework is proposed in [102]. It takes a resampling approach by splitting data in half. After performing a series of dimension reduction on the data, a label is predicted using Gaussian mixture model. Then, correlation between all pairs of spatially aligned voxels is calculated. A signal-to-noise ratio measure is computed based on the correlation.

- A statistics called estimation stability (ES) is proposed in [108]. It is similar to a variance estimator computed through delete-d jacknife resampling. It is applied to smoothing parameter selection in Lasso and is shown to obtain a great reduction of model without sacrificing prediction performance in a task fMRI study.

4.3 Discriminabilty

4.3.1 Discriminability to Guide Processing

In this section, we present the discriminability as a framework to guide processing. Discriminability measures the overall consistency and differentiability of observations. For example, if a subject is measured twice under the same conditions, two observations should be close to each other given the measure is consistent. In addition, one should be able to tell these two observations come from the same subject when compared to observations from other subjects given the measure is differentiable. We quantify this idea of consistency and differentiability through discriminability.

To formalize the definition of discriminability, consider the following generative process. For each sample i, there exists some true physical property $\mathbf{v}_i$. Unfortunately, we do not get directly to observe $\mathbf{v}_i$, rather, we measure it with some device, that transforms the truth from $\mathbf{v}_i$ to $\mathbf{w}_i$ via $f_{\boldsymbol{\phi}}$. The parameter $\boldsymbol{\phi} \in \boldsymbol{\Phi}$ characterizes all options in the measurement, including, for example, which scanner to use, which resolution, the number of images, sampling rate, etc. The output of $f_{\boldsymbol{\phi}}$ is the "raw" observation data $\mathbf{w}_i$, but it is corrupt in various ways, including movement or intensity artifacts introduced by the measurement process. Therefore, rather than operating directly on $\mathbf{w}_i$, we intentionally "pre-process" the data, in an effort to remove a number of nuisance variables. This pre-processing procedure further transforms the data from $\mathbf{w}_i$ to $\mathbf{x}_i$ via $g_{\boldsymbol{\psi}}$. The parameter $\boldsymbol{\psi} \in \boldsymbol{\Psi}$ indexes all pre-processing options.

In neuroimaing, these options may include whether to perform motion correction, which motion correction, deconvolution, etc. More specifically, the entire code base, including dependencies, and even the hardware the pre-processing is running on, could count as $\boldsymbol{\psi}$. For brevity, we define $\mathbf{x}_i := g_{\boldsymbol{\psi}}\big(f_{\boldsymbol{\phi}}(\mathbf{v}_i)\big)$. We should notice that $g_{\boldsymbol{\psi}}$ and $f_{\boldsymbol{\phi}}$ by their natures are random functions which means even if we measure the same physical property $\mathbf{v}_i$ twice the results could be different.

Let i denote the sample's unique *identity* (hereafter, referred to as the *subject*) and t denote the trial number. Thus, there is a single $\mathbf{v}_i$ for subject i, but we have $\mathbf{x}_{i,t}$, which is the t^{th} trial, implicitly also a function of $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$, which encodes all the details of the measurement and pre-processing. If both $g_{\boldsymbol{\psi}}$ and $f_{\boldsymbol{\phi}}$ together do not introduce too much noise, then we would expect that $\mathbf{x}_{i,t}$ and $\mathbf{x}_{i,t'}$ are *closer* to one another than either are to any other subject's measurement, $\mathbf{x}_{i',t''}$. Define δ to be a metric computing the distance between two measurements, $\delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$. Formally, we expect that $\delta(\mathbf{x}_{i,t}, \mathbf{x}_{i,t'}) < \delta(\mathbf{x}_{i,t}, \mathbf{x}_{i',t''})$, for most combinations of $i, i' \neq i, t, t' \neq t, t''$. For brevity, let $\delta_{i,t,t'} := \delta(\mathbf{x}_{i,t}, \mathbf{x}_{i,t'})$ and $\delta_{i,i',t,t''} := \delta(\mathbf{x}_{i,t}, \mathbf{x}_{i',t''})$. This intuition leads to our definition of discriminability:

$$D(\boldsymbol{\psi}, \boldsymbol{\phi}) = \mathbb{P}(\delta_{i,t,t'} < \delta_{i,i',t,t''}). \tag{4.1}$$

In words, discriminability is the probability that within subject distance is smaller than across subject distance. $D(\boldsymbol{\psi}, \boldsymbol{\phi})$ depends on three matters, namely measurement options $f_{\boldsymbol{\phi}}$, processing options $g_{\boldsymbol{\psi}}$ and the distribution of true physical property

$\mathbf{v}$. To understand the equation 4.1 better, we can expand it,

$$D(\boldsymbol{\psi}, \boldsymbol{\phi}) = \mathbb{E}(\mathbb{P}(\delta(g_{\boldsymbol{\psi}}(f_{\boldsymbol{\phi}}(\mathbf{v}_i))_t, g_{\boldsymbol{\psi}}(f_{\boldsymbol{\phi}}(\mathbf{v}_i))_{t'}) < \delta(g_{\boldsymbol{\psi}}(f_{\boldsymbol{\phi}}(\mathbf{v}_i))_t, g_{\boldsymbol{\psi}}(f_{\boldsymbol{\phi}}(\mathbf{v}_{i'}))_{t''})|\mathbf{v}_i, \mathbf{v}_{i'})). \tag{4.2}$$

The distribution of $\mathbf{v}$ is usually out of the control of researchers. However, we want to find the best data collection and processing options. To achieve this, we consider maximizing the discriminability of processed data, that is

$$\underset{\boldsymbol{\psi}\in\boldsymbol{\Psi},\boldsymbol{\phi}\in\boldsymbol{\Phi}}{\text{maximize}} \quad D(\boldsymbol{\psi}, \boldsymbol{\phi}). \tag{4.3}$$

It is often the case that data collection is also out of control of researchers, that is $\boldsymbol{\phi}$ is a fixed element in $\boldsymbol{\Phi}$. Therefore, we are only interested in finding the best processing routine encoded by $\boldsymbol{\psi}$. This is also the focus of this paper, since we do not have opportunity to make decisions on the data collection choices. In this case, we drop $\boldsymbol{\phi}$ in our notation and only maximize the discriminability over set $\boldsymbol{\Psi}$

$$\underset{\boldsymbol{\psi}\in\boldsymbol{\Psi}}{\text{maximize}} \quad D(\boldsymbol{\psi}) \tag{4.4}$$

This approach is intuitive and easy to understand. We will show that maximizing discriminability leads to good prediction performance. In addition, an unbiased estimator is designed to compute discriminability from test-retest data set. Furthermore, we have developed a one sample test procedure to determine whether there are subject specific information in the data, and a two sample test procedure to compare two processing pipelines. In experiment section, we will demonstrate the utility of discriminability through data experiments.

4.3.2 Discriminability Estimator

In real applications, distribution of $\mathbf{x}_{i,t}$ may never known to us; hence, it is not possible to compute discriminability $D(\boldsymbol{\psi})$ or D in short when there is no ambiguity in processing pipelines under consideration. However, samples $x_{i,t}$ are observed, and we can approximate true discriminability D using an estimator $\hat{D}$ which is a function of observed samples. For each pair of observations $x_{i,t}$ and $x_{i,t'}$ from subject i, we first define

$$\hat{D}_{i,t,t'} = \frac{\sum_{i' \neq i}^{n} \sum_{t''=1}^{s} \mathbb{I}\{\delta_{i,t,t'} < \delta_{i,i',t,t''}\}}{(n-1)s},$$

where $\mathbb{I}\{\cdot\}$ is the indicator function, n is the number of subjects, and s denotes the number of observations per subject. $\hat{D}_{i,t,t'}$ is the faction of observations from other subjects farther away from $x_{i,t}$ than $x_{i,t'}$. It approximates the probability that distances from observations of other subjects to the t^{th} observation of subject i is larger than the distance between t^{th} and t'^{th} trial of subject i. Then, we define the discriminability estimator $\hat{D}$ to be the mean of $\hat{D}_{i,t,t'}$ averaged over all pairs of observations from same subjects,

$$\hat{D} := \frac{\sum_{i=1}^{n} \sum_{t=1}^{s} \sum_{t' \neq t}^{s} \hat{D}_{i,t,t'}}{ns(s-1)}.$$

$\hat{D}$ is the sample discriminability which approximates discriminability or population discriminability. In Section 4.4, we discuss theoretical properties of discriminability. One important property is that the discriminability estimator $\hat{D}$ is unbiased and converges to D as the number of subjects n goes to infinity [63].

4.3.3 One Sample Test for Discriminability

In applications, we sometimes are interested in whether there is any subject specific information in the data. In other words, we want to know whether $\mathbf{x}_{i,t}$ is independent of $\mathbf{v}_i$. Formally, it is equivalent to test the hypothesis that $\mathbf{x}_{i,t}$ is independent of $\mathbf{v}_i$. If we fail to reject the hypothesis, it implies the measurement $\mathbf{x}_{i,t}$ reveals no information of true physical property $\mathbf{v}_i$. As a consequence, $\mathbf{x}_{i,t}$ is independent of any phenotype $\mathbf{y}_i$, and there is no hope in predicting $\mathbf{y}_i$ based on $\mathbf{x}_{i,t}$. If this is the case, the researchers should consider collecting more data or processing data differently. Since $\mathbf{v}_i$ is unobserved and $\mathbf{y}_i$ is unknown, a direct independence test is not applicable. We consider a test through discriminability. If measurements are independent of physical properties, $\mathbf{x}_{i,t}$ and $\mathbf{x}_{i',t'}$ should follow the same distribution. In this case, within subject distances should not differ across subject distances in distribution; therefore, discriminability should be 0.5. Conversely, we show in Lemma 4.4.5 that discriminability being 0.5 implies that $\mathbf{x}_{i,t}$ and $\mathbf{v}_i$ are independent. If we think any phenotype $\mathbf{y}_i$ is independent of measurement $\mathbf{x}_{i,t}$ conditioned on true physical property $\mathbf{v}_i$, an immediate consequence is that we can test the null hypothesis that measurements $\mathbf{x}_{i,t}$ are independent of any phenotype $\mathbf{y}_i$ through testing the hypothesis whether discriminability is 0.5.

$$H_0 : \mathbf{x} \perp \mathbf{y} \text{ , and } H_A : \mathbf{x} \not\perp \mathbf{y}.$$

We reject the null hypothesis above when there are strong evidences suggesting that $D > 0.5$.

We have two valid approaches to determine $D > 0.5$ through discriminability estimate $\hat{D}$. The first approach takes the advantage of the bound on variance of $\hat{D}$ which we derived in proving Lemma 4.4.4. Specifically, we show that the variance of $\hat{D}$ is less than or equal to $1/n$. Based on Chebyshev's inequality, we can derive a 95 percent confidence interval $(\hat{D} - \frac{2\sqrt{5}}{\sqrt{n}}, \hat{D} + \frac{2\sqrt{5}}{\sqrt{n}})$. If 0.5 lies in the confidence interval, we do not reject the null hypothesis; otherwise, we reject the null hypothesis. This approach is computationally simple; however, generally has small power due to the bound on variance is not tight. The second approach based on estimating a null distribution for $\hat{D}$ through permutation. In particular, we randomly permute subject labels for each trial and then estimate discriminability based on permuted labels. We repeat this procedure a large number of times and find the 95^{th} quantile of permuted discriminability estimates. If $\hat{D}$ is less than the 95^{th} quantile, we do not reject the null hypothesis; otherwise, we reject the null hypothesis. The details of this approach is described by Algorithm 5. This approach has larger power than the first approach, the only downside is that estimating discriminability for permuted samples takes sometime. In most applications, with less than a few hundred measurements, we recommend using the second approach.

