STATISTICAL METHODS TO ANALYZE MASSIVE HIGH-DIMENSIONAL NEUROIMAGING DATA

by

Shaojie Chen

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

Baltimore, Maryland

August, 2015

Abstract

The statistical analysis of neuroimaging data poses several challenges today, partly due to their size, high dimensionality and noise. In this work, we address three different methods for analyzing massive, high-dimensional and noisy functional magnetic resonance images (fMRI) data. In the first method, parallel computing techniques are combined with an independent component analysis (ICA) algorithm to decompose resting state fMRI data. The algorithm's performance is greatly improved compared to existing methods. In the second method, a graphical model, referred to as state space model (SSM) is extended by enforcing L-1 and L-2 penalties on parameters. The model scales well to very high dimensions and can be applied to a vast class of different neuroimaging analysis applications. In the third method, a two-stage method is developed to extract information from noisy fMRI data. We first use functional regression to extract features from fMRI data and then use the features to predicts physical pains that human feels. A support vector machine (SVM) is trained for prediction and it achieves high prediction accuracy.

Advisor:

ABSTRACT

Brian Caffo, PhD

Committee:

James Pekar, PhD (chair, SOM radiology & radiological science)

Brian Caffo, PhD (advisor, SPH biostatistics)

Michelle Carlson, PhD (SPH mental health & epidemiology)

Martin Lindquist, PhD (SPH biostatistics)

Alternates:

Gregory Pontone, MD (SOM psychiatry)

Ani Eloyan, PhD (SPH biostatistics)

Jonathan Links, PhD (SPH environmental health)

Acknowledgments

First I would like to thank my advisor, Brian Caffo, who has been a great friend, a great advisor and great mentor. He has been a constant source of advice and encouragement. His nice personality and endless f ow of ideas have made it a great pleasure to work with him. I have learned from him how to be a good biostatistical researcher, and more importantly, how to be a great person.

Thanks to my thesis committee for their advice over the years: James Pekar, Michelle Carlson, Martin Lindquist, Gregory Pontone, Ani Eloyan and Jonathan Links.

Thanks to Joshua Vogelstein and Ciprian Crainiceanu for your great advice on my different projects.

Thanks to all the SMARTies. It is so great to be part of the group. The group is like a big family and is always full of great ideas and projects.

Warmest thanks to the students of the biostatistics department who overlapped with my time here. Special thanks to Chen Yue, Lei Huang, Huitong Qiu, Detian Deng and Yuting Xu: you guys are so amazing that I could not imagine what my PhD life will be without you. I am grateful to the students that began the program before me for their help on every

aspect of my research and career (especially Juemin Yang and Shanshan Li).

Finally, I would say thanks to my family. Thanks to my parents who have been supporting me all the time on every decision I made. They do not speak English and might never know what I am writing here, but I am sure they can feel it on the other side of the planet. Thanks to my brother, Shujie Chen, who has always been a last resort for inspiration and strength when I got confused and don't know what to do with my life. I feel so lucky to have been born in such a great family. Thanks to Yao, my f ancee, for all the care, support and everything we have experienced together.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Statistical challenges in neuroimaging data analysis

There has been fast growth in the number of neuroimaging studies performed using functional Magnetic Resonance Imaging (fMRI) in recent years. A standard fMRI study gives rise to huge amounts of noisy data with complicated spatio-temporal correlation structures. Statistics plays an important role in understanding the nature of this massive, high-dimensional, and noisy data and obtaining results that can be interpreted by neuroscientists.

Three properties of fMRI data make it challenging to analyze with statistical methods: data size, high-dimensionality and noise. As mentioned above, fMRI data is massive and is increasing in size as large multi-subject observational studies become the norm. Existing statistical methods fail when applied to these massive data sets, as they do not scale well to the increased number of subjects. One solution that we exploit is to take advantage of growing computing capacity, in particular by leveraging parallel computing in multi-processing/core settings. Functional MRI data is also high dimensional - there can be as many as hundreds of thousands of voxels per time point in an fMRI image. Most of the commonly used statistical models have limited power when analyzing high-dimensional data, due to the curse of dimensionality. New statistical and computational methods need to be developed to overcome these shortcomings. Finally, fMRI data is noisy. It is diff cult to extract useful information that can be interpreted by biomedical researchers from them.

This research aims to develop new tools and overcome the above challenges, as outlined

in the next section.

1.2 Organizational overview

In this research, three new methods are developed to tackle the above mentioned challenges in neuroimaging data analysis.

In the first method, we propose a two-stage likelihood-based algorithm for performing group ICA, which we denote as Parallel Group Independent Component Analysis (PGICA). By utilizing the sequential nature of the algorithm and parallel computing techniques, we are able to analyze datasets from a large numbers of subjects efficiently. We illustrate the efficacy of PGICA with simulation studies and application to rs-fMRI data. Two large multi-subject data sets, consisting of 301 and 779 subjects respectively, are analyzed. The algorithm has been implemented in R and is freely available through the Comprehensive R Archive Network (CRAN).

In the second method, we developed a penalized linear dynamical system (PLDS) by generalizing the linear dynamical system (LDS) model to high-dimensional setting and introducing L-1 and L-2 penalties on model parameters. An Expectation-Maximization algorithm is also developed for efficient estimation of the model parameters. PLDS is useful in decomposing neuroimaging data and find the connectivity among the components. To illustrate our approach, we apply it to test/re-test fMRI data measured over the motor cortex and the Human Connectome Project (HCP) data.

In the third method, a functional regression model is built to extract features from noisy high-dimensional time series and then a support vector machine (SVM) is trained with the features for physical pain prediction with fMRI data. The two-stage method connects environment stimuli to neuroimaging signals.

1.3 Software

The first method is publicly available on Comprehensive R Archive Network (CRAN) now, with package name PGICA. The second method is implemented in Matlab and publicly available on Github, with toolbox name PLDS.

Chapter 2

A Parallel Group Independent Component Analysis Algorithm for Massive fMRI Data Analysis

Abstract

Independent component analysis (ICA) is widely used in the f eld of functional neuroimaging to decompose data into spatio-temporal patterns of co-activation. In particular, it has found wide usage in the analysis of resting state fMRI (rs-fMRI) data. Recently, a number of large-scale data sets have become publicly available that consist of rs-fMRI scans from thousands of subjects. Unfortunately, currently used ICA algorithms fail when applied to these massive data sets, as they do not scale well to the increased number of subjects. To circumvent this problem, we propose a two-stage likelihood-based algorithm for performing group ICA, which we denote Parallel Group Independent Component Analysis (PGICA). By utilizing the sequential nature of the algorithm and parallel computing techniques, we are able to eff ciently analyze data sets from large numbers of subjects. We illustrate the eff cacy of PGICA, which has been implemented in R and is freely available through the Comprehensive R Archive Network, through simulation studies and application to rs-fMRI data from two large multi-subject data sets, consisting of 301 and 779 subjects respectively.

Keywords: signal processing, parallel computing, ICA, functional MRI

2.1 Introduction

Independent component analysis (ICA) is a blind source separation technique (Jutten and Herault, 1991) that assumes the observed signals are linear mixings of independent underlying sources. A framework for using ICA to make group inferences from functional Magnetic Resonance Imaging (fMRI) data was first introduced by Calhoun et al. (2001). A major methodological contribution of this work was the circumvention of the permutation ambiguity of ICA by eliminating the requirement to match components across subjects. Since its introduction, ICA has become an extremely popular approach to analyzing fMRI data, as it does not require the a priori definition of a hemodynamic response function or seed regions of interest and is able to capture both spatial and temporal inter-subject variability (Koch et al., 2010; Michael et al., 2014). Several algorithms have been developed to estimate parameters in ICA (Beckmann and Smith, 2005; Guo and Pagnoni, 2008). However, concerns have recently been raised about the scalability of the group ICA approach (Smith et al., 2014). With the neuroscience community taking cues from the the crowdsourcing model of labor and encouraging the public distribution of large collections of data including thousands of subjects collected at multiple sites, the development of algorithms for analyzing such high dimensional data is imperative.