Algorithm 5 One Sample Test for Discriminability

procedure TEST THE NULL HYPOTHESIS $\hat{D} = 0.5$
    Compute discriminability with true subject label $\hat{D}$
    for i=1 **do** n
        Compute discriminability with permuted subject label $\hat{D}_i$
    end for
    Compute p-value as the fraction of times that $\hat{D}_i > \hat{D}$
    Reject the null hypothesis if p-value is less than 0.05
end procedure

4.3.4 Two Sample Test for Discriminability

In many applications, we want to know whether one data processing pipeline $\boldsymbol{\psi}_1$ yields more discriminable data set than another pipeline $\boldsymbol{\psi}_2$. Based on the theory, by choosing the processing pipeline with larger discriminability, we can have a lower bound on Bayes prediction error. To achieve this, we consider testing the null hypothesis that two discriminabilities are equal:

$$H_0 : D(\boldsymbol{\psi}_1) = D(\boldsymbol{\psi}_2) \text{ , and } H_A : D(\boldsymbol{\psi}_1) > D(\boldsymbol{\psi}_2).$$

However, $D(\boldsymbol{\psi}_1)$ and $D(\boldsymbol{\psi}_2)$ are not known to us, we have to decide based on estimators $\hat{D}(\boldsymbol{\psi}_1)$ and $\hat{D}(\boldsymbol{\psi}_2)$. We have two valid approaches to test this. The first approach takes the advantage of the bound on variance of $\hat{D}$ which we derived in proving Lemma **??**. Specifically, we show that the variance of discriminability esti-

mate is bounded by $1/n$. Therefore, we can derive two confidence intervals centered at $\hat{D}(\boldsymbol{\psi}_1)$ and $\hat{D}(\boldsymbol{\psi}_2)$. Then, the null hypothesis is rejected if two confidence intervals does not overlap. Unfortunately, due to the fact that inequalities are not tight, this approach has very low power. For this method to work, the number of subjects n usually needs to be larger than a thousand. It is impractical for most of the data set. The second approach estimates null distribution of $\hat{D}(\boldsymbol{\psi}_1) - \hat{D}(\boldsymbol{\psi}_2)$ through bootstrapping. We can bootstrap copies of the original data set and compute discriminability on bootstrapped data set to approximate the null distribution. Specifically, let $\hat{D}^{(i)}(\boldsymbol{\psi}_j)$ denote the discriminability estimate for ith bootstrapped copy with data processed by pipeline j. If the null hypothesis is true, $\hat{D}(\boldsymbol{\psi}_1) - \hat{D}(\boldsymbol{\psi}_2)$ should have similar distribution as $\hat{D}^{(i)}(\boldsymbol{\psi}_j) - \hat{D}^{(i')}(\boldsymbol{\psi}_j)$. To bootstrap a copy of original data set, we need to make sure that the copy have the same number of subjects and number of measurements per subject as the original data set. To bootstrap measurements for a subject, we first randomly choose two subjects from original data sets, and then take a random convex linear combination of measurements of these two subjects. We keep repeating this step until the bootstrapped data set has the same number of subjects as the original data set, and discriminability $\hat{D}^{(i)}(\boldsymbol{\psi}_j)$ is estimated. To approximate the null distribution, a large number of bootstrapped discriminabilities are computed, and their pairwise differences $\hat{D}^{(i)}(\boldsymbol{\psi}_j) - \hat{D}^{(i')}(\boldsymbol{\psi}_j)$ are used to compute a p-value for $\hat{D}(\boldsymbol{\psi}_1) - \hat{D}(\boldsymbol{\psi}_2)$. We should notice that bootstrapped data tends to be less discriminable than the original data due to the fact bootstrapped subjects are closer

to each other. However, we only use differences in bootstrapped discriminability. The Algorithm 6 summarizes the steps to estimate p-value for testing $D(\boldsymbol{\psi}_1) = D(\boldsymbol{\psi}_2)$.

Algorithm 6 Two Sample Test for Discriminability

procedure TEST THE NULL HYPOTHESIS $D(\boldsymbol{\psi}_1) = D(\boldsymbol{\psi}_2)$

    Process the data set with pipelines $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$

    Compute $\hat{D}(\boldsymbol{\psi}_1)$ and $\hat{D}(\boldsymbol{\psi}_2)$

    for i in 1 through number of repeats **do**

        for j in 1 through number of subjects **do**

            Randomly select two subjects from data set

            Linearly combine measurements of these subjects

        end for

        Form two bootstrapped data sets processed by $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$

        Compute $\hat{D}^{(i)}(\boldsymbol{\psi}_1)$ and $\hat{D}^{(i)}(\boldsymbol{\psi}_2)$

    end for

    Compute pairwise differences $\hat{D}^{(i)}(\boldsymbol{\psi}_1) - \hat{D}^{(i')}(\boldsymbol{\psi}_1)$ and $\hat{D}^{(i)}(\boldsymbol{\psi}_2) - \hat{D}^{(i')}(\boldsymbol{\psi}_2)$

    Compute p-value as the fraction of times that $\hat{D}(\boldsymbol{\psi}_1) - \hat{D}(\boldsymbol{\psi}_2) > \hat{D}^{(i)}(\boldsymbol{\psi}_j) - \hat{D}^{(i')}(\boldsymbol{\psi}_j)$

    Reject the null hypothesis if p-value is less than 0.05.

end procedure

4.4 Theoretical Results

4.4.1 Optimizing Discriminability Optimizes Performance For Any Classification Task

Consider the situation that the downstream inference task is classification, that is in addition to $\mathbf{v}_i$, there are other properties of subject i of interest; we call all of them $\mathbf{y}_i \in \mathcal{Y}$. These may include, for example, the phenotype of the subject, including personality tests, demographic information, and genetic data. In this paper, we focus on binary classification problem that is $\mathcal{Y} = \{0, 1\}$. The goal of experimental design, in this context, is to choose $\boldsymbol{\psi} \in \boldsymbol{\Psi}$ to make good prediction of $\mathbf{y}_i$ based on observation $\mathbf{x}_i$. In this section, we will see that given two pipelines $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$, the one with larger discriminability is more likely to have better prediction performance.

To quantify the performance of our choice, we introduce some assumptions. First, assume that each $(\mathbf{v}_i, \mathbf{y}_i)$ pair is sampled independently and identically from some distribution, $(\mathbf{v}_i, \mathbf{y}_i) \overset{i.i.d.}{\sim} F_{V,Y}$. The goal is to predict the binary-valued *target* variable $\mathbf{y}_i$, using $\mathbf{x}_i$ as the *predictor* variables. Given a classifier $g : \mathcal{X} \to \mathcal{Y}$, to quantify the performance of classifier, we define the loss function $L(g)$ to be the probability of making error in prediction that is

$$L(g) = \mathbb{P}(g(\mathbf{x}_i) \neq \mathbf{y}_i).$$

It is known that the minimal prediction error $L^*(\mathbf{x}_i, \mathbf{y}_i)$ among all possible prediction

function is achieved by Bayes classifier [77]

$$L^*(\mathbf{x}_i, \mathbf{y}_i) := L(g^*),$$

where g^* is the Bayes classifier which is defined by

$$g^*(\mathbf{x}_i) := \underset{y \in \{0,1\}}{\operatorname{argmax}} \mathbb{P}(\mathbf{y}_i = y | \mathbf{x}_i).$$

Since $\mathbf{x}_i$ depends on pipeline $\boldsymbol{\psi}$, we denote the loss of pipeline $\boldsymbol{\psi}$ by $\ell(\boldsymbol{\psi})$ which is the Bayes prediction error of $(\mathbf{x}_i, \mathbf{y}_i)$,

$$\ell(\boldsymbol{\psi}) := L^*(\mathbf{x}_i, \mathbf{y}_i) = L^*(g_{\boldsymbol{\psi}}(f_{\boldsymbol{\phi}}(\mathbf{v}_i)), \mathbf{y}).$$

The next theorem shows the relationship between Bayes classification error and discriminability. Under assumptions that the noise is additive, we can prove theorem 1 which asserts that Bayes classification error is bounded by a decreasing function of discriminability.

Theorem 4.4.1 *There is a decreasing function h which only depends on $\mathbf{v}$ and $\mathbf{y}$, such that*

$$\ell(\boldsymbol{\psi}) \leq h(D(\boldsymbol{\psi})).$$

As a consequence, we expect the classification error to be small when the discriminability is large. An immediate corollary justifies using discriminability to select the optimal processing pipeline.

Corollary 4.4.2 *Given two processing pipelines $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$, suppose $\boldsymbol{\psi}_1$ is more discriminable than $\boldsymbol{\psi}_2$, that is $D(\boldsymbol{\psi}_1) > D(\boldsymbol{\psi}_2)$. If $\ell(\boldsymbol{\psi}_2) \geq h(D(\boldsymbol{\psi}_1))$, then*

$$\ell(\boldsymbol{\psi}_1) \leq \ell(\boldsymbol{\psi}_2).$$

Also, we must have

$$\ell(\boldsymbol{\psi}_1) \leq h(D(\boldsymbol{\psi}_2)).$$

It tells us for any distribution of $\mathbf{y}$, we have a tighter bound on Bayes error using the more discriminable pipeline. When choosing from two processing pipelines $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$, we should first compute $D(\boldsymbol{\psi}_1)$ and $D(\boldsymbol{\psi}_2)$. We then select the pipeline which yields larger discriminability to have lower bound on the Bayes classification error. This theorem justifies maximizing discriminability for subsequent classification tasks. Figure 4.1 summarizes the framework to find the optimal processing pipeline.

4.4.2 Discriminability and Its Estimator

In this section, we discuss some properties of discriminability and its estimator. First, the next lemma asserts that the sample discriminability is an unbiased estimator of discriminability.

Lemma 4.4.3 *$\hat{D}$ is an unbiased estimator of D, that is*

$$\mathbb{E}(\hat{D}) = D.$$

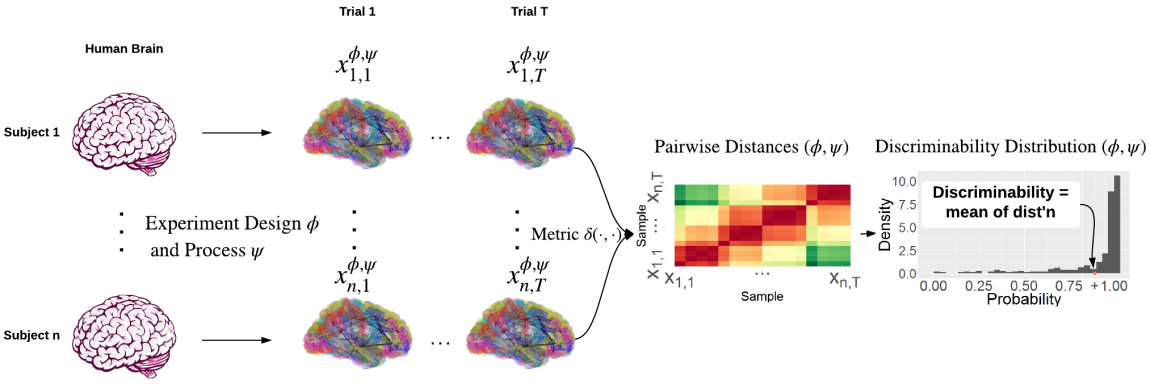

Figure 4.1: Decision Making Through Discriminability Framework. Test-retest data set is collected under experiment design options $\boldsymbol{\phi}$ and processed by pipeline $\boldsymbol{\psi}$. The pairwise distances of all measurements are computed using a metric $\delta(\cdot,\cdot)$. For each pair of measurements of the same subject, we estimate the probability of across subject distances being larger than the within subject distance. Discriminability is the mean of estimated probabilities. Select the option and pipeline with maximum discriminability.

If we keep sampling from new subjects, the sample discriminability will converge to the true discriminability in probability.

Lemma 4.4.4 *As $n \to \infty$, $\hat{D}$ converges to D in probability, that is*

$$\hat{D} \xrightarrow{p} D.$$

To justify our one sample test, we show that under additive noise model discriminability is 0.5 implies independence of $\mathbf{x}$ and $\mathbf{v}$.

Lemma 4.4.5 *Under some regularity conditions, discriminability is* 0.5 *implies measurements are independent of physical property, that is*

$$D = 0.5 \Rightarrow \mathbf{x} \perp \mathbf{v}.$$

4.5 Numerical Results

4.5.1 Simulation: Convergence of Discriminability Estimator

In Lemma 4.4.3 and 4.4.4, we claim discriminability $\hat{D}$ is unbiased and converges to the true population discriminability in probability. We demonstrate these two lemmas through simulation. We consider a simple case that $g_{\boldsymbol{\psi}}$ and $f_{\boldsymbol{\phi}}$ together introduce independent additive Gaussian noise ϵ, that is

$$\mathbf{x}_{i,t} = g_{\boldsymbol{\psi}}\big(f_{\boldsymbol{\phi}}(\mathbf{v}_i)\big) = \mathbf{v}_i + \boldsymbol{\epsilon}_{i,t}. \tag{4.5}$$

$\mathbf{v}_i$ and $\boldsymbol{\epsilon}_{i,t}$ are both independent and identically distributed standard Gaussian random variable that is

$$\mathbf{v}_i \overset{i.i.d.}{\sim} \mathbb{G}(0,1) \text{ , and } \boldsymbol{\epsilon}_{i,t} \overset{i.i.d.}{\sim} \mathbb{G}(0,1).$$

In addition, $\mathbf{v}_i$ and $\boldsymbol{\epsilon}_{i,t}$ are assumed to be independent. For each subject, we sample one true physical property v_i and two noises $\epsilon_{i,t}$ with $t \in \{1,2\}$. Then, two measure-

ments are generated by $x_{i,t} = v_i + \epsilon_{i,t}$. We let the number of subjects n vary from 10 to 200. For each value of n, we repeatedly generate data and compute discriminability 100 times using Euclidean distance. It leaves us 100 estimates of discriminability $\hat{D}$. With this data generation scheme, we can actually compute the population discriminability D through numerical integration, which turns out to be 0.615. Subtracting D from 100 $\hat{D}$s, we can estimate the distribution of estimation error. Figure 4.2 shows the difference between $\hat{D}$ and D. We can see that the mean of difference is centered around 0, and discriminabiity estimates $\hat{D}$ converge to D as the number of subject increases.

4.5.2 Simulation: Test Power of Discriminability

In this section, we investigate the power of one sample and two sample tests for discriminability through simulation. For one sample test for discriminability, we consider the simple additive noise case as in the previous section, that is,

$$\mathbf{x}_{i,t} = \mathbf{v}_i + \boldsymbol{\epsilon}_{i,t}, \mathbf{v}_i \overset{i.i.d.}{\sim} \mathbb{G}(0,1); \boldsymbol{\epsilon}_{i,t} \overset{i.i.d.}{\sim} \mathbb{G}(0,1).$$

Again, we let the number of subjects n to increase from 10 to 200, and for each subject we sample two observations. For each generated data set, we first estimate discriminability, and a p-value is computed based on 100 permutations. The null hypothesis that $D = 0.5$ is rejected when p-value is less than 0.5. Under this data generation scheme, the true discriminability is 0.615; therefore, rejecting the null

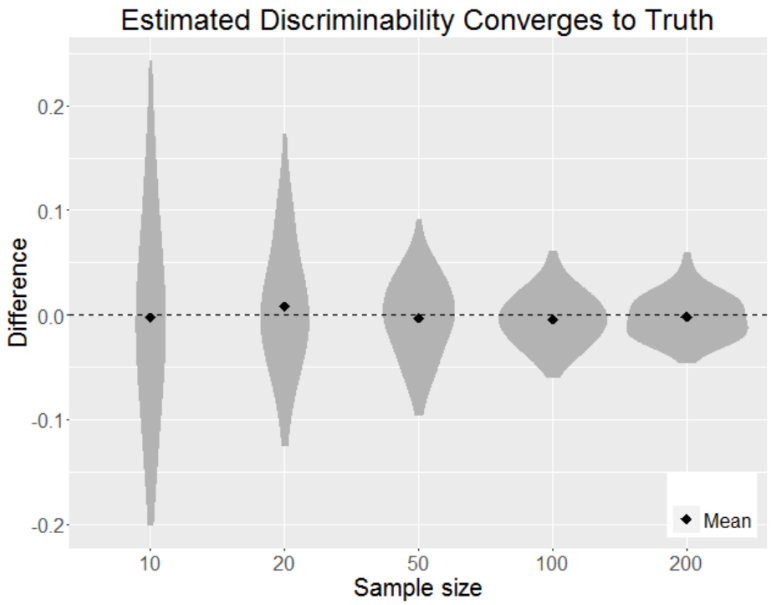

Figure 4.2: Convergence of $\hat{D}$. Distribution of difference between discriminability estimates and truth is shown. The physical property and noise are generated from standard Gaussian distribution as described in the simulation section. The black dots indicate the mean over 100 repeats. As the number of subjects increases, the sample discriminability converges to the true population discriminability.

hypothesis is preferred. For each value of n, we independently generate 100 data sets and perform the one sample test. The fraction of times in which the null hypothesis is rejected with its standard error is shown in Figure 4.3. The power of the test quickly increases as the number of subjects increases, and is close to 1 with more than 50 subjects. For two sample test for discriminability, we generate two sets of measurements $\mathbf{x}_{i,t}^1$ and $\mathbf{x}_{i,t}^2$. The superscript is to denote the pipeline which the measurements come from. The noise is still Gaussian and additive; however, the

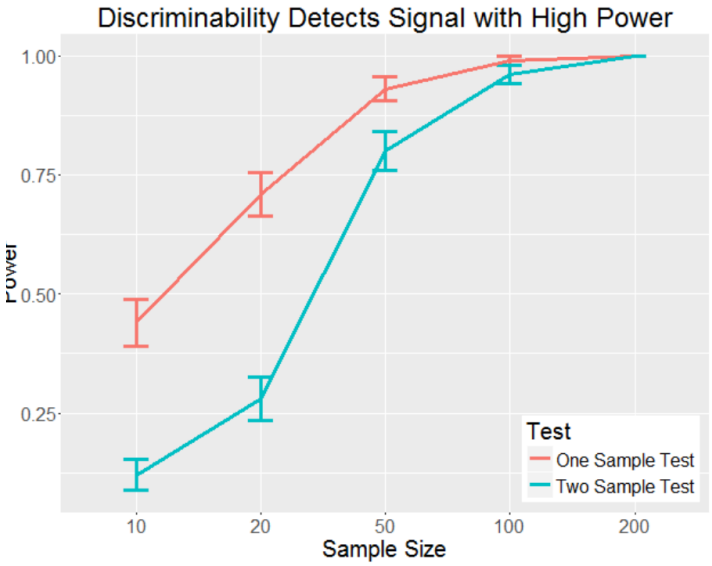

Figure 4.3: Discriminability Test Power. One sample and two sample test power of discriminability with varying sample size is shown. The physical property and additive noise are generated from standard Gaussian distribution as described in the simulation section. At level of 0.05, the power is estimated based on 100 repeats. The power of two tests become close to 1 with more than 100 samples.

pipeline 1 has smaller noise level compared to pipeline 2. Specifically,

$$\mathbf{x}^1_{i,t} = \mathbf{v}_i + \boldsymbol{\epsilon}^1_{i,t}; \mathbf{x}^2_{i,t} = \mathbf{v}_i + \boldsymbol{\epsilon}^2_{i,t}; \mathbf{v}_i \overset{i.i.d.}{\sim} \mathbb{G}(0,1); \boldsymbol{\epsilon}^1_{i,t} \overset{i.i.d.}{\sim} \mathbb{G}(0,0.25); \boldsymbol{\epsilon}^2_{i,t} \overset{i.i.d.}{\sim} \mathbb{G}(0,1).$$

We let the number of subjects n to increase from 10 to 200, and for each subject we sample two observations. We generate measurements for both pipelines, and apply the two sample test procedure as described in Algorithm 6. Under this data generation scheme, the true discriminability of pipeline 1 is larger than that of pipeline 2; therefore, rejecting the null hypothesis is preferred. For each value of n, we in-

dependently generate 100 pairs of data sets and perform the two sample test. The fraction of times in which the null hypothesis is rejected with its standard error is shown in Figure 4.3. The power of the test quickly increases as the number of subjects increases, and is close to 1 with more than 100 subjects.

4.5.3 Simulation: Parameter Selection Through Discriminability

In this simulation, we consider the task of projecting 2-dimensional measurements linearly into 1-dimensional space. Like in the previous experiment, we assume independent additive noise. In addition to $\mathbf{x}_{i,t}$, there is a binary class label $\mathbf{y}_i$ associated with subject i. The true physical property is Gaussian distributed conditioned on $\mathbf{y}_i$,

$$\mathbf{v}_i|\mathbf{y}_i = 1 \overset{i.i.d.}{\sim} \mathbb{G}(\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}) \text{ , and } \mathbf{v}_i|\mathbf{y}_i = 0 \overset{i.i.d.}{\sim} \mathbb{G}(\begin{bmatrix}-1\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}).$$

We consider two cases for the distribution $\boldsymbol{\epsilon}_{i,t}$. The first case is that $\boldsymbol{\epsilon}_{i,t}$ has larger variance in the first coordinate; the other case is that $\boldsymbol{\epsilon}_{i,t}$ has larger variance in the second coordinate, that is

$$\text{Case 1: } \boldsymbol{\epsilon}_{i,t} \sim \mathbb{G}(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}2 & 0\\0 & 1\end{bmatrix})$$

$$\text{Case 2: } \boldsymbol{\epsilon}_{i,t} \sim \mathbb{G}(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 4\end{bmatrix})$$

The noise is assumed to be independent of $\mathbf{v}_i$ and $\mathbf{y}_i$. The Figure 4.4 shows the scatter plot of measurements. Under this generation scheme, the class signal only exists in the first coordinate. Therefore, the optimal linear projection should only keep the first coordinate.

We sample 200 subjects with v_i from each class conditional distribution. Furthermore, 2 measurements are sampled for each subject. We use both discriminability and principal component analysis (PCA) [13] to find the optimal linear projection. After finding the projection, we estimate two class conditional distribution through a kernel density estimator [110]. The results of two cases are provided in two columns of Figure 4.4. In the first case, both methods find the optimal linear projection which separates two classes. However, in the second case only discriminability recovers the optimal projection. PCA finds linear projection with little class signal.

4.5.4 Real Data: Optimal Discriminability Yields Optimal Predictive Accuracy

In this experiment, we are going to investigate the thresholding step in processing resting state functional magnetic resonance imaging (fMRI). In fMRI processing, time series is first extracted for each region of interest (ROI) of brain [111]. Then, a pairwise connectivity matrix is estimated through computing absolute Pearson correlation [112]. To remove noise and obtain a binary graph, the pairwise connectivity

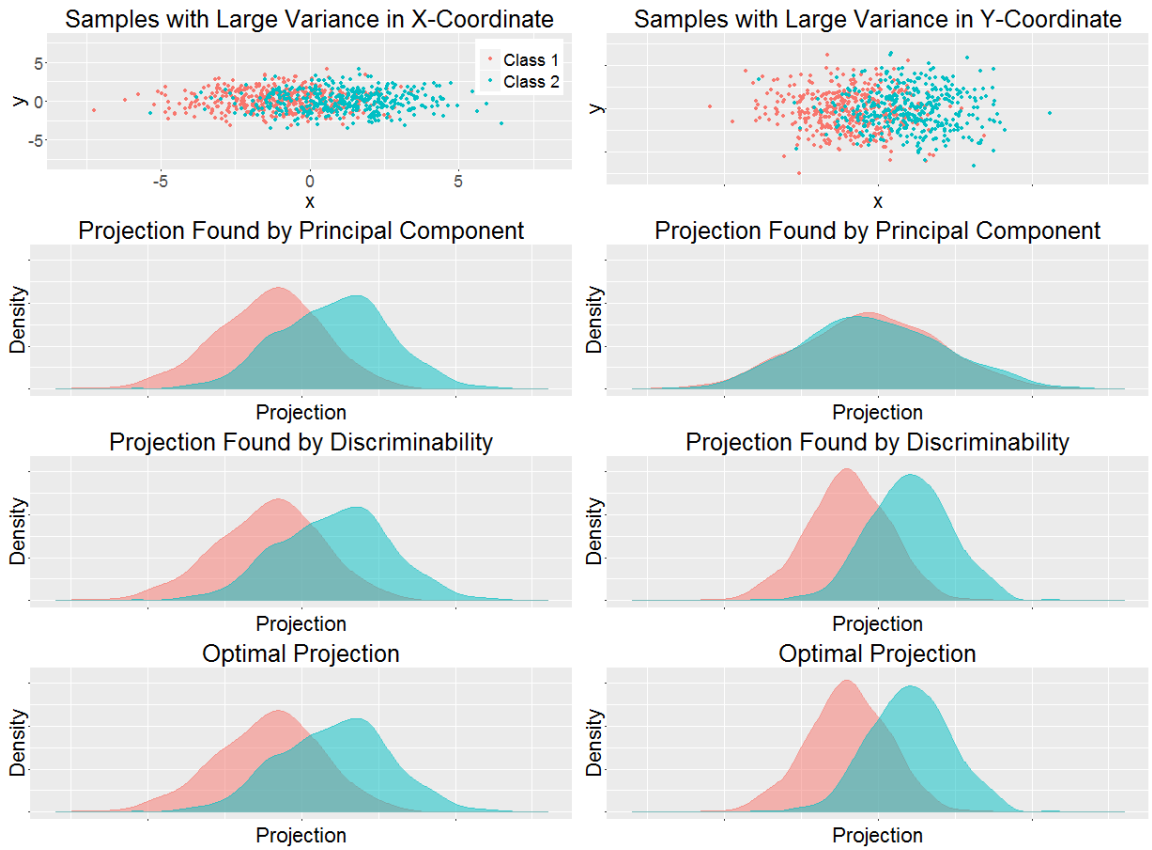

Figure 4.4: Finding the optimal projection. Linear projections are computed using PCA and optimizing discriminability. Physical properties $\mathbf{v}_i$ of 200 subjects are sampled from 2-D two class conditional Gaussian distribution. 2 measurements are sampled for each subject with additive Gaussian noise. Noise could have either large variance in x-coordinate or y-coordinate. The details of generating data can be found in simulations section. The results for two cases are shown in two columns. Maximizing discriminability yields separated samples which have Bayes optimal classification error.

matrix needs to be thresholded by a value which lies in $[0, 1]$ [113, 114]. We would like to find the optimal value for the threshold. In addition to neuroimages, demographic information and five neuro factors [115] are also collected from each subject. We also want to find the threshold which leads to graphs with the best prediction performance.

HCP100 data set is used in this experiment [116]. It contains data from 461 subjects with 4 measurements per subject. We let the threshold vary from 0 to 1. For each value of the threshold, binary graphs is constructed by thresholding correlations. Then, the discriminability is computed with Euclidean distance. In addition, sex, age and the neuro factors are predicted using k-nearest neighbor [117]. For comparison, another reliability statistics, namely image intraclass correlation coefficient (I2C2) is also computed which generalizes intraclass correlation coefficient for high dimensional observations [106]. The discriminability, I2C2, and prediction errors versus the values of threshold are shown in figure 4.5. The threshold which maximizes discriminability is close to the thresholds yielding smallest predicting errors for three covariates.