A common starting point for most group ICA approaches is the singular value decomposition (SVD). While the SVD is a means for avoiding the estimation of an overdetermined system, it is also the means for throwing away massive amounts of data through repeated application (as desribed by Smith et al., 2014). A notable exception is the work

by Eloyan et al. (2013), which does not require repeated SVD steps to be scalable. Gaussian distributional assumptions can provide little insight to further explore the data, and we are motivated to search for components that are as non-Gaussian as possible. The densities of the underlying components in the algorithm proposed by Eloyan et al. (2013) are approximated with f nite mixtures of smooth densities, while the time courses for each subject are updated using a gradient-based optimization algorithm. A Quasi-Newton algorithm is used for optimization to estimate the parameters in the mixing matrix.

In this paper, we propose a more direct solution to the scalability issue described by Smith et al. (2014) by building upon the two-stage likelihood-based algorithm proposed by Eloyan et al. (2013) and use parallel computing techniques to improve algorithmic performance for large groups of observations. The algorithm proposed by Eloyan et al. (2013), is scalable, but performs calculations serially. We decompose the problem into computationally unrelated tasks and distributed over a parallel computing system. The proposed Parallel Group Independent Component Analysis (PGICA) is different from fastICA and JADE in that the algorithm is likelihood-based and uses MLE for parameter estimation. Compared to the ML implementation of ICA by Bell and Sejnowski (1995), PGICA does not require a highly restricted likelihood. Instead, f exible mixtures of Gaussian densities are used to approximate the densities of the underlying components. Another advantage of PGICA is its ability to analyze massive data. Current ICA algorithms have limited power for scaling to analyze large data sets, especially in the f eld of resting state fMRI analysis. The current standard is thus to throw away massive amounts of data with repeated applica-

tions of the SVD (e.g., as described by Smith et al., 2014). PGICA can handle hundreds to thousands of subjects simultaneously with the help of parallel computing. Many parallel programming environments exist that provide basic tools, language features and application programming interfaces (APIs) needed to construct a parallel program. Widely used environments include: OpenMP (thread-level parallelization), MPI (cluster-level) and CUDA / OpenCL (GPGPU-level). The RSGE package in the R software provides an interface to perform cluster-level parallel programming on Sun Grid Engines (SGE) (Bode, 2012) and the SNOW package can be used for thread-level parallel computing (Tierney et al., 2012). In newer versions of R ($\geq 2.14.0$), the package *parallel* is included in its core, which provides drop-in replacements for most of the functionalities of *snow*. The R package we built for this work is based on package *parallel*. At the end, we illustrate the performance of PGICA by applying it to rs-fMRI data from two large multi-subject data sets. The first is a collection of 301 adults, while the second is a set of 779 fMRI scans, consisting of 379 with autism spectrum disorder (ASD) and 400 typically developing controls.

2.2 Materials and Methods

2.2.1 The ICA model

A general term that indexes a broad class of models, ICA has several algorithmic implementations and theoretical foundations, but the linear factor analytic model with the

assumption of independent underlying factors is the primary commonality of all ICA algorithms (Harman, 1976). In this paper, we focus on noise-free ICA, a version of ICA which only requires an "unmixing" of the input data matrix. (Thus, the noise in the data is absorbed into the estimated independent components.)

Suppose that for each subject i, $i = 1, \ldots, I$, a $T \times V$ dimensional matrix is observed. In the neuroimaging context, the rows represent time points and the columns represent voxels. Let $\mathbf{X}_i(t, v)$ represent row t, column v of $\mathbf{X}_i$. (The same notational convention applies to other vectors and matrices.) The noise-free group ICA decomposition model can be expressed as follows.

$$\mathbf{X}_i(t, v) = \sum_{q=1}^{Q} \mathbf{A}_i(t, q)\mathbf{S}(q, v), \tag{2.2.1}$$

for $i = 1, \ldots, I$. This model assumes that the spatio-temporal process, $X_i(t, v)$, for each subject, i, can be decomposed into a f nite sum of products between subject-specif c time series, $A_i(t, q)$, and subject-independent spatial maps, $S(q, v)$. Let $\mathbf{X} = [\mathbf{X_1^T}...\mathbf{X_I^T}]^\mathbf{T}$ and $\mathbf{A} = [\mathbf{A_1^T}...\mathbf{A_I^T}]^\mathbf{T}$ be the $IT \times V$ and $IT \times Q$ matrices obtained by stacking the $\mathbf{X_i}$ and $\mathbf{A_i}$ respectively, then the above model is equivalent to $\mathbf{X} = \mathbf{AS}$. In the fMRI context, one often interprets $\mathbf{S}(q, \cdot)$ as brain networks and $\mathbf{A}_i$ as subject specif c temporal mixing matrices (Calhoun et al., 2001).

As a technical consideration, (2.2.1) maybe overdetermined. So we f rst preprocess the data at subject level via an singular value decomposition (SVD) on the observed matrices and remains only the f rst Q components for each subject. This f rst-step SVD is unavoidable. Henceforth, we assume that the number of component to estimation is equal to time

points, i.e. $Q = T$. The square matrices $\mathbf{A}_i$ are further assumed to be of full rank, hence one can def ne the inverse of these matrices as $\mathbf{W}_i = \mathbf{A}_i^{-1}$ and the densities of the underlying components as $f_1, ..., f_Q$. Thus, for a given q, $\{\mathbf{S}(q,v)\}_{v=1}^{V}$ can be considered as V iid draws from f_q.

2.2.2 Parameter Estimation

The likelihood of the above model can be written as

$$L(\mathbf{W}, \mathbf{f}) = \prod_{i=1}^{I} \prod_{v=1}^{V} \prod_{q=1}^{Q} f_q\Big(\sum_{l=1}^{Q} w_{iql} x_{ilv}\Big) |\det(W_i)|, \tag{2.2.2}$$

If the f_q were known, any optimization algorithm could be used to obtain the MLE of $\mathbf{W}_i$. However, since the densities of the underlying components are unknown, an iterative algorithm must be implemented that alternates between density estimation and estimation of the $\mathbf{W}_i$. This manuscript uses mixture density estimates (MDE) introduced by Eloyan and Ghosh (2011). Specif cally, we parameterize the densities as:

$$f_q(s) = \sum_{j=1}^{J_q} \theta_{qj} \frac{1}{\sigma_q} \phi\left(\frac{s - \mu_{qj}}{\sigma_q}\right), \tag{2.2.3}$$

where $\phi(\cdot)$ is the standard normal density function. The number of densities in the mixture $J_q = 1 + \frac{2}{3}\text{Range}_v\{\mathbf{S}(q,v)\}$ is chosen empirically. Similarly, $\mu_{qj} = \min_v \mathbf{S}(q,v) + \frac{j-1}{J_q - 1}\text{Range}_v\{\mathbf{S}(q,v)\}$ for $j = 1, \cdots, J_q$. The underlying rationale behind this is to set the

means μ_{qj} as an equally spaced grid between the extremes of the data so that the distance between the means decreases as J_q increases and to set σ_q^2 such that σ_q decreases as J_q increases. Denote $\mathcal{M}_{J_q} = \{\mu_{q1} < ... < \mu_{qJ_q}\}$. The value of J_q is allowed to vary in different iterations; as J_q increases, the set $\mathcal{M}_{J_q+1}$ is constructed by adding the median of one of the intervals $[\mu_{q,j}, \mu_{q,j-1}]$. More details on the choice of the mean sequence and the variance are presented by Eloyan and Ghosh (2011).

Since the underlying independent components are the same for all subjects, the length of the vector $\mathbf{S}(q,\cdot)$ depends only on the number of non-background voxels. In most fMRI studies $\mathbf{S}(q,\cdot)$ has a large sample size ($\approx 70,000$ voxels for example), hence nonparametric estimation of the density can be problematic. To address this issue, Eloyan et al. (2013) proposed a binning algorithm for the density estimation, essentially looking at the approximation to the histogram of the data. With this binning procedure, the weights of the mixture densities in equation (2.2.3) given by $(\theta_{q1}, ..., \theta_{qJ_q})$ are estimated using a constrained EM algorithm. The resulting density estimates satisfy the moment constraints required for full identif ability of the model by $E[\mathbf{S}(q,\cdot)] = 0$, $E[\mathbf{S}(q,\cdot)^2] = 1$, $0 < E[\mathbf{S}(1,\cdot)^3] < \ldots < E[\mathbf{S}(Q,\cdot)^3]$, for $q = 1, \cdots, Q$. Given the density estimation above as $\widehat{f}_1, \cdots, \widehat{f}_Q$, the likelihood function of matrix $\mathbf{W}$ can be constructed as

$$L(\mathbf{W}, \widehat{\mathbf{f}}) = \sum_{i=1}^{I} \{ \sum_{v=1}^{V} \sum_{q=1}^{Q} [\widehat{f}_q(\sum_{l=1}^{Q} w_{iql} x_{ilv})] + V \log|\det \mathbf{W}_i| \}, \qquad (2.2.4)$$

where $\widehat{f}_q(s) = \sum_{j=1}^{J_q} \widehat{\theta}_{qj} \frac{1}{\sigma_q} \phi\left(\frac{s-\mu_{qj}}{\sigma_q}\right)$. The maximum of $(2.2.4)$ can be found by Quasi-

Newton algorithm. The algorithm proceeds by iterating between the estimation of $\widehat{f}$ and $\mathbf{W}$ until convergence. The complete algorithm pseudo code for f tting PGICA is given below.

PGICA

For each iteration M

1 Let $\mathbf{S}_i^{(M)} = \mathbf{W}_i^{(M-1)}\mathbf{X}_i$, for each $i = 1, ..., I$.

2 For each Independent Component q construct the set of midpoints $M_{q1}, ..., M_{qp}$.

of the bins and the corresponding counts $c_{q1}, c_{q2}, ..., c_{qp}$.

3 For each $q = 1, ..., Q$, construct the set of means $\mathcal{M}_{J_q^{(M)}} \supset \mathcal{M}_{J_q{}^{(M-1)}}$

and the variance component σ_q.

4 Estimate $(\theta_{q1}^{(M)}, ..., \theta_{qJ_q^{(M)}}^{(M)})$ using MDE.

5 For each $i = 1, ..., I$, compute the gradient $L'(\widehat{\mathbf{W}_i}^{(M)})$ and hessian matrix

$L''(\widehat{\mathbf{W}_i}^{(M)})$ **in parallel.**

6 For each $i = 1, ..., I$, update the unmixing matrix

$\widehat{\mathbf{W}_i}^{(M+1)} = \widehat{\mathbf{W}_i}^{M} - L''(\widehat{\mathbf{W}_i}^{(M)})^{-1}L'(\widehat{\mathbf{W}_i}^{(M)})$.

7 $\delta = \max |\widehat{\mathbf{W}_i}^{(M+1)} - \widehat{\mathbf{W}_i}^{M}|$. If $\delta > \epsilon$ return to step 1.

In the above algorithm, Step 5 is the most time-consuming. Fortunately, the structure of the likelihood in Equation $(2.2.4)$ makes it possible to simplify computations. Note that the likelihood is a product of the likelihoods of multiple subjects. Thus after taking logs,

the gradients for different W_i s do not depend on each other. As a consequence, one can calculate the gradients and Hessians in parallel. According to Amdahl's law (Amdahl, 1967), the theoretical speedup obtainable using parallelization is speedup= $\frac{1}{\frac{P}{N}+S}$, where P is the parallel proportion of the computations, S is serial proportion of the computations and N is the number of processors. Here P and S differ when the sizes of input data differ. The parallel proportion increases with the number of subjects. It encompasses more than 90% of the theoretical time for 300 or more subjects. Of course, the practical speedup will not be exactly the same as the theoretical one due to many factors such as messages passing overhead; see Section 2.4 for more information.

2.2.3 1000 Functional Connectomes Project Data

First, PGICA was applied to data from the 1,000 Functional Connectomes Project, which consists of thousands of resting state scans combined across multiple sites with the goal of facilitating discovery and analysis of brain networks (Biswal et al., 2010). The quality and scanning parameters vary across sites. Thus, we focus on data from two sites that each provided a large number of scans: Cambridge and Oulu. We include 301 subjects in the analyses presented below, 198 are from Cambridge and 103 from Oulu. As discussed above, directly applying currently used group ICA methods to data of this size is computationally infeasible for regular computers due to limitations of memory and running time. As such, it provides an important test case for PGICA.

Scanning parameters used to acquire the data from each site are detailed elsewhere (for

complete information see http://fcon_1000.projects.nitrc.org/fcpClassic/FcpTable.html). Each subject's data consisted of either 119 time points collected every 3 s or 245 time points collected every 1.8 s. Note that even though the number of time points varies across subjects, the algorithm can still be applied, as the first PCA step reduces the dimensions of each dataset to be the same. However, the variance related consequences of including data with varying scan lengths and sampling frequencies remain an open topic. All scans were collected using a 3T scanner. The data were preprocessed using the processing scripts available on the NITRC website (www.nitrc.org/projects/fcon_1000/). Anatomical images were de-obliqued, reoriented, and skull stripped, while the functional scans were de-obliqued, reoriented, motion corrected, skull stripped, grand mean scaled, temporal bandpass filtered, and de-trended (linear and quadratic). Functional scans were registered to anatomical scans using FLIRT in FSL Smith et al. (2004). The structural scans were registered to the Montreal Neurological Institute (MNI) space using FLIRT and the transformation was subsequently applied to the functional scans. A mask based on the MNI template is used to separate the background of the images. For each time point, the 3D array is vectorized to obtain a V dimensional vector of intensities that are then concatenated over time. Hence we obtain a $T \times V$ dimensional matrix $\mathbf{X_i}$ for each subject. PGICA is then applied to these $\mathbf{X_i}$ matrices.

2.2.4 Autism Brain Imaging Data Exchange

Next, PGICA was applied to data from the Autism Brain Imaging Data Exchange (ABIDE) consortium, a collaboration between 17 imaging centers to openly share existing resting state fMRI scans with corresponding structural MRI and phenotypic information. In total, the database consists of 539 individuals with ASD and 573 age-matched typical controls (Di Martino et al., 2014). Site-specif c protocols for recruitment and image acquisition are available online (http://fcon_1000.projects.nitrc.org/indi/abide); in short, 5 to 10 minutes of rs-fMRI data collected using repetition times (TR) between 1.5 s and 3 s were shared for each subject. The f rst 10 s of each resting state scan were ignored to allow for magnetization stabilization. Resting state scans were then slice-time adjusted using the slice acquired in the middle of the TR, and rigid body realignment parameters were estimated with respect to the f rst (stabilized) functional volume. An iterative process previously described by Nebel et al. (2014a) was used to coregister and normalize the structural and functional images to MNI space. Each resting state scan was then temporally detrended on a voxel-wise basis and spatially smoothed (2-mm FWHM Gaussian kernel). Finally, each resting state scan was downsampled by randomly sampling 67,749 of the 229,263 non-background voxels to reduce computation demands. Downsampling the voxels is only performed to estimate starting values of the parameters for initialization of the algorithm, but is not necessary for the algorithm itself. The FSL package was used to smooth the original NIFTI images (Smith et al., 2004).