4.5.5 Real Data: fMRI Processing Pipelines

In this experiment, we are going to investigate the pre-processing options in acquiring resting state fMRI graphs [118]. There have been a lot of steps proposed for pre-processing connectomes in the last decade. Here, we study a subset of them. In particular, we are interested in options include atlas [**?**], anatomical registration [119],

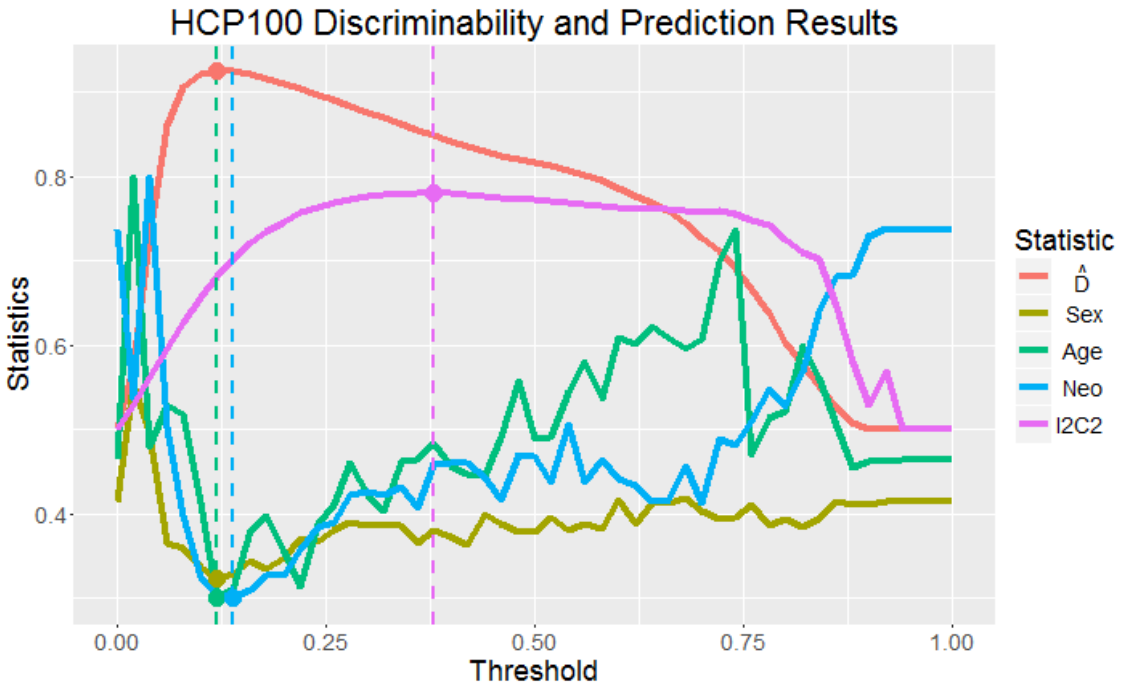

Figure 4.5: Optimizing discriminability yields optimal prediction accuracy for multiple covariates. HCP100 is used to investigate optimal threshold to convert correlation graphs into binary graphs. Curves are scaled to have similar value range. For each statistic, the optimal threshold and value pair is indicated by a circle on the curve. The threshold maximizing discriminability is close to the optimal thresholds for predicting three covariates.

temporal filtering [120], motion correction [121] and nuisance signal regression [122]. We want to find the optimal pre-processing pipeline and the best decision for each option. We are going to index each pipeline by five letters which is explained in Table 4.1. As an example, the best pipeline found is CFXSG which means the data is pre-processed using CC200 atlas, registered with FSL, no frequency filtering, with

Option	Letter
Atlas	C for CC200, H for HOX, A for AAL, D for DES [55, 123]
Anatomical Registration	F for FSL, A for ANTS [124, 125]
Temporal Filtering	F for frequency filtering, X for not [120]
Motion Correction	S for scrubbing, X for not [121]
Nuisance Signal Regression	G for global signal regression , X for not [122]

Table 4.1: fMRI processing options.

scrubbing and with global signal regression. There are 4 possible choices for atlas and 2 possible choices for other options. This leaves us 64 different combinations of options. We select 13 test-retest fMRI data sets with the number of measurements ranging from 50 to 300. The details of these data sets are given in Table 4.2. These data sets are pre-processed by the 64 pipelines through the configurable pipeline for the analysis of connectomes (c-pac) [126]. We also consider an extra rank conversion step which proves to be helpful in boosting discriminability. Rank conversion transforms a weighted undirected graph into a graph with rank weights. Specifically, in the previous experiment all edge weights are absolute correlations which lie in $[0, 1]$. In rank conversion step, for each edge in a graph, its weight w is replaced by the rank of w among all edge weights. If we denote a graph by a node set and an edge weight set pair (V, E) with $E = \{w_{i,j}\}$, rank conversion is a function maps (V, E) to (V, E'), that is

$$(V, E) \to (V, E') \text{ , where } E' = \{\text{rank}(w_{i,j})\}.$$

The rank conversion is designed to improve signal to noise ratio by removing background noise. We carry out this step on the 13 data sets pre-processed by 64 pipelines and compare the difference in discriminability with and without rank conversion. It turns out that the rank conversion does help improving mean discriminability in all pipelines. When global signal regression is not performed, rank conversion significantly boosts discriminability. The Figure 4.6 shows the discriminability of rank fMRI graphs and the discriminability of raw fMRI graphs are provided in Figure 4.7.

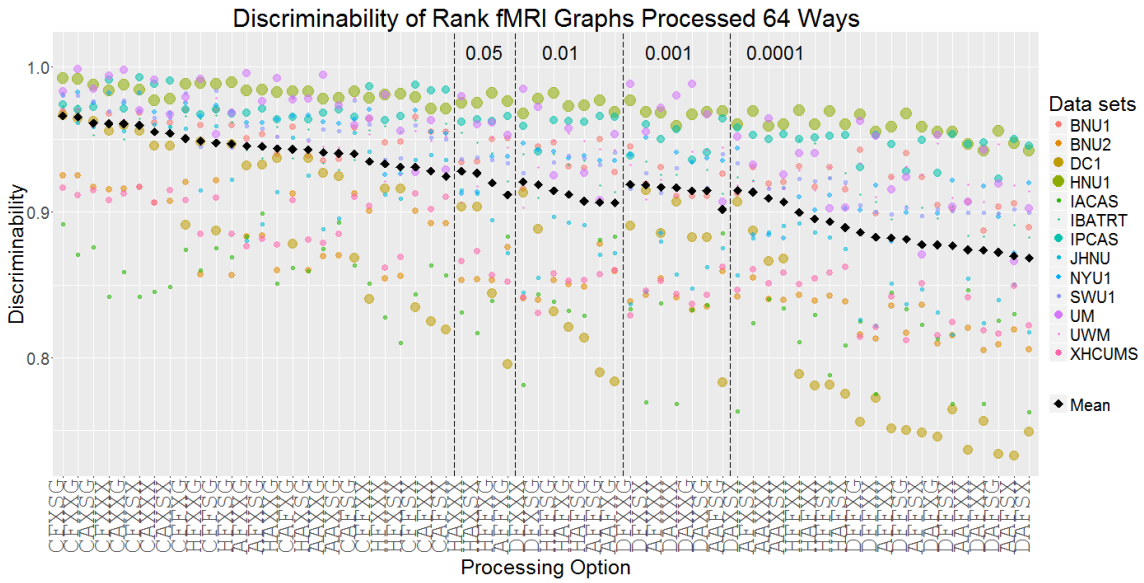

Figure 4.6: Discriminability of rank fmri graphs from 13 data sets processed 64 ways. Discriminability of BNU1, BNU2, DC1, HNU1, IACAS, IBATRT, IPCAS, JHNU, NYU1, SWU1, UM, UWM and XHCUMS pre-processed by 64 pipelines are shown in the plot. Color of each dot indicates data set and size indicates the number of measurements in data set. The black square indicates the weighted mean discriminability across 13 data sets. For each data set, all pipelines are compared to the pipeline CFXSG using two sample test, and a single p-value is calculated by Fisher's method. The pipelines are grouped by p-values. The number at the top indicates the range of the p-values. Within each group, the pipelines are ordered by the mean discriminability. CFXSG pipeline has the best mean discriminability across data sets.

There is notable variation in discriminability. The discriminability of 13 data sets processed by 64 pipelines vary from 0.732 to 0.997. The sample-size weighted mean discriminability of 64 pipelines vary from 0.868 to 0.966. CFXSG turns out to be the

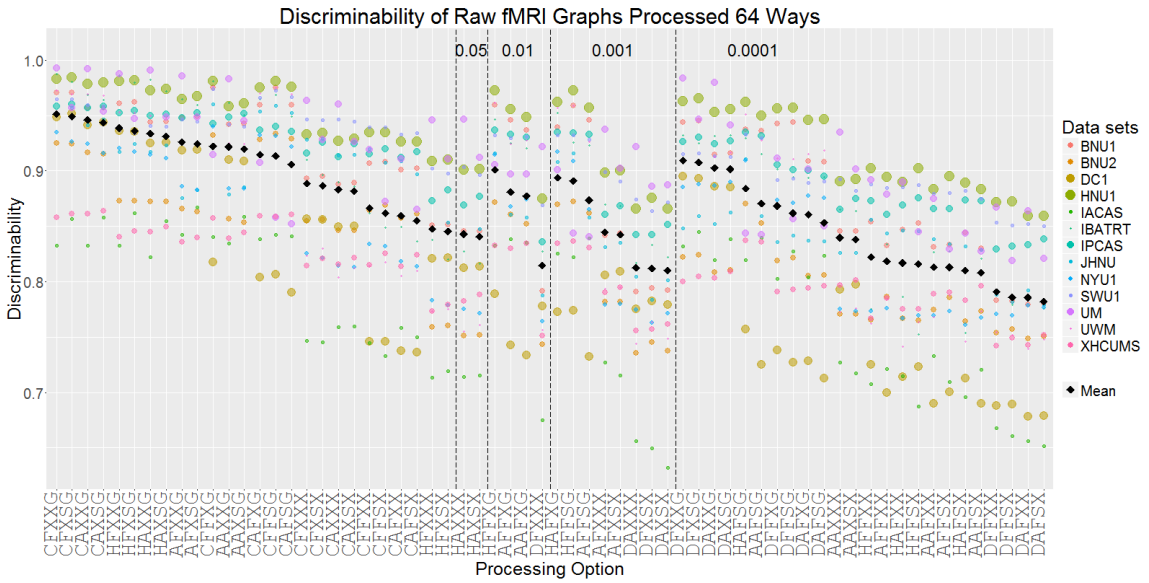

Figure 4.7: Discriminability of raw fmri graphs from 13 data sets processed 64 ways. Discriminability of BNU1, BNU2, DC1, HNU1, IACAS, IBATRT, IPCAS, JHNU, NYU1, SWU1, UM, UWM and XHCUMS pre-processed by 64 pipelines are computed and shown in the figure. Color of each dot indicates data set and size indicates the number of measurements in data set. The black square indicates the weighted mean discriminability across 13 data sets. For each data set, all pipelines are compared to the pipeline CFXSG using two sample test, and a single p-value is calculated by Fisher's method. The pipelines are grouped by p-values. The number at the top indicates the range of the p-values. Within each group, the pipelines are ordered by the mean discriminability. CFXXG pipeline has the best mean discriminability across data sets.

best pipeline with maximum mean discriminability. In Figure 4.6, for each data set, we compare CFXSG to all the other pipelines using the two sample test. We combine

Data set	Scanner	Num. of channel	Structrual Sequence	Functional Sequence	Flip Angle of fMRI	Echo Time (TE in ms)	Repetition Time (TR in ms)	Dimensions (mm x mm x mm)
BNU1	Siemans TrioTim	12 Channel	3D MPRAGE	EPI	90	30	2000	3.1 x 3.1 x 3.5
BNU2 first scan	Siemans TrioTim	12 Channel	3D MPRAGE	EPI	90	30	2000	3.1 x 3.1 x 3.0
BNU2 retest	Siemans TrioTim	12 Channel	3D MPRAGE	EPI	90	30	1500	3.1 x 3.1 x 4
DC1	Philips	32 Channel	3D T1-TFE	EPI	90	35	2500	3 x 3 x 3.5
HNU1	GE Discovery MR750	8 Channel	3D SPGR	EPI	90	30	2000	3.4 x 3.4 x 3.4
JHNU	Siemans TrioTim	8 Channel	3D MPRAGE	EPI	90	30	2000	3.75 x 3.75 x 4
IACAS	GE Sigma HDx	8 Channel	3D BRAVO	EPI	90	30	2000	3.4 x 3.4 x 4
IBATRT	Siemans TrioTim	12 Channel	3D MPRAGE	EPI	90	30	1750	3.4 x 3.4 x 3.6
NYU1	Siemans Allegro				90	15	2000	3 x 3 x 4
SWU1								
UM								
UWM								
IPCAS								
XHCUMS	Siemans Allegro				90	15	2000	3 x 3 x 3

Table 4.2: fMRI data sets with scanning parameters

the p-values by Fisher's method [127], then we group pipelines by the magnitude of their p-values and order them by mean discriminability. Furthermore, we carried out

a multi-factor analysis of variance test to study each option [128]. Specifically, we fix decisions for all options except one, and investigates whether there is significant difference in discriminability. It turns out that FSL, no frequency filtering, no scrubbing, global signal regression and rank conversion is better than their alternatives in terms of mean discriminability. However, fsl and no scrubbing is not statistical significantly better at level 0.05. No frequency filtering, global signal regression and rank conversion is better than their alternatives at level 0.001. Figure 4.8 shows the distribution of paired difference in discriminability.

4.5.6 Real Data: DTI Experiment Design and Processing

In this experiment, we consider the experiment design of collecting DTI data. In particular, we are interested the effect of b-value and number of directions on discriminability [129]. We pick four data sets with different b-value and number of directions and compute discriminability. The result is show in the right panel of Figure 4.9. We can see they have comparable discriminability. Given four data sets, we cannot conclude the optimal value for the parameters. It would be ideal if we could carry out a more controlled study with more data.

We also consider the processing of diffusion tensor imaging (DTI) [129]. In particular, we are interested in finding the optimal number of ROI, and the optimal

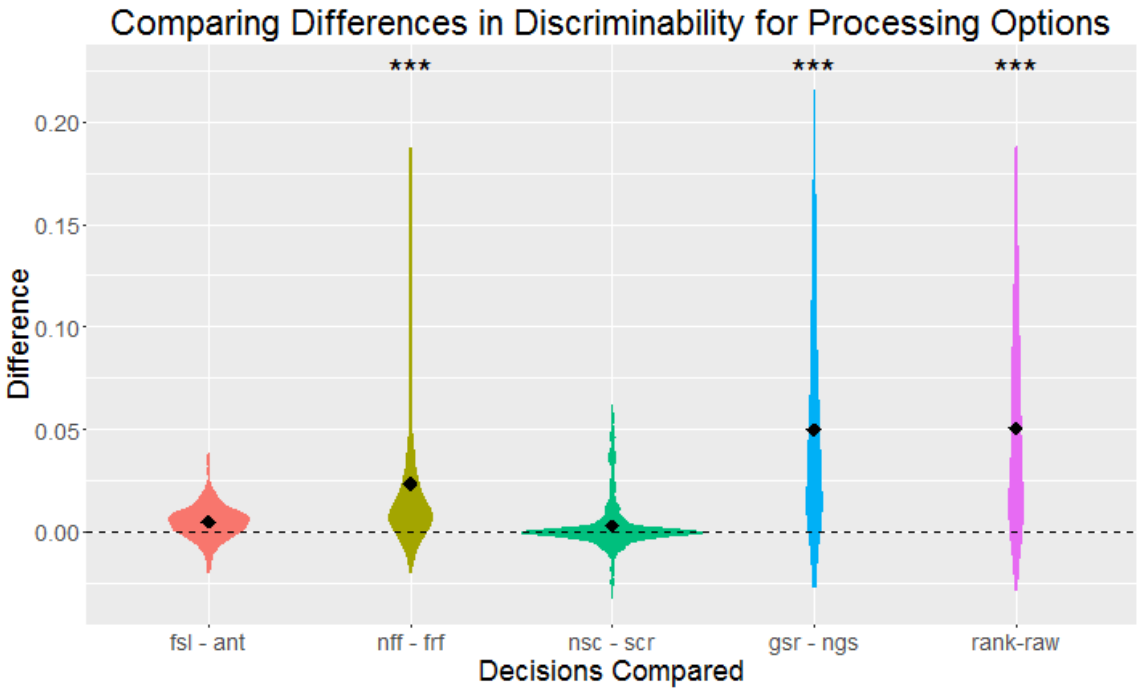

Figure 4.8: Paired difference in discriminability of pre-processing options. Difference in discriminability for each option is compared by fixing the other options and data set. The symbols at top indicates the significance. No frequency filtering, global signal regression and rank conversion are statistical significantly better than their alternatives at level 0.001. Fsl and no scrubbing are not significantly better.

approach to process edge weights. SWU4 data set is used in this experiment. We process four DTI data sets using 15 atlases with the number of ROI ranging from 48 to 1875 [130]. For edge weights, we consider three options. First, raw edge weights are used which are fiber counts. Furthermore, we consider two alternatives: log weights and rank weights as discussed in the previous experiment. Top left panel of figure 4.9 shows the results. We see discriminability is basically stable across different atlases

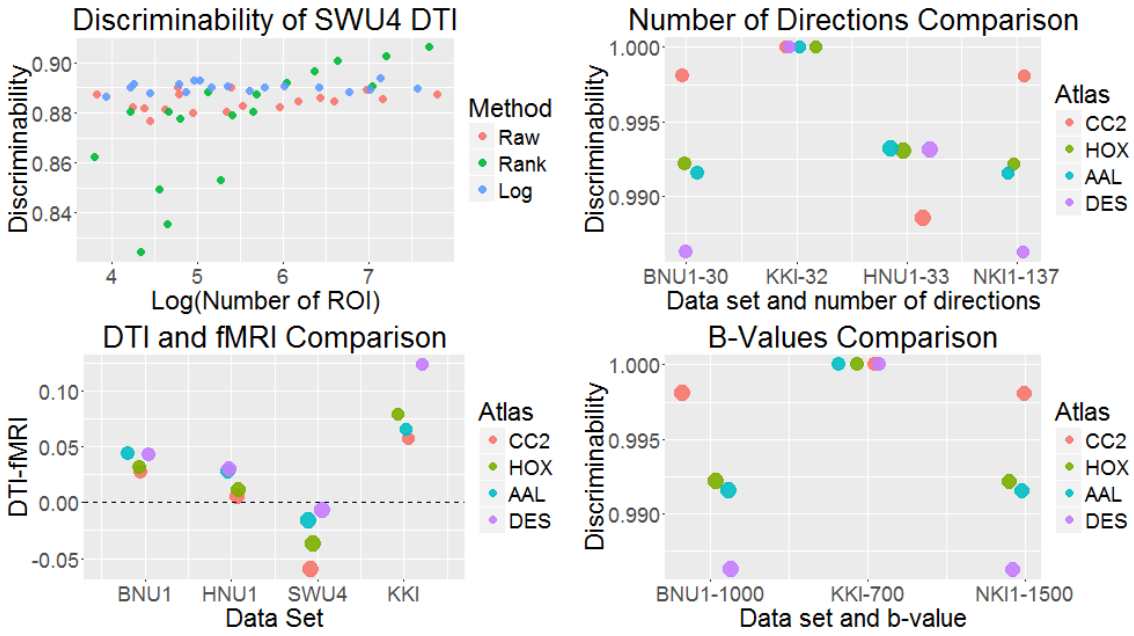

Figure 4.9: Discriminability of DTI data sets. The top left plot shows the discriminability of SWU4 registered with 15 atlases with ROI varying from 48 to 1875. Raw, rank and log edges weights are considered. Discriminability of DTI and fMRI graphs are compared for BNU1, HNU1, SWU4 and KKI data set. The results are shown in the bottom left panel. DTI data sets tend to be more discriminable than fMRI data sets. The plots in the right column show the discriminability of different data sets with different b-values and number of directions.

when raw and log edge weights are used. When using the rank weights, discriminability is low when the number of ROI is small and high when ROI is large. For three out of four data sets, the discriminability is very close to 1. As a consequence, we cannot find any statistical relationship between the number of ROI and discriminability.

Furthermore, we want to compare discriminanility of fMRI and DTI data sets.

Four data sets with both fMRI and DTI images are selected for the comparison. In processing fMRI data sets, the most dicriminable pipeline (*FXXG) and raw edge weights are used. In processing DTI data sets, the raw edge weights are also used. The detailed DTI processing configurations and parameters are provided in the appendix. The result is shown in the bottom left panel of Figure 4.9. Our conclusion is that DTI data sets have at least comparable discriminability as fMRI data sets. Actually, DTI measurements are better than fMRI measurements in three out of four data sets.

4.6 Discussion

We propose a non-parametric statistics of discriminability which is define to be the probability that within subject distance is smaller than across subject distance. We prove discriminability bounds Bayes prediction error. An estimator is designed to estimate the discriminability based on test-retest data set. We show the estimator is unbiased and converges to the discriminability asymptotically. Furthermore, we developed one sample and two sample tests for discriminability, which can be used to detect subject signal in data set and compare discriminability of two processing pipelines. We apply the discriminability framework under various setups in neuroimaging processing. We find the best processing pipeline for fMRI pre-processing and look into options in DTI processing. Furthermore, fMRI and DTI are shown to have comparable discriminability.

From the theoretical point of view, most of our theories require the noise to be additive and independent of subjects. The effects of subject specific noise on discriminability are left uninvestigated. As for applications, more experiments should be carried out to analyze processing options. In particular, we could investigate processing of DTI more thoroughly given more data sets. Also, the effect of the number of ROI on discriminability is still not determined. Second, metrics other than Euclidean distance could be studied. Third, a testing procedure could be developed for comparing discriminability of multiple data sets.

4.7 Proofs

Proof of Theorem 4.4.1 Consider the additive noise setting, that is $\mathbf{x}_{i,t} = \mathbf{v}_i + \boldsymbol{\epsilon}_{i,t}$,

$$
\begin{aligned}
& \mathbb{P}(\delta_{i,t,t'} < \delta_{i,i',t,t''}) \\
= \; & \mathbb{P}(\|\mathbf{x}_{i,t} - \mathbf{x}_{i,t'}\| < \|\mathbf{x}_{i,t} - \mathbf{x}_{i',t''}\|) \\
= \; & \mathbb{P}(\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| < \|\mathbf{v}_i + \boldsymbol{\epsilon}_{i,t} - \mathbf{v}_{i'} - \boldsymbol{\epsilon}_{i',t''}\|) \\
\leq \; & \mathbb{P}(\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| < \|\mathbf{v}_i - \mathbf{v}_{i'}\| + \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\|) \\
= \; & \mathbb{P}(\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\| < \|\mathbf{v}_i - \mathbf{v}_{i'}\|) \\
= \; & \frac{1}{2}\mathbb{P}(\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\| < \|\mathbf{v}_i - \mathbf{v}_{i'}\| | \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\| < 0) + \\
& \frac{1}{2}\mathbb{P}(\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\| < \|\mathbf{v}_i - \mathbf{v}_{i'}\| | \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\| > 0) \\
= \; & \frac{1}{2} + \frac{1}{2}\mathbb{P}(\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\| < \|\mathbf{v}_i - \mathbf{v}_{i'}\| | \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\| > 0) \\
= \; & \frac{1}{2} + \frac{1}{2}\mathbb{P}(\big|\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\|\big| < \|\mathbf{v}_i - \mathbf{v}_{i'}\|) \\
= \; & 1 - \frac{1}{2}\mathbb{P}(\big|\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\|\big| > \|\mathbf{v}_i - \mathbf{v}_{i'}\|).
\end{aligned}
$$

To bound the probability above, we bound the $\|\mathbf{v}_i - \mathbf{v}_{i'}\|$ and $\big|\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\|\big|$ separately. We start with the first term

$$
\begin{aligned}
& \mathbb{E}(\|\mathbf{v}_i - \mathbf{v}_{i'}\|^2) \\
= \; & \mathbb{E}(\mathbf{v}_i^T\mathbf{v}_i + \mathbf{v}_{i'}^T\mathbf{v}_{i'} - 2\mathbf{v}_i^T\mathbf{v}_{i'}) \\
= \; & 2\sigma_2^2.
\end{aligned}
$$

Here, σ_2^2 is the trace of covariance matrix of $\mathbf{v}_i$. We can apply Markov's Inequality

$$\mathbb{P}(\|\mathbf{v}_i - \mathbf{v}_{i'}\| < t) \geq 1 - \frac{2\sigma_2^2}{t^2}.$$

Let σ_1^2 denote the trace of covariance matrix of $\epsilon_{i,t}$, and let a and b be two constants satisfy

$$\mathbb{E}(\big|\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\|\big|^2) \geq a^2\sigma_1^2,$$
$$\frac{\mathbb{E}^2(\big|\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\|\big|^2)}{\mathbb{E}(\big|\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\|\big|)^4} \geq b.$$

Then, we can apply Paley-Zygmund Inequality [131],

$$\mathbb{P}(\big|\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\|\big|^2 > t^2) \geq b(1 - \frac{t^2}{a^2\sigma_1^2})^2.$$

Understand the fact that $\mathbf{v}$s and $\boldsymbol{\epsilon}$s are independent, we can combine the two inequalities and get a bound on $\mathbb{P}(\delta_{i,t,t'} < \delta_{i,i',t,t''})$

$$\begin{aligned}
& \mathbb{P}(\delta_{i,t,t'} < \delta_{i,i',t,t''}) \\
= \ & \mathbb{P}(\|\mathbf{x}_{i,t} - \mathbf{x}_{i,t'}\| < \|\mathbf{x}_{i,t} - \mathbf{x}_{i',t''}\|) \\
\leq \ & 1 - \frac{1}{2}\mathbb{P}(\big|\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\|\big| > \|\mathbf{v}_i - \mathbf{v}_{i'}\|) \\
\leq \ & 1 - \frac{1}{2}\mathbb{P}(\big|\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| - \|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\|\big|^2 > t^2)P(\|\mathbf{v}_i - \mathbf{v}_{i'}\|^2 < t^2) \\
\leq \ & 1 - \frac{1}{2}b(1 - \frac{t^2}{a^2\sigma_1^2})^2(1 - \frac{2\sigma_2^2}{t^2}).
\end{aligned}$$

Assume $a^2\sigma_1^2 \geq 2\sigma_2^2$ and set $t^2 = \sqrt{2}a\sigma_1\sigma_2$,

$$\mathbb{P}(\|\mathbf{x}_{i,t} - \mathbf{x}_{i,t'}\| < \|\mathbf{x}_{i,t} - \mathbf{x}_{i',t''}\|) \leq 1 - \frac{1}{2}b(1 - \frac{\sqrt{2}\sigma_2}{a\sigma_1})^3.$$