As opposed to the f rst application presented in this paper, we found that a much larger

subset of the data can be used for simultaneous analysis due to the data quality and consistency across the sites. Because they made up a low percentage of the total number of subjects ($\sim 10\%$), girls were excluded from the analysis. Age was restricted to individuals between 6 and 40 years old. Individuals with framewise displacement more than two standard deviations away from the mean were also excluded from the analysis. The data collected at the Kennedy Krieger site was also excluded from the analysis for comparison of the results in future studies. As a result, scans for 779 subjects are analyzed in this application, 400 typical controls, 379 individuals with ASD. The histograms of age, the intelligence quotient (IQ), and the social responsiveness scores (SRS) are shown in Figure 2.1.

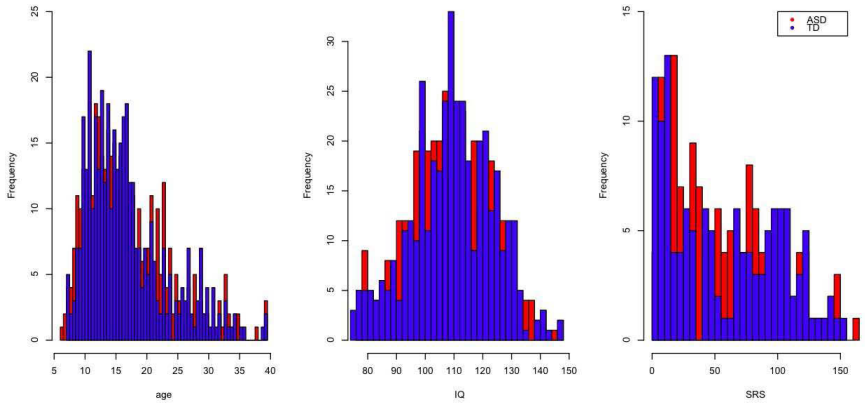


Figure 2.1: Histograms of age (left), IQ (middle), and SRS (right) for participants in ABIDE plotted and colored by disease diagnosis and overlaid, where blue corresponds to TD controls and red corresponds to ASD.

2.2.5 Simulation Studies

To demonstrate the validity of the proposed method and compare the accuracy of the parameter estimates with the commonly used fastICA algorithm we considered simulated data using two different simulation scenarios. We considered various shapes in the underlying independent components to estimate the accuracy of prediction of the brain networks in the imaging context. The first four shapes shown in Figure 2.2 are used in the first scenario, while all 8 underlying signals are used for the second simulation. The fastICA method (Hyvärinen and Oja, 1997) is used as a comparison. The mixing matrices for each subject are predefined in each simulation example. The underlying sources are generated for 100 simulation runs as described below. The observed matrices for each subject are then computed and fastICA and PGICA are used to estimate the mixing matrices for each subject and the underlying sources. Finally, the correlations of each component with the true underlying sources are calculated. Ideally, these correlations should be equal to 1 if the networks are perfectly estimated. For each example, we averaged the correlations for all the underlying components for each of the simulation runs and presented the boxplots of logarithms of the correlations for better visualization for each method in each simulation scenario to compare the results. The goal of the simulation studies is to compare the parameter estimates in high dimensional settings and demonstrate the performance of the proposed method in estimating the parameters. The real data examples show the power of the proposed method to perform group ICA in settings where other algorithms would fail because of the dimensionality of the data.

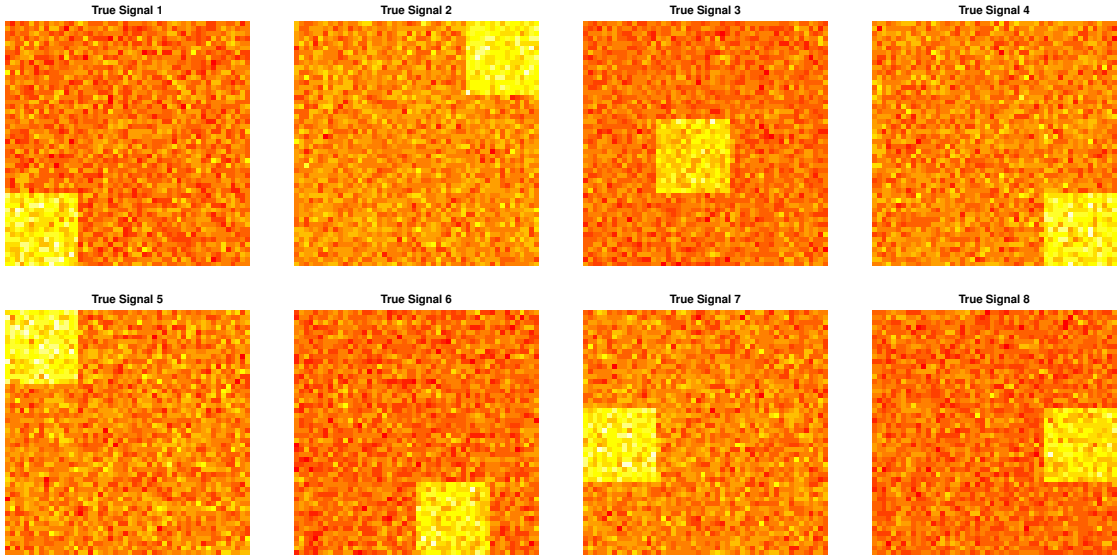


Figure 2.2: True signals for the simulation examples. Each component is a two dimensional array where the pixels in a square have higher intensities than the rest of the array. A random noise is added to each of the components at all pixels.

Simulation 1: Suppose there are 4 subjects and 4 underlying sources, i.e. $I = 4$ and $Q = 4$. Only 4 subjects are included in this simulation study so that all model generating parameters can be included in the paper for reproducibility. The data are generated from the group ICA model $\mathbf{X_i} = \mathbf{A_i S}$, with $T = Q = 4$ and $V = 2500$ where the independent components are the first 4 signals in Figure 2.2. The four mixing matrices are defined as follows.

$$A_1 = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 3 & 3 & 1 & .5 \\ 1 & 2 & 2 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix}, A_2 = \begin{pmatrix} 2 & 3 & 2 & 1 \\ 3 & 4 & 1 & .5 \\ 3 & 2 & 3 & 4 \\ 2 & 3 & 3 & 1 \end{pmatrix}, A_3 = \begin{pmatrix} 1 & 2 & 2 & 1 \\ 3 & 4 & 1 & .5 \\ 3 & -1 & 3 & 4 \\ 2 & 1 & 3 & 1 \end{pmatrix}, A_4 = \begin{pmatrix} 3 & 2 & 2 & -1 \\ 3 & 3 & 2 & 1 \\ 3 & 1 & 1 & 4 \\ 1 & 1 & 4 & .5 \end{pmatrix}.$$

The boxplots of the average correlations across four independent components for each of the 100 simulation runs are shown in Figure 2.3 while the summary statistics of the estimated correlations are presented in Table 2.1. The two methods perform similarly to each other with PGICA performing marginally better than fastICA.

Simulation 2: In this example, we assume that the number of subjects is 50 while the number of underlying components is 8, $I = 50$, $Q = 8$. The data are, again, generated from the group ICA model $\mathbf{X_i} = \mathbf{A_i S}$, with $T = Q = 8$ and $V = 2500$ where the independent components are the signals in Figure 2.2. Here, the components 4 and 6 were deliberately generated so that the "activated" regions in the two components are spatially overlapping, however, the signals are statistically independent.

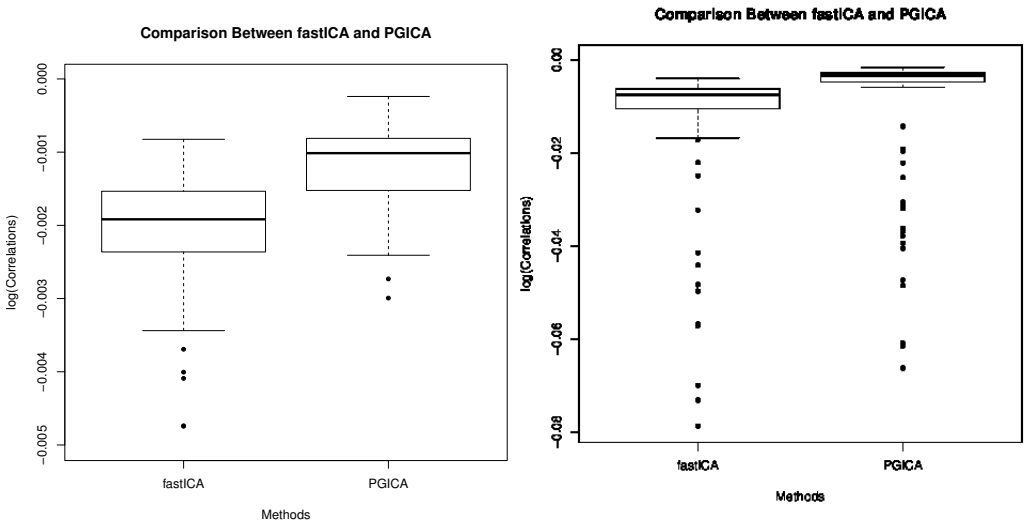


Figure 2.3: Boxplots (for both fastICA and PGICA) of the average correlations (log-transformed) of the true signals with the estimated signals from simulation 1 on the left and simulation 2 on the right.

The results of the 100 simulation examples shown in Figure 2.3 and Table 2.1 demonstrate that the correlations of the estimated components with the true underlying signals

using the proposed PGICA method are signif cantly better than those estimated using the conventional fastICA algorithm.

Table 2.1: Summary measures of the correlations in the two simulation examples.

		min	1st quantile	median	3rd quantile	max
Sim 1	fastICA	0.9953	0.9977	0.9981	0.9985	0.9992
	PGICA	0.9944	0.9985	0.9990	0.9992	0.9998
Sim 2	fastICA	0.9243	0.9895	0.9924	0.9938	0.9960
	PGICA	0.9359	0.9952	0.9966	0.9972	0.9984

2.3 Results

2.3.1 1,000 Functional Connectomes Project Data

Following the design of group ICA analysis described by Biswal et al. (2010), group ICA was used to obtain $Q = 20$ components for the 301 subjects in the 1,000 Functional Connectomes Project Dataset. Figure 2.4 shows axial, sagittal, and frontal planes of four of the estimated networks by PGICA: auditory, control, default mode, and visual. The estimated networks are thresholded at (5%) and the map is overlaid on a grayscale template MNI image. The networks shown in this example were identif ed visually as a proof of concept exercise. The estimated networks have clear edges and less noise in the areas that are not a part of the networks showing the importance of estimating the networks using larger datasets. Figure 2.5 shows three dimensional renderings of the same networks shown in Figure 2.4 colored in red and overlaid on an opaque template image conf rming that the

estimated networks are more noise-free.

The increase in speed when using PGICA as compared to non-parallel version of the algorithm called HDICA (high dimensional ICA) as the number of subjects increases is shown in Table 2.2. The memory usage of HDICA increases linearly with the number of subjects (memory usage = number of subjects $\times$ single subject), while the memory usage of PGICA remains constant as the number of subjects increases (memory usage = single subject). For PGICA, each slave computer only calculates the gradient for a single subject, as long as we have enough slave computers. In practice, total memory usage is several times higher than just the input data. Thus memory usage of HDICA quickly goes beyond the ability of even super computers, making it incapable of dealing with large groups of observations.

Table 2.2: Speed increase of PGICA

# of subjects	1	10	50	150	300
Non-Parallel GICA time (min)	20	400	4000	12000	NA
PGICA time (min)	20	80	592	2000	4100

In this example, 15 computing clusters were used for estimation using the PGICA on a Sun Grid Engine (SGE). All computations are performed on clusters with the same or very similar hardware properties such as speed and age.

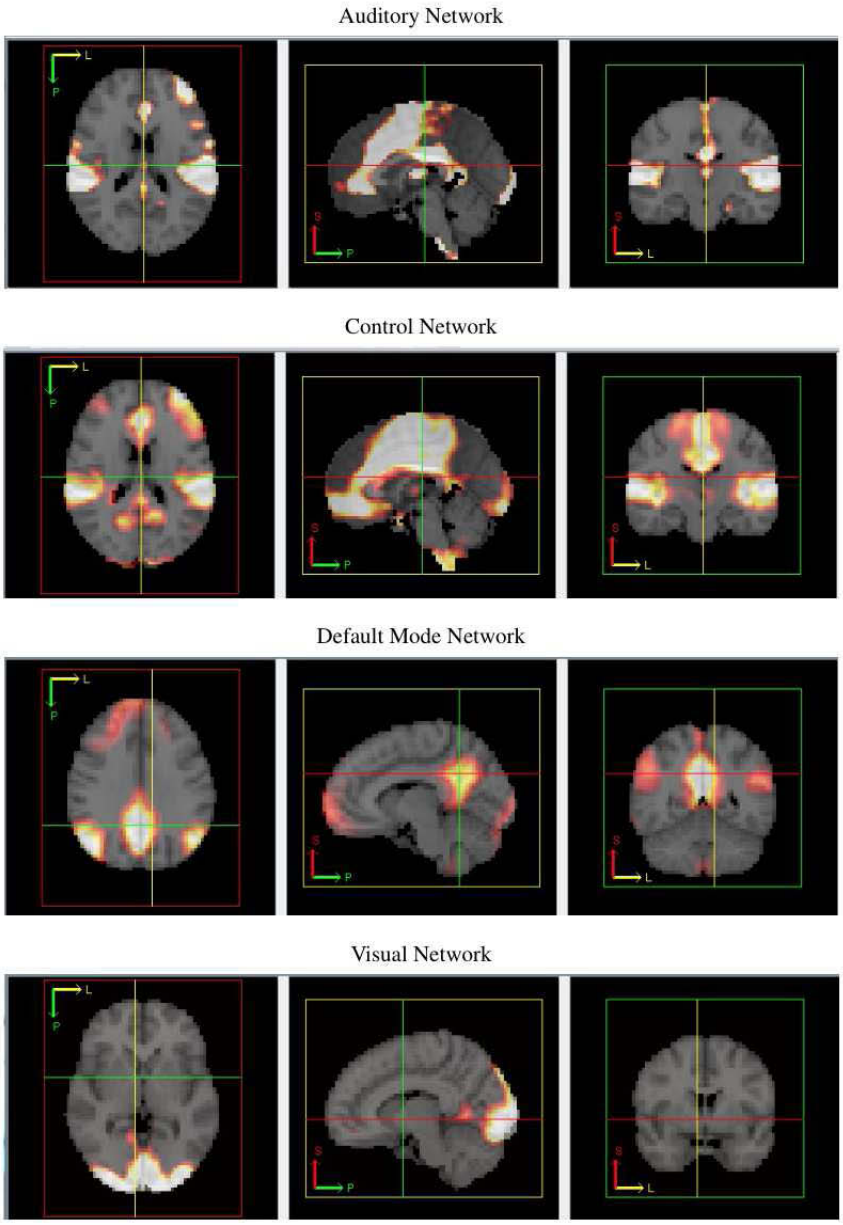


Figure 2.4: Axial, sagittal, and frontal (left to right) planes of the auditory, control, default mode and visual networks (from top) estimated using 301 fMRI scans from the 1,000 Functional Connectomes Project dataset. The thresholded maps are overlaid on a greyscale MNI template brain. The 90th slice is shown from the MNI template in each of the plots. The colors correspond to the intensities in the estimated brain networks where white: high intensity to red: low intensities.