By definition, $D = \mathbb{P}(\|\mathbf{x}_{i,t} - \mathbf{x}_{i,t'}\| < \|\mathbf{x}_{i,t} - \mathbf{x}_{i',t''}\|)$, we can have a bound on $\frac{\sigma_2}{\sigma_1}$,

$$\frac{\sigma_2}{\sigma_1} \geq \frac{a}{\sqrt{2}}(1 - (\frac{2-2D}{b})^{1/3}). \tag{4.6}$$

To obtain a bound on Bayes error, we apply Devijver and Kittler's result [132], which is

$$L(g^*) \leq \frac{2\pi_0\pi_1}{1 + \pi_0\pi_1\Delta\mu^T\Sigma^{-1}\Delta\mu}.$$

Here, π_0 and π_1 are prior probabilities for two classes. $\Delta\mu$ is the difference between means of two classes. Since $\boldsymbol{\epsilon}$ is assumed to be independent of $\mathbf{x}$ and $\mathbf{y}$,

$$\Delta\mu = \mathbb{E}(\mathbf{x}|\mathbf{y}=0) - \mathbb{E}(\mathbf{x}|\mathbf{y}=1) = \mathbb{E}(\mathbf{v}|\mathbf{y}=0) - \mathbb{E}(\mathbf{v}|\mathbf{y}=1).$$

Σ is the weighted covariance matrix of $\mathbf{x}$,

$$\begin{aligned}\Sigma &= \pi_0\mathrm{Var}(\mathbf{x}|\mathbf{y}=0) + \pi_1\mathrm{Var}(\mathbf{x}|\mathbf{y}=1) \\ &= \pi_0\mathrm{Var}(\mathbf{v}|\mathbf{y}=0) + \pi_1\mathrm{Var}(\mathbf{v}|\mathbf{y}=1) + \mathrm{Var}(\boldsymbol{\epsilon}).\end{aligned}$$

If we further assume $\mathrm{Var}(\boldsymbol{\epsilon}) = \lambda\Sigma'$ where the trace of Σ is 1, then equation 6 implies $\lambda \leq \lambda_*$, where

$$\lambda_* = \frac{\sqrt{2}\sigma_2}{a(1 - (\frac{2-2D}{b})^{1/3})}.$$

Hence, $\Sigma \leq \Sigma_*$ where

$$\Sigma_* = \pi_0\mathrm{Var}(\mathbf{v}|\mathbf{y}=0) + \pi_1\mathrm{Var}(\mathbf{v}|\mathbf{y}=1) + \lambda^*\Sigma'.$$

Therefore, $\Sigma^{-1} \geq \Sigma_*^{-1}$, and we have

$$\begin{aligned} L(g^*) &\leq \frac{2\pi_0\pi_1}{1+\pi_0\pi_1\Delta\mu^T\Sigma^{-1}\Delta\mu} \\ &\leq \frac{2\pi_0\pi_1}{1+\pi_0\pi_1\Delta\mu^T\Sigma_*^{-1}\Delta\mu}. \end{aligned}$$

Proof of Lemma 4.4.3 By definition of $\hat{D}$,

$$\hat{D} = \frac{\sum_{i=1}^{n}\sum_{t=1}^{s}\sum_{t'\neq t}^{s}\hat{D}_{i,t,t'}}{ns(s-1)}.$$

Notice that the expectation of $\hat{D}_{i,t,t'}$ is actually D,

$$\begin{aligned} \mathbb{E}(\hat{D}_{i,t,t'}) &= \frac{\sum_{i'\neq i}^{n}\sum_{t''=1}^{s}\mathbb{E}(\mathbb{I}\{\delta_{i,t,t'} < \delta_{i,i',t,t''}\})}{(n-1)s} \\ &= \frac{\sum_{i'\neq i}^{n}\sum_{t''=1}^{s}\mathbb{P}[\delta_{i,t,t'} < \delta_{i,i',t,t''}]}{(n-1)s} \\ &= \frac{\sum_{i'\neq i}^{n}\sum_{t''=1}^{s}D}{(n-1)s} \\ &= D. \end{aligned}$$

Therefore, we have

$$\begin{aligned} \mathbb{E}(\hat{D}) &= \frac{\sum_{i=1}^{n}\sum_{t=1}^{s}\sum_{t'\neq t}^{s}\mathbb{E}(\hat{D}_{i,t,t'})}{ns(s-1)} \\ &= \frac{\sum_{i=1}^{n}\sum_{t=1}^{s}\sum_{t'\neq t}^{s}D}{ns(s-1)} \\ &= D. \end{aligned}$$

This concludes that $\hat{D}$ is an unbiased estimator of discriminability D.

Proof of Lemma 4.4.4 By definition of $\hat{D}$,

$$\begin{aligned}\hat{D} &= \frac{\sum_{i=1}^{n}\sum_{t=1}^{s}\sum_{t'\neq t}^{s}\hat{D}_{i,t,t'}}{ns(s-1)} \\ &= \frac{\sum_{i=1}^{n}\sum_{t=1}^{s}\sum_{t'\neq t}^{s}\sum_{i'\neq i}^{n}\sum_{t''=1}^{s}\mathbb{I}\{\delta_{i,t,t'} < \delta_{i,i',t,t''}\}}{ns(s-1)(n-1)s} \\ &= \frac{\sum_{i,i',t,t',t''}\mathbb{I}\{\delta_{i,t,t'} < \delta_{i,i',t,t''}\}}{ns(s-1)(n-1)s}.\end{aligned}$$

In the last sum above, we should keep in mind that $i \neq i'$ and $t \neq t'$. We show in the previous lemma that $\mathbb{E}(\hat{D}) = D$. To demonstrate that $\hat{D}$ converges to D in probability, it is suffice to show that $\mathrm{Var}(\hat{D}) \to 0$. Since then, by Chebyshev's inequality,

$$\mathbb{P}[|\hat{D} - D| \geq \epsilon] \leq \frac{\mathrm{Var}(\hat{D})}{\epsilon^2} \to 0.$$

If we expand the variance of R,

$$\mathrm{Var}(\hat{D}) = \frac{\sum_{i,i',t,t',t''}\sum_{j,j',r,r',r''}\mathrm{Cov}(\mathbb{I}\{\delta_{i,t,t'} < \delta_{i,i',t,t''}\}, \mathbb{I}\{\delta_{j,r,r'} < \delta_{j,j',r,r',r''}\})}{(ns(s-1)(n-1)s)^2}.$$

There are $(ns(s-1)(n-1)s)^2$ covariance terms in the sum of nominator; however, most of them are actually 0. $\mathbb{I}\{\delta_{i,t,t'} < \delta_{i,i',t,t''}\}$ is a function of $\mathbf{x}_{i,t}$, $\mathbf{x}_{i,t'}$ and $\mathbf{x}_{i',t''}$; therefore, is independent of any observations of subjects other than i and i'. This implies $\mathbb{I}\{\delta_{i,t,t'} < \delta_{i,i',t,t''}\}$ is independent of $\mathbb{I}\{\delta_{j,r,r'} < \delta_{j,j',r,r''}\}$ as long as $\{i, i'\} \cap \{j, j'\} = \emptyset$. As a consqeunce, there are $(4n-6)(s(s-1)s) = ns(s-1)(n-1)s-(n-2)s(s-1)(n-3)s$ combinations of j, j', r, r', r'' such that covariance between $\mathbb{I}\{\delta_{i,t,t'} < \delta_{i,i',t,t''}\}$ and $\mathbb{I}\{\delta_{j,r,r'} < \delta_{j,j',r,r',r''}\}$ maybe non-zero. Furthermore, the covariance must be less $\frac{1}{4}$ due

to the fact that they are indicator random variables. Therefore, we have

$$\begin{aligned}
\mathrm{Var}(\hat{D}) &= \frac{\sum_{i,i',t,t',t''} \sum_{j,j',r,r',r''} \mathrm{Cov}(\mathbb{I}\{\delta_{i,t,t'} < \delta_{i,i',t,t''}\}, \mathbb{I}\{\delta_{j,r,r'} < \delta_{j,j',r,r',r''}\})}{(ns(s-1)(n-1)s)^2} \\
&\leq \frac{\sum_{i,i',t,t',t''} (4n-6)(s(s-1)s)}{4(ns(s-1)(n-1)s)^2} \\
&= \frac{(4n-6)(s(s-1)s)}{4ns(s-1)(n-1)s} \\
&= \frac{4n-6}{4n(n-1)} \\
&< \frac{1}{n} \\
&\to 0 \text{ , as } n \to \infty.
\end{aligned}$$

As discussed before, this concludes that $\hat{D}$ converges to D in probability.

Proof of Lemma 4.4.5 Consider the additive noise setting, that is $\mathbf{x}_{i,t} = \lambda \mathbf{v}_i + \boldsymbol{\epsilon}_{i,t}$. We further assume $\mathbf{v}_i$ and $\boldsymbol{\epsilon}_{i,t}$ have continuous distributions, and $\mathbf{v}_i$ has spherical distribution. We will show that $D = 0.5$ implies $\lambda = 0$, hence $\mathbf{x}_{i,t} = \boldsymbol{\epsilon}_{i,t}$. This implies $\mathbf{x}_{i,t}$ is independent of physical property $\mathbf{v}_i$ and hence, any phenotype $\mathbf{y}_i$. First, we rewrite the definition of discriminability.

$$\begin{aligned}
& \mathbb{P}(\delta_{i,t,t'} < \delta_{i,i',t,t''}) \\
= \; & \mathbb{P}(\|\mathbf{x}_{i,t} - \mathbf{x}_{i,t'}\| < \|\mathbf{x}_{i,t} - \mathbf{x}_{i',t''}\|) \\
= \; & \mathbb{P}(\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| < \|\lambda \mathbf{v}_i + \boldsymbol{\epsilon}_{i,t} - \lambda \mathbf{v}_{i'} - \boldsymbol{\epsilon}_{i',t''}\|) \\
= \; & \mathbb{E}(\mathbb{P}(\|\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}\| < \|\lambda \mathbf{v}_i - \lambda \mathbf{v}_{i'} + \boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}\|) | \|\lambda \mathbf{v}_i - \lambda \mathbf{v}_{i'}\| = v).
\end{aligned}$$

Let $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{V}$ denote the $\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i,t'}$, $\boldsymbol{\epsilon}_{i,t} - \boldsymbol{\epsilon}_{i',t''}$ and $\lambda \mathbf{v}_i - \lambda \mathbf{v}_{i'}$ respectively. Due

to the assumption that $\mathbf{v}_i$ has spherical distribution,

$$\mathbb{P}(\|\mathbf{A}_1\| < \|\mathbf{A}_2 + \mathbf{V}\| | \|\mathbf{V}\| = v) = \int_{\mathbb{S}^{d-1}} \mathbb{P}(\|\mathbf{A}_1\| < \|\mathbf{A}_2 + t\|) dS / Area(\mathbb{S}^{d-1}),$$

where $\mathbb{S}^{d-1}$ is the ball in $\mathbb{R}^d$ with radius v. We are going to show the expression above is greater than 0.5 as long as $v > 0$. Therefore, $D = 0.5$ implies $\lambda \mathbf{v}_i - \lambda \mathbf{v}_{i'} = 0$. Since $\mathbf{v}_i$ is not constant, we have $\lambda = 0$. Due to symmetry in $\mathbf{A}_1$ and $\mathbf{A}_2$, we have

$$\begin{aligned}
& 2 \int_{\mathbb{S}^{d-1}} \mathbb{P}(\|\mathbf{A}_1\| < \|\mathbf{A}_2 + t\|) dS \\
= & \int_{\mathbb{S}^{d-1}} \mathbb{P}(\|\mathbf{A}_1\| < \|\mathbf{A}_2 + t\|) dS + \int_{\mathbb{S}^{d-1}} \mathbb{P}(\|\mathbf{A}_2\| < \|\mathbf{A}_1 + t\|) dS \\
= & \int_{\mathbb{S}^{d-1}} \mathbb{P}(\|\mathbf{A}_1\| < \|\mathbf{A}_2 + t\|) + \mathbb{P}(\|\mathbf{A}_2\| < \|\mathbf{A}_1 + t\|) dS \\
= & \int_{\mathbb{S}^{d-1}} \int \mathbb{I}(\|a_1\| < \|a_2 + t\|) + \mathbb{I}(\|a_2\| < \|a_1 + t\|) d\mathbb{P}(a_1, a_2) dS \\
= & \iint_{\mathbb{S}^{d-1}} \mathbb{I}(\|a_1\| < \|a_2 + t\|) + \mathbb{I}(\|a_2\| < \|a_1 + t\|) dS d\mathbb{P}(a_1, a_2).
\end{aligned}$$

Let us consider the inner integral $\int_{\mathbb{S}^{d-1}} \mathbb{I}(\|a_1\| < \|a_2 + t\|) + \mathbb{I}(\|a_2\| < \|a_1 + t\|) dS$ and denote its value by V. Next, we show V is greater than or equal to $Area(\mathbb{S}^{d-1})$ for any a_1 and a_2. First, let us consider the case that t lies on the circle which is contained in the plane spanned by a_1 and a_2, there are three cases.