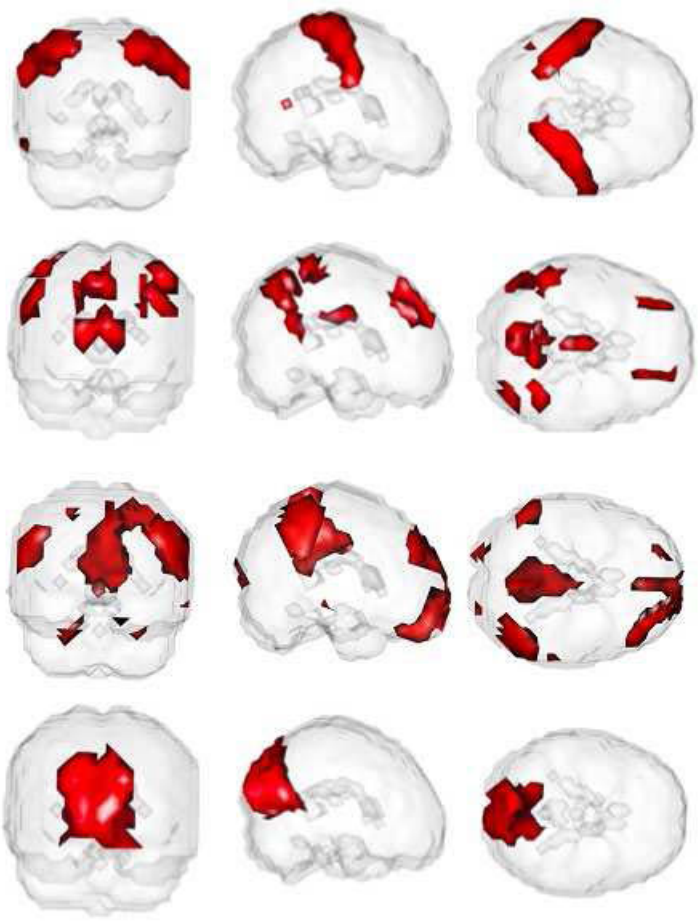


Figure 2.5: 3D view of auditory, control, default mode and visual networks (from top).

2.3.2 Autism Brain Imaging Data Exchange

Similar to the data analysis performed for the 1000 FCP data in Section 2.3.1, we estimated $Q = 20$ components using the fMRI scans for 779 participants in the ABIDE sample. The networks were identified and labeled by finding the closest network given by Allen et al. (2011) in terms of maximizing the correlation between the estimated network and the networks identified by Allen et al. (2011). Six networks were identified: control,

auditory, right executive, attention, default mode network, and visual network. Examples of the identified networks are shown in Figure 2.6. The networks demonstrate clear edges again as are well estimated further demonstrating the ability of the proposed method to estimate ICs for such high dimensional data.

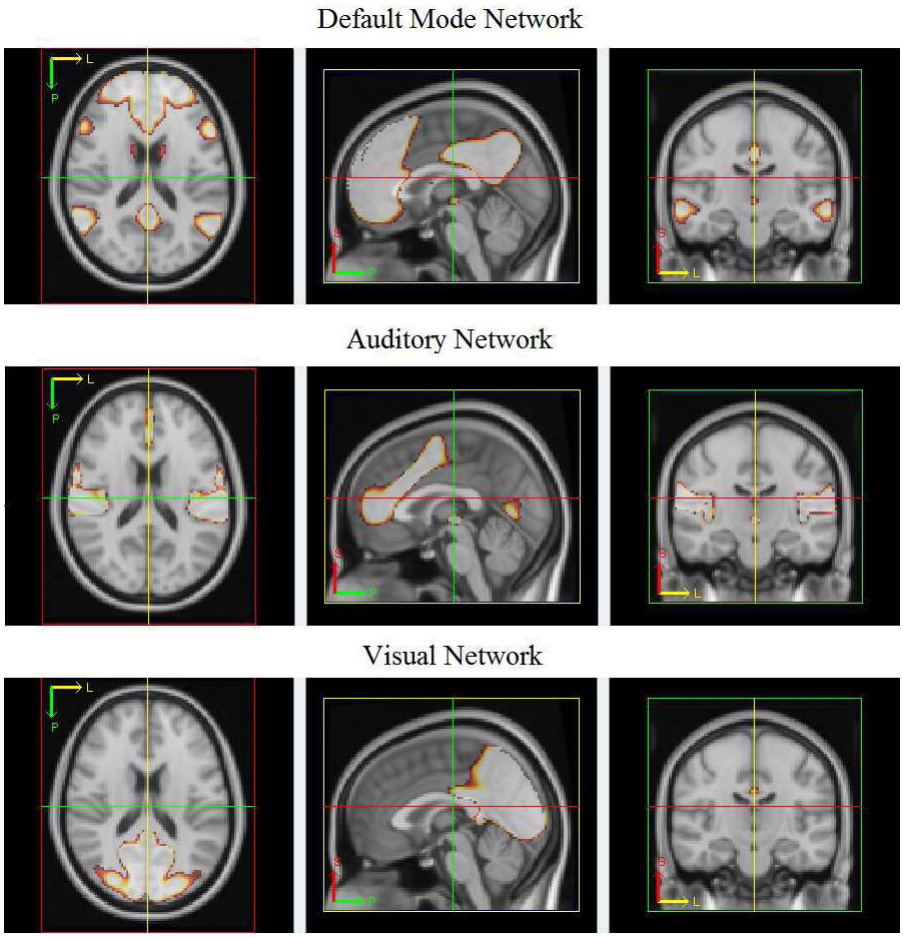


Figure 2.6: Axial, sagittal, and frontal (left to right) planes of the default mode, auditory and visual networks (from top) estimated using 779 fMRI scans from the ABIDE dataset. The thresholded maps are overlaid on a greyscale MNI template brain. The colors correspond to the intensities in the estimated brain networks where white: high intensity to red: low intensities.

This example is one of the few direct runs of group ICA for rs-fMRI in the literature for such high dimensional data. One of the largest group ICA runs we identified in the

literature is presented by Allen et al. (2011) and is based on rs-fMRI data for 603 healthy adolescents and adults. In most cases, the dataset is split into subsets, group ICA is applied to each subset, and the results are aggregated after the individual group ICA runs (e.g. Biswal et al., 2010). The uniqueness of the proposed algorithm is the application to the whole dataset directly which can provide new insights for group comparisons without the necessity of splitting the groups into parts. In addition, the algorithm can be applied to data with even more number of subjects barring any issues with data quality.

2.4 Discussion

In this paper, we extended the group ICA algorithm of Eloyan et al. (2013) using high-performance computing. The new PGICA algorithm can analyze large-scale data efficiently. Essentially, the sequential nature of the algorithm turns a memory-intensive, constant-time computing problem into a constant-memory, time-intensive problem, and then uses parallel computing to turn the resulting time-intensive problem into a constant time problem. With this algorithm, two large resting-state fMRI datasets were analyzed. An interesting byproduct of this work is a comprehensive brain network atlas from over 300 healthy adults and one based on 779 scans including both individuals with ASD and control subjects.

Although the method of Eloyan et al. (2013) is theoretically scalable in terms of its memory requirements, the approach requires the serial calculation of gradients to opti-

mize parameter estimation, which can be very slow for high-dimensional data to the point where it would still be practically infeasible in terms of computation time. Computing the gradients for different subjects in parallel could potentially speed up the algorithm dramatically, provided the cost is much lower than the necessary data transfer time. Parallel programming has been widely utilized in scientif c computing since the 1950s (Mattson et al., 2004). According to Flynn's taxonomy (Flynn, 1972), most current computers are Multiple Instruction Multiple Data (MIMD) systems. MIMD computers are typically categorized as shared-memory, distributed-memory or hybrid systems. In a shared-memory system, all processes share addressable memory and communicate via shared variables. In distributed-memory systems, such as supercomputers and clusters, one process communicate with others through message passing. A supercomputer's processors and the network infrastructure are tightly coupled and specialized for parallelization. In contrast, clusters are composed of off-the-shelf computers connected by an off-the-shelf network. Recently, General-purpose computing on graphics processing units, or GPGPU, is developing fast and provides a new scheme for parallel computing. In this work, the PGICA algorithm is performed on both shared-memory and distributed-memory systems. It can be implemented to f t other parallel computing schemes in the future.

We used an extensive simulation study to validate the accuracy of the proposed algorithm in high dimensional settings. The simulation studies show that the proposed PGICA algorithm performs as well as a commonly used method for ICA for small sample sizes, while the performance is signif cantly better as the number of subjects in the study in-

creases. The information provided on the computation time gain is presented using the real data example as the purpose of the simulation studies was to assess accuracy rather than required computation time. In principle, given enough nodes the algorithm can be applied to a dataset with any number of subjects and the simulation results indicate that the accuracy of the results will improve with the increased number of subjects under the assumption of no biologically irrelevant systematic differences between subgroups in the data.

Large, freely available multisite datasets such as the 1000 FCP and ABIDE are invaluable for a number of reasons including accelerating neuroimaging discovery science and providing a means to validate neuroimaging f ndings through replication. However, these datasets also contain some inherent limitations. Each participating data collection site was motivated by its own research questions, leading to potentially large inconsistencies in acquisition parameters, subject populations, and research protocols across sites that may limit our ability to estimate networks and our sensitivity to detect biologically meaningful group differences. We did not analyze the 1000 Functional Connectome Project dataset in its entirety, as there are site-specif c variations, which plague the quality of results. More specif cally, functional imaging data collected in each data collection site have different features, such as population demographics, scanner types, data quality and so on. The data in each site have been collected for addressing specif c research questions introducing issues while analyzing the data collectively. The factors for different sites interfere when analyzing data together. Thus we have found a degradation in the quality of results as more data is included. In this paper, focus lied on only two sites: Cambridge and Oulu, where

site-specif c effects presented no major issue. For future work, aggregating methods to properly combine data from different sites are needed. This example shows that estimating networks using data for ever more subjects can result in highly precise estimates of the networks. However, if the variability between the scans for the subjects in the data is very high (especially due to biologically unrelated reasons), it can obscure the results instead of improving the estimates.

The functional imaging scans in the ABIDE dataset, while still collected in various data collection sites, was more homogeneous when analyzing the data together. We analyzed a subset of 779 fMRI scans simultaneously in this paper. The networks identif ed in this example can be used as a powerful tool for exploring possible differences in network engagement over time between the two groups: ASD and TD, using the second level analyses as described by Joel et al. (2011). The estimates of ICs in the ABIDE example presented assume common spatial maps for all subjects in the study including those with ASD and their TD peers. A question still remains whether the spatial networks are the same between the two groups or whether two sets of networks should be estimated using two instances of group ICA. The proposed method can be used to test the hypothesis of signif cant differences between spatial networks of each group. In this example, we used a downsampling approach to estimate the starting values of the parameters for our model. While not implicitly stated by the proposed method, the voxel intensities in the observed images are assumed to be statistically independent. The assumption may be violated when the voxel sizes are very small where the correlations between neighboring voxels may not be small

enough to be ignored. Hence, as the spatial resolution of images improves a more thorough investigation of the effect of spatial correlations on the parameter estimates is necessary. It is interesting to note that the regions comprising networks def ned using ABIDE are generally more diffuse than those def ned using the 1000 FCP set which can be seen more strikingly for the DMN and visual networks which could suggest differences between networks based on data including both ASD and TD children as well as differences between adults and children. The networks identif ed using the proposed method can be used to investigate these questions further.

Chapter 3

A Sparse High Dimensional State-Space Model with an Application to Neuroimaging Data

Abstract

In the past decade functional magnetic resonance imaging (fMRI) has facilitated major advances in our understanding of human brain function. The data that arise from a standard fMRI experiment are both high dimensional and complex in nature, making statistical analysis challenging. Matrix decomposition methods, such as factor analysis, principal component analysis (PCA) and independent component analysis (ICA), are commonly used to investigate spatio-temporal patterns present in fMRI data. It can be shown that the linear time-invariant state-space model, commonly used in time series analysis, unifies this broad class of models. While state-space models have been applied to fMRI data, these applications have been limited by constraints on the amount of data that can be included in the analysis. This is primarily because analysis in modern high-dimensional settings, such as neuroimaging, parameter estimation is challenging. This issue is addressed by introducing a penalized state-space model that applies L-1 and L-2 penalties to model coefficients. In addition, an Expectation-Maximization algorithm is provided that allows for efficient estimation of the model parameters. To illustrate our approach, we apply it to fMRI data measured over the motor cortex.

***keywords*: state-space model, parameter estimation, sparsity, high dimensional, imaging processing, fMRI**

3.1 Introduction

In the past decade functional magnetic resonance imaging (fMRI) has given researchers unprecedented access to the brain in action and provided numerous insights into human brain function. Any given fMRI experiment generates massive amounts of data. For example, a standard experiment collects a few hundred 3D brain images, each consisting of roughly 100,000 uniformly spaced volume elements (voxels) that partition the brain. Intensity values from each individual voxel can be extracted to create a set of time series of length T, where T corresponds to the number of acquired images. The analysis of fMRI data can therefore fruitfully be viewed as a multivariate time series problem. However, the signal of interest is relatively weak and the data exhibits a complicated temporal and spatial noise structure Lindquist et al. (2008).

To date numerous statistical methods have been applied to fMRI data. Many construct separate univariate models at each voxel, thus assuming an improbable independence between voxels. In this work we instead focus on the multivariate statistical methods that have been used to analyze fMRI data. In particular, multivariate decomposition methods, such as Principal Components Analysis (PCA) Andersen et al. (1999) and Independent Components Analysis (ICA) Calhoun et al. (2009), have been utilized to identify patterns of brain activation McKeown et al. (1998).

Interestingly, several of these commonly applied statistical techniques for modeling both multivariate data can be seen as variants of state-space models (SSMs). For example, according to Roweis and Ghahramani Roweis and Ghahramani (1999), factor analysis,

principal component analysis (PCA), mixture of Gaussian clusters, independent component analysis (ICA), Kalman f lter models and hidden Markov models (HMMs) can all be viewed as special cases of SSMs.