1. If $\|t\| \leq |\|a_1\| - \|a_2\||$, then one of the two indicators holds for all t; hence, $V = Area(\mathbb{S}^1)$.

2. If $|\|a_1\| - \|a_2\|| < \|t\| \leq |\|a_1\| + \|a_2\||$, then due to symmetry $V = Area(\mathbb{S}^1)$.

3. If $\|t\| > |\|a_1\| + \|a_2\||$, then both of the two indicators holds for all t; hence, $V = 2Area(\mathbb{S}^1)$.

If t does not lie in the plane spanned by a_1 and a_2, we can always project t on to the plane first. This discussion shows that

$$\mathbb{I}(\|a_1\| < \|a_2 + t\|) + \mathbb{I}(\|a_2\| < \|a_1 + t\|) \geq 1,$$

for any a_1, a_2 and t. Therefore, this implies V always greater than or equal to $Area(\mathbb{S}^{d-1})$. Since $\mathbf{A}_1$ and $\mathbf{A}_2$ have positive mass at any open ball centered at origin, case (3) must happen with positive probability. As a consequence,

$$\iint_{\mathbb{S}^{d-1}} \mathbb{I}(\|a_1\| < \|a_2 + t\|) + \mathbb{I}(\|a_2\| < \|a_1 + t\|)dSd\mathbb{P}(a_1, a_2) > Area(\mathbb{S}^{d-1})$$

This shows $\mathbb{P}(\|\mathbf{A}_1\| < \|\mathbf{A}_2 + \mathbf{V}\| | \|\mathbf{V}\| = v) > 0.5$ as long as $v \neq 0$. Therefore, $\mathbb{P}(\|\mathbf{A}_1\| < \|\mathbf{A}_2 + \mathbf{V}\|) = 0.5$ implies $\mathbf{V} = 0$. As discussed above, this shows $\mathbf{x}_{i,t}$ is independent of $\mathbf{v}_i$.

Bibliography

[1] E. T. Bullmore and D. S. Bassett, "Brain graphs: graphical models of the human brain connectome," *Annual review of clinical psychology*, vol. 7, pp. 113–140, 2011.

[2] P. Emsley and K. Cowtan, "Coot: model-building tools for molecular graphics," *Acta Crystallographica Section D: Biological Crystallography*, vol. 60, no. 12, pp. 2126–2132, 2004.

[3] E. Otte and R. Rousseau, "Social network analysis: a powerful strategy, also for the information sciences," *Journal of information Science*, vol. 28, no. 6, pp. 441–453, 2002.

[4] D. L. Sussman, M. Tang, D. E. Fishkind, and C. E. Priebe, "A consistent adjacency spectral embedding for stochastic blockmodel graphs," *Journal of the American Statistical Association*, vol. 107, no. 499, pp. 1119–1128, 2012.

[5] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction

and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[6] S. Wang, J. T. Vogelstein, and C. E. Priebe, "Joint embedding of graphs," *arXiv preprint arXiv:1703.03862*, 2017.

[7] J. Fan and J. Lv, "Sure independence screening for ultrahigh dimensional feature space," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 5, pp. 849–911, 2008.

[8] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity," 2011.

[9] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[10] D. P. Ballou and H. L. Pazer, "Modeling data and process quality in multi-input, multi-output information systems," *Management science*, vol. 31, no. 2, pp. 150–162, 1985.

[11] A. M. Dale, "Optimal experimental design for event-related fmri," *Human brain mapping*, vol. 8, no. 2-3, pp. 109–114, 1999.

BIBLIOGRAPHY

[12] J. R. Banga and E. Balsa-Canto, "Parameter estimation and optimal experimental design," *Essays in biochemistry*, vol. 45, pp. 195–210, 2008.

[13] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[14] G. Li, M. Semerci, B. Yener, and M. J. Zaki, "Graph classification via topological and label attributes," in *Proceedings of the 9th international workshop on mining and learning with graphs (MLG), San Diego, USA*, vol. 2, 2011.

[15] Y. Park, C. E. Priebe, and A. Youssef, "Anomaly detection in time series of graphs using fusion of graph invariants," *IEEE journal of selected topics in signal processing*, vol. 7, no. 1, pp. 67–75, 2013.

[16] C. Jiang, F. Coenen, and M. Zito, "A survey of frequent subgraph mining algorithms," *The Knowledge Engineering Review*, vol. 28, no. 01, pp. 75–105, 2013.

[17] J. Huan, W. Wang, and J. Prins, "Efficient mining of frequent subgraphs in the presence of isomorphism," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, 2003, pp. 549–552.

[18] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 1151–1157.

BIBLIOGRAPHY

[19] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social networks*, vol. 5, no. 2, pp. 109–137, 1983.

[20] B. Karrer and M. E. Newman, "Stochastic blockmodels and community structure in networks," *Physical Review E*, vol. 83, no. 1, p. 016107, 2011.

[21] V. Lyzinski, M. Tang, A. Athreya, Y. Park, and C. E. Priebe, "Community detection and classification in hierarchical stochastic blockmodels," *arXiv preprint arXiv:1503.02115*, 2015.

[22] P. ERDdS and A. R&WI, "On random graphs i," *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.

[23] S. J. Young and E. R. Scheinerman, "Random dot product graph models for social networks," in *Algorithms and models for the web-graph*. Springer, 2007, pp. 138–149.

[24] P. D. Hoff, A. E. Raftery, and M. S. Handcock, "Latent space approaches to social network analysis," *Journal of the american Statistical association*, vol. 97, no. 460, pp. 1090–1098, 2002.

[25] M. Tang, D. L. Sussman, C. E. Priebe *et al.*, "Universally consistent vertex classification for latent positions graphs," *The Annals of Statistics*, vol. 41, no. 3, pp. 1406–1430, 2013.

[26] A. Athreya, C. Priebe, M. Tang, V. Lyzinski, D. Marchette, and D. Sussman, "A

limit theorem for scaled eigenvectors of random dot product graphs," *Sankhya A*, pp. 1–18, 2013.

[27] D. Bini, M. Capovani, F. Romani, and G. Lotti, "O (n2. 7799) complexity for n× n approximate matrix multiplication," *Information processing letters*, vol. 8, no. 5, pp. 234–235, 1979.

[28] V. De Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1084–1127, 2008.

[29] J. Nolte, "The human brain: an introduction to its functional anatomy," 2002.

[30] J. Yan, Y. Li, W. Liu, H. Zha, X. Yang, and S. M. Chu, "Graduated consistency-regularized optimization for multi-graph matching," in *European Conference on Computer Vision*. Springer, 2014, pp. 407–422.

[31] H.-M. Park and K.-J. Yoon, "Encouraging second-order consistency for multiple graph matching," *Machine Vision and Applications*, vol. 27, no. 7, pp. 1021–1034, 2016.

[32] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[33] B. N. Flury and W. Gautschi, "An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal

form," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 1, pp. 169–184, 1986.

[34] A. Ziehe, P. Laskov, G. Nolte, and K.-R. MÃžller, "A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation," *Journal of Machine Learning Research*, vol. 5, no. Jul, pp. 777–800, 2004.

[35] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[36] W. Tang, Z. Lu, and I. S. Dhillon, "Clustering with multiple graphs," in *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on.* IEEE, 2009, pp. 1016–1021.

[37] T. G. Kolda, "Numerical optimization for symmetric tensor decomposition," *Mathematical Programming*, vol. 151, no. 1, pp. 225–248, 2015.

[38] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," *Neural, Parallel & Scientific Computations*, vol. 11, no. 4, pp. 351–368, 2003.

[39] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.

[40] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent

type methods," *SIAM journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.

[41] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[42] N. Bell and M. Garland, "Implementing sparse matrix-vector multiplication on throughput-oriented processors," in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*. ACM, 2009, p. 18.

[43] C. J. Hillar and L.-H. Lim, "Most tensor problems are np-hard," *Journal of the ACM (JACM)*, vol. 60, no. 6, p. 45, 2013.

[44] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM journal on matrix analysis and applications*, vol. 30, no. 2, pp. 713–730, 2008.

[45] M. Aharon, M. Elad, and A. Bruckstein, "*rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[46] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[47] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements

via orthogonal matching pursuit," *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[48] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2, 1995, pp. 1137–1145.

[49] J. Neyman and E. L. Scott, "Consistent estimates based on partially consistent observations," *Econometrica: Journal of the Econometric Society*, pp. 1–32, 1948.

[50] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[51] S. N. Dorogovtsev, A. V. Goltsev, J. F. Mendes, and A. N. Samukhin, "Spectra of complex networks," *Physical Review E*, vol. 68, no. 4, p. 046109, 2003.

[52] D. Koutra, J. T. Vogelstein, and C. Faloutsos, "D elta c on: A principled massive-graph similarity function," in *Proceedings of the SIAM International Conference in Data Mining. Society for Industrial and Applied Mathematics*. SIAM, 2013, pp. 162–170.

[53] R. Arden, R. S. Chavez, R. Grazioplene, and R. E. Jung, "Neuroimaging creativity: a psychometric view," *Behavioural brain research*, vol. 214, no. 2, pp. 143–156, 2010.

BIBLIOGRAPHY

[54] M. Brant-Zawadzki, G. D. Gillan, and W. R. Nitz, "Mp rage: a three-dimensional, t1-weighted, gradient-echo sequence–initial experience in the brain." *Radiology*, vol. 182, no. 3, pp. 769–775, 1992.

[55] R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman *et al.*, "An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest," *Neuroimage*, vol. 31, no. 3, pp. 968–980, 2006.

[56] G. Kiar, W. Gray Roncal, D. Mhembere, E. Bridgeford, R. Burns, and J. Vogelstein, "ndmg: Neurodata's mri graphs pipeline," Aug. 2016, open-source code. [Online]. Available: http://dx.doi.org/10.5281/zenodo.60206

[57] T. M. Amabile, "The social psychology of creativity: A componential conceptualization." *Journal of personality and social psychology*, vol. 45, no. 2, p. 357, 1983.

[58] S. Suwan, D. S. Lee, R. Tang, D. L. Sussman, M. Tang, C. E. Priebe *et al.*, "Empirical bayes estimation for the stochastic blockmodel," *Electronic Journal of Statistics*, vol. 10, no. 1, pp. 761–782, 2016.

[59] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[60] D. Steinley, "Properties of the hubert-arable adjusted rand index." *Psychological methods*, vol. 9, no. 3, p. 386, 2004.

[61] E. Rendón, I. Abundez, A. Arizmendi, and E. Quiroz, "Internal versus external cluster validation indexes," *International Journal of computers and communications*, vol. 5, no. 1, pp. 27–34, 2011.

[62] R. I. Jennrich, "Asymptotic properties of non-linear least squares estimators," *The Annals of Mathematical Statistics*, vol. 40, no. 2, pp. 633–643, 1969.

[63] P. J. Bickel and K. A. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics, volume I.* CRC Press, 2015, vol. 117, ch. 6.

[64] C. Davis and W. M. Kahan, "The rotation of eigenvectors by a perturbation. iii," *SIAM Journal on Numerical Analysis*, vol. 7, no. 1, pp. 1–46, 1970.

[65] R. Govindan and H. Tangmunarunkit, "Heuristics for internet map discovery," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2000, pp. 1371–1380.

[66] D. M. Da Zheng, R. Burns, J. Vogelstein, C. E. Priebe, and A. S. Szalay, "Flashgraph: Processing billion-node graphs on an array of commodity ssds," in *Proceedings of the 13th USENIX Conference on File and Storage Technologies*, 2015, pp. 45–58.

[67] J. Fan, R. Samworth, and Y. Wu, "Ultrahigh dimensional feature selection: beyond the linear model," *Journal of Machine Learning Research*, vol. 10, no. Sep, pp. 2013–2038, 2009.

[68] J. Fan, R. Song *et al.*, "Sure independence screening in generalized linear models with np-dimensionality," *The Annals of Statistics*, vol. 38, no. 6, pp. 3567–3604, 2010.

[69] G. J. Székely, M. L. Rizzo, N. K. Bakirov *et al.*, "Measuring and testing dependence by correlation of distances," *The annals of statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[70] G. J. Székely, M. L. Rizzo *et al.*, "Brownian distance covariance," *The annals of applied statistics*, vol. 3, no. 4, pp. 1236–1265, 2009.

[71] G. J. Székely and M. L. Rizzo, "The distance correlation t-test of independence in high dimension," *Journal of Multivariate Analysis*, vol. 117, pp. 193–213, 2013.

[72] C. Shen, C. E. Priebe, M. Maggioni, and J. T. Vogelstein, "Discovering relationships and their structures across disparate data modalities," *https://arxiv.org/abs/1609.05148*, 2017.

[73] C. Shen, C. E. Priebe, and J. T. Vogelstein, "From distance correlation to multiscale generalized correlation," *https://arxiv.org/abs/1710.09768*, 2017.

[74] Y. Lee, C. Shen, , and J. T. Vogelstein, "Network dependence testing via diffusion maps and distance-based correlations," *https://arxiv.org/abs/1703.10136*, 2017.

[75] L.-P. Zhu, L. Li, R. Li, and L.-X. Zhu, "Model-free feature screening for ultrahigh-dimensional data," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1464–1475, 2011.

[76] R. Li, W. Zhong, and L. Zhu, "Feature screening via distance correlation learning," *Journal of the American Statistical Association*, vol. 107, no. 499, pp. 1129–1139, 2012.

[77] L. Devroye, L. Györfi, and G. Lugosi, *A probabilistic theory of pattern recognition*. Springer Science & Business Media, 2013, vol. 31.

[78] C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park, "Scan statistics on enron graphs," *Computational & Mathematical Organization Theory*, vol. 11, no. 3, pp. 229–247, 2005.

[79] D. Cartwright and F. Harary, "Structural balance: a generalization of heider's theory." *Psychological review*, vol. 63, no. 5, p. 277, 1956.

[80] M. Zhu and A. Ghodsi, "Automatic dimensionality selection from the scree plot via the use of profile likelihood," *Computational Statistics & Data Analysis*, vol. 51, no. 2, pp. 918–930, 2006.

[81] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[82] P. Robert and Y. Escoufier, "A unifying tool for linear multivariate statistical methods: the rv-coefficient," *Applied statistics*, pp. 257–265, 1976.

[83] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[84] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[85] S. Ogawa, T.-M. Lee, A. R. Kay, and D. W. Tank, "Brain magnetic resonance imaging with contrast dependent on blood oxygenation," *Proceedings of the National Academy of Sciences*, vol. 87, no. 24, pp. 9868–9872, 1990.

[86] J. Qiu, Z. Qinglin, T. Bi, G. Wu, D. Wei, and W. Yang, "Southwest university longitudinal imaging multimodal (slim) brain data repository: A long-term test-retest sample of young healthy adults in southwest china." [Online]. Available: http://dx.doi.org/10.15387/fcpindi.retro.slim

[87] X. Weng and X. Zuo, "One-month test-retest reliability and dynamical resting-state study." [Online]. Available: http://dx.doi.org/10.15387/fcpindi.corr.hnu1

[88] A. Badea, G. A. Johnson, and R. Williams, "Genetic dissection of the mouse

brain using high-field magnetic resonance microscopy," *Neuroimage*, vol. 45, no. 4, pp. 1067–1079, 2009.

[89] E. Calabrese, A. Badea, G. Cofer, Y. Qi, and G. A. Johnson, "A diffusion mri tractography connectome of the mouse brain and comparison with neuronal tracer data," *Cerebral Cortex*, vol. 25, no. 11, pp. 4628–4637, 2015.

[90] G. A. Johnson, A. Badea, J. Brandenburg, G. Cofer, B. Fubara, S. Liu, and J. Nissanov, "Waxholm space: an image-based reference for coordinating mouse brain research," *Neuroimage*, vol. 53, no. 2, pp. 365–372, 2010.

[91] B. B. Avants, N. J. Tustison, G. Song, P. A. Cook, A. Klein, and J. C. Gee, "A reproducible evaluation of ants similarity metric performance in brain image registration," *Neuroimage*, vol. 54, no. 3, pp. 2033–2044, 2011.

[92] F.-C. Yeh, T. D. Verstynen, Y. Wang, J. C. Fernández-Miranda, and W.-Y. I. Tseng, "Deterministic diffusion fiber tracking improved by quantitative anisotropy," *PloS one*, vol. 8, no. 11, p. e80713, 2013.

[93] S. Spring, J. P. Lerch, and R. M. Henkelman, "Sexual dimorphism revealed in the structure of the mouse brain using three-dimensional magnetic resonance imaging," *Neuroimage*, vol. 35, no. 4, pp. 1424–1433, 2007.

[94] A. Raznahan, F. Probst, M. R. Palmert, J. N. Giedd, and J. P. Lerch, "High

resolution whole brain imaging of anatomical variation in xo, xx, and xy mice," *Neuroimage*, vol. 83, pp. 962–968, 2013.

[95] Q. K. Telesford, A. R. Morgan, S. Hayasaka, S. L. Simpson, W. Barret, R. A. Kraft, J. L. Mozolic, and P. J. Laurienti, "Reproducibility of graph metrics in fmri networks," *Frontiers in neuroinformatics*, vol. 4, 2010.

[96] R. Tang, M. Ketcha, J. T. Vogelstein, C. E. Priebe, and D. L. Sussman, "Law of large graphs," *arXiv preprint arXiv:1609.01672*, 2016.

[97] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American statistical association*, vol. 58, no. 301, pp. 13–30, 1963.

[98] L. Reiter, O. Rinner, P. Picotti, R. Hüttenhain, M. Beck, M.-Y. Brusniak, M. O. Hengartner, and R. Aebersold, "mprophet: automated data processing and statistical validation for large-scale srm experiments," *Nature methods*, vol. 8, no. 5, pp. 430–435, 2011.

[99] M. D. Fox, A. Z. Snyder, J. L. Vincent, M. Corbetta, D. C. Van Essen, and M. E. Raichle, "The human brain is intrinsically organized into dynamic, anticorrelated functional networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 27, pp. 9673–9678, 2005.

[100] B. B. Biswal, M. Mennes, X.-N. Zuo, S. Gohel, C. Kelly, S. M. Smith, C. F. Beckmann, J. S. Adelstein, R. L. Buckner, S. Colcombe *et al.*, "Toward discov-

ery science of human brain function," *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4734–4739, 2010.

[101] P. E. Shrout and J. L. Fleiss, "Intraclass correlations: uses in assessing rater reliability." *Psychological bulletin*, vol. 86, no. 2, p. 420, 1979.

[102] S. C. Strother, J. Anderson, L. K. Hansen, U. Kjems, R. Kustra, J. Sidtis, S. Frutiger, S. Muley, S. LaConte, and D. Rottenberg, "The quantitative evaluation of functional neuroimaging experiments: the npairs data analysis framework," *NeuroImage*, vol. 15, no. 4, pp. 747–771, 2002.

[103] M. L. Rizzo, G. J. Székely *et al.*, "Disco analysis: A nonparametric extension of analysis of variance," *The Annals of Applied Statistics*, vol. 4, no. 2, pp. 1034–1055, 2010.

[104] X.-N. Zuo, C. Kelly, J. S. Adelstein, D. F. Klein, F. X. Castellanos, and M. P. Milham, "Reliable intrinsic connectivity networks: test–retest evaluation using ica and dual regression approach," *Neuroimage*, vol. 49, no. 3, pp. 2163–2177, 2010.

[105] U. Braun, M. M. Plichta, C. Esslinger, C. Sauer, L. Haddad, O. Grimm, D. Mier, S. Mohnke, A. Heinz, S. Erk *et al.*, "Test–retest reliability of resting-state connectivity network characteristics using fmri and graph theoretical measures," *Neuroimage*, vol. 59, no. 2, pp. 1404–1412, 2012.

[106] H. Shou, A. Eloyan, S. Lee, V. Zipunnikov, A. Crainiceanu, M. Nebel, B. Caffo, M. Lindquist, and C. Crainiceanu, "Quantifying the reliability of image replication studies: the image intraclass correlation coefficient (i2c2)," *Cognitive, Affective, & Behavioral Neuroscience*, vol. 13, no. 4, pp. 714–724, 2013.

[107] C. Yue, S. Chen, H. I. Sair, R. Airan, and B. S. Caffo, "Estimating a graphical intra-class correlation coefficient (gicc) using multivariate probit-linear mixed models," *Computational statistics & data analysis*, vol. 89, pp. 126–133, 2015.

[108] B. Yu *et al.*, "Stability," *Bernoulli*, vol. 19, no. 4, pp. 1484–1500, 2013.

[109] Q. Li, J. B. Brown, H. Huang, and P. J. Bickel, "Measuring reproducibility of high-throughput experiments," *The annals of applied statistics*, pp. 1752–1779, 2011.

[110] B. W. Silverman, *Density estimation for statistics and data analysis*. CRC press, 1986, vol. 26.

[111] S. C. Strother, "Evaluating fmri preprocessing pipelines," *IEEE Engineering in Medicine and Biology Magazine*, vol. 25, no. 2, pp. 27–41, 2006.

[112] X. Liang, J. Wang, C. Yan, N. Shu, K. Xu, G. Gong, and Y. He, "Effects of different correlation metrics and preprocessing factors on small-world brain functional networks: a resting-state functional mri study," *PloS one*, vol. 7, no. 3, p. e32766, 2012.

[113] M. Hampson, B. S. Peterson, P. Skudlarski, J. C. Gatenby, and J. C. Gore, "Detection of functional connectivity using temporal correlations in mr images," *Human brain mapping*, vol. 15, no. 4, pp. 247–262, 2002.

[114] M. P. Van Den Heuvel and H. E. H. Pol, "Exploring the brain network: a review on resting-state fmri functional connectivity," *European Neuropsychopharmacology*, vol. 20, no. 8, pp. 519–534, 2010.

[115] P. T. Costa and R. R. MacCrae, *Revised NEO personality inventory (NEO PI-R) and NEO five-factor inventory (NEO FFI): Professional manual.* Psychological Assessment Resources, 1992.

[116] D. C. Van Essen, K. Ugurbil, E. Auerbach, D. Barch, T. Behrens, R. Bucholz, A. Chang, L. Chen, M. Corbetta, S. W. Curtiss *et al.*, "The human connectome project: a data acquisition perspective," *Neuroimage*, vol. 62, no. 4, pp. 2222–2231, 2012.

[117] V. N. Vapnik and V. Vapnik, *Statistical learning theory.* Wiley New York, 1998, vol. 1.

[118] S. A. Huettel, A. W. Song, and G. McCarthy, *Functional magnetic resonance imaging.* Sinauer Associates Sunderland, 2004, vol. 1.

[119] A. Klein, J. Andersson, B. A. Ardekani, J. Ashburner, B. Avants, M.-C. Chiang, G. E. Christensen, D. L. Collins, J. Gee, P. Hellier *et al.*, "Evaluation of 14

nonlinear deformation algorithms applied to human brain mri registration," *Neuroimage*, vol. 46, no. 3, pp. 786–802, 2009.

[120] A. M. Smith, B. K. Lewis, U. E. Ruttimann, Q. Y. Frank, T. M. Sinnwell, Y. Yang, J. H. Duyn, and J. A. Frank, "Investigation of low frequency drift in fmri signal," *Neuroimage*, vol. 9, no. 5, pp. 526–533, 1999.

[121] J. D. Power, K. A. Barnes, A. Z. Snyder, B. L. Schlaggar, and S. E. Petersen, "Spurious but systematic correlations in functional connectivity mri networks arise from subject motion," *Neuroimage*, vol. 59, no. 3, pp. 2142–2154, 2012.

[122] M. D. Fox, D. Zhang, A. Z. Snyder, and M. E. Raichle, "The global signal and observed anticorrelated resting state brain networks," *Journal of neurophysiology*, vol. 101, no. 6, pp. 3270–3283, 2009.

[123] R. C. Craddock, G. A. James, P. E. Holtzheimer, X. P. Hu, and H. S. Mayberg, "A whole brain fmri atlas generated via spatially constrained spectral clustering," *Human brain mapping*, vol. 33, no. 8, pp. 1914–1928, 2012.

[124] J. L. Andersson, M. Jenkinson, S. Smith *et al.*, "Non-linear registration, aka spatial normalisation fmrib technical report tr07ja2," *FMRIB Analysis Group of the University of Oxford*, vol. 2, 2007.

[125] B. B. Avants, N. Tustison, and G. Song, "Advanced normalization tools (ants)," *Insight J*, vol. 2, pp. 1–35, 2009.

[126] S. Sikka, B. Cheung, R. Khanuja, S. Ghosh, C. Yan, Q. Li, J. Vogelstein, R. Burns, S. Colcombe, C. Craddock *et al.*, "Towards automated analysis of connectomes: The configurable pipeline for the analysis of connectomes (c-pac)," in *5th INCF Congress of Neuroinformatics, Munich, Germany*, vol. 10, 2014.

[127] R. A. Fisher, *Statistical methods for research workers.* Genesis Publishing Pvt Ltd, 1925.

[128] J. F. Hair, "Multivariate data analysis," 2009.

[129] C.-F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis, "Processing and visualization for diffusion tensor mri," *Medical image analysis*, vol. 6, no. 2, pp. 93–108, 2002.

[130] S. Mori, S. Wakana, P. C. Van Zijl, and L. Nagae-Poetscher, *MRI atlas of human white matter.* Elsevier, 2005.

[131] R. Paley and A. Zygmund, "On some series of functions,(3)," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 28, no. 02. Cambridge Univ Press, 1932, pp. 190–205.

[132] P. A. Devijver and J. Kittler, *Pattern recognition: A statistical approach.* Prentice hall, 1982.

Vita

Shangsi Wang was born in 1989 in Beijing, China. He attended the Beijing National Day School from 2002 to 2008. He then studied in the Faculty of Mathematics at University of Waterloo. In the spring of 2012, he received Bachelor of Mathematics in Honours Actuarial Science and Honours Pure Mathematics; he graduated with Distinction - Dean's Honours List. He next enrolled in the Ph.D. program in the Department of Applied Mathematics and Statistics at Johns Hopkins University.

At Johns Hopkins University, Shangsi quickly gained an interest in statistics and machine learning. He was fortunate enough to work under the supervision of Dr. Carey E. Priebe in the area of statistical inference on graphs. In addition to research, he was passionate about teaching mathematics and served as a teaching assistant for a few classes. In 2015, he won the Joel Dean Award for Excellence in Teaching. Beginning in the spring of 2018, Shangsi will be working as a quantitative researcher at an investment company in New York City.