In the time-invariant linear case, an SSM is also referred to as a linear dynamical system (LDS) or linear Gaussian model (LGM). In this work, LDS and its extensions are discussed, so we will use LDS and SSM interchangeably in the following sections. The LDS can be seen as a continuous-state analogue of the hidden Markov model (HMM) Rabiner and Juang (1986). The forward step of the forward-backward algorithm used to inference HMMs is equivalent to the well-known Kalman f lter used in LDS, and similarly the backward step can be computed using Rauchs recursion Rauch (1963). Together these two steps can be employed to perform inference on the posterior probabilities of latent states given the observed sequence.

Likewise, factor analysis and PCA can each be derived from the LDS by applying particular constraints on the latent states dynamics coeff cients and the observation error covariance matrix. Specif cally, by constraining the latent states dynamics coeff cients to $\mathbf{0}$, one gets a static model. Factor analysis can be implemented by further constraining the observation error covariance matrix to be diagonal. PCA can be applied by forcing the observation error covariance matrix to be a multiple of the identity matrix approaching $\mathbf{0}$. A corresponding detailed review can be found in Roweis and Ghahramani Roweis and Ghahramani (1999).

Finally, LDS can also be represented as a probabilistic graphical model. Here the

Kalman f lter and smoother are special cases of the belief propagation algorithm that has been developed to analyze general graphical models Lauritzen and Spiegelhalter (1988)Pearl (1988).

Because of their f exibility state-space models have found wide usage in a number of different spheres, including time series analysis, statistics, signal processing, control theory and machine learning. In neuroimaging analysis, the LDS exhibits substantial relevance. For example, Harini et al have discussed the applications of HMM in learning functional network dynamics in resting state fMRI Eavani et al. (2013). Valdez-Sosa et al used sparse multivariate autoregression to estimate brain functional connectivity Valdés-Sosa et al. (2005). Havlicek et al modeled neuronal responses in fMRI using cubature Kalman f ltering along with Kalman f lter based Dynamic Granger Causality to evaluate functional connectivity in fMRI data Havlicek et al. (2011). A systematic framework for functional connectivity measures is proposed by HE Wang et al Wang et al. (2014).

In this work, a penalized linear dynamical system model (PLDS) is proposed as an generalization of the generic LDS model. An Expectation-Maximization (EM) algorithm is also developed for parameter estimations. Compared to the generic LDS model, PLDS is highly scalable and yields more accurate estimations and predictions under some circumstances. The generic LDS model is just a special case of PLDS with zero penalties. As an application, the PLDS model is applied to fMRI data measured over the motor cortex.

3.2 The Model

The generic time-invariant state-pace model, or LDS, can be written as:

$$\begin{aligned} \mathbf{x}_{t+1} &= A\mathbf{x}_t + \mathbf{w}_t, \quad \mathbf{w}_t \sim N(\mathbf{0}, Q), \quad \mathbf{x}_0 \sim N(\pi_0, V_0) \\ \mathbf{y}_t &= C\mathbf{x}_t + \mathbf{v}_t, \qquad \mathbf{v}_t \sim N(\mathbf{0}, R) \end{aligned} \tag{3.2.1}$$

where A is the $d \times d$ state transition matrix and C is the $p \times d$ generative matrix. $\mathbf{x}_t$ is a $d \times 1$ vector and $\mathbf{y}_t$ is a $p \times 1$ vector. The sequence of vectors $\{\mathbf{y}\} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ are the observed data and $\{\mathbf{x}\} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ represent the unknown hidden states. The output noise covariance R is $p \times p$, while the state noise covariance Q is $d \times d$. Initial state mean π_0 is $d \times 1$ and covariance V_0 is $d \times d$.

Without applying further constraints, the model itself is unidentif able and too general to be useful. Supplemental constraints are are thus introduced to address both identif ability and utility. Three basic constraints are required to make the model identif able:

Constraint 1: Q is the identity matrix

Constraint 2: the ordering of the columns of C is f xed based on their norms

Constraint 3: $V_0 = \mathbf{0}$

Note that the f rst two constraints follow directly from Roweis and Ghahramani (1999) Roweis and Ghahramani (1999).

The logic for Constraint 1 is as follows. Since Q is a covariance matrix, it is symmetric

and positive semidef nite and thus can be expressed in the form $E\Lambda E^T$ where E is a rotation matrix of eigenvectors and Λ is a diagonal matrix of eigenvalues. Thus, for any model where Q is not the identity matrix, one can generate an equivalent model using a new state vector $\mathbf{x}' = \Lambda^{-1/2}E^T\mathbf{x}$ with $A' = (\Lambda^{-1/2}E^T)A(E\Lambda^{1/2})$ and $C' = C(E\Lambda^{1/2})$ such that the new covariance of $\mathbf{x}'$ is the identity matrix, i.e., $Q' = \mathbf{I}$. Thus one can constrain $Q = \mathbf{I}$ without loss of generality.

For Constraint 2, the components of the state vector can be arbitrarily reordered; this corresponds to swapping the columns of C and A. Therefore,the order of the columns of matrix C must be f xed. We follow Roweis and Ghahramani and choose the order by decreasing the norms of columns of C.

Additionally, V_0 is set to zero, meaning the starting state $\mathbf{x}_0 = \pi_0$ is an unknown constant instead of a random variable, since there is only a single chain of time series in the neuroimaging application. To estimate V_0 accurately, multiple series of observations are required.

The following three new constraints are further applied to achieve a more useful model.

Constraint 4: R is a diagonal matrix

Constraint 5: A is sparse

Constraint 6: C has smooth columns

Consider the case where the observed data are high dimensional and the R matrix is very large. One can not accurately estimate the many free parameters in R with limited

observed data. Therefore some constraints on R are necessary. In the simplest case, R is set to an identity matrix or its multiple. More generally, one can also constrain matrix R to be diagonal. In the static model with no temporal dynamics, a diagonal R is equivalent to the generic Factor Analysis method, while multiples of the identity R matrix lead to Principal Component Analysis (PCA) Roweis and Ghahramani (1999).

The A matrix is the transition matrix of the hidden states. In our application, it is a central construct of interest representing a so-called connectivity graph. In many applications, it is desirable for this graph to be sparse. In this work, an L-1 penalty term on A is used to impose sparsity on the connectivity graph..

Similarly, for many applications, one wants the columns of C to be smooth. For example, in the neuroimaging data analysis of section 6, each column of C is a signal in the primary motor cortex. Having those signals spatially smooth allows capturing the active regions within the motor cortex. In this context, an L-2 penalty term on C is used to enforce smoothness.

With all those constraints, the model becomes:

$$\begin{aligned} &\mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{w}_t, \quad \mathbf{w}_t \sim N(\mathbf{0}, \mathbf{I}), \quad \mathbf{x}_0 = \pi_0, \quad A \text{ is sparse} \\ &\mathbf{y_t} = C\mathbf{x}_t + \mathbf{v}_t, \qquad \mathbf{v}_t \sim N(\mathbf{0}, R), \quad C \text{ has smooth columns} \end{aligned} \tag{3.2.2}$$

For notational convenience, a sequence of T output vectors $(\mathbf{y}_1, \ldots, \mathbf{y}_T)$ is denoted by $\{\mathbf{y}\}$; a subsequence $(\mathbf{y}_{t_0}, \mathbf{y}_{t_0+1}, \ldots, \mathbf{y}_{t_1})$ by $\{\mathbf{y}\}_{t_0}^{t_1}$. Similarly for the latent states. In addition, let $\Theta = \{A, C, R, \pi_0\}$ represents all unknown parameters and $P(\{\mathbf{x}\}, \{\mathbf{y}\})$ be

the likelihood for a generic LDS model, then model 3.2 is equivalent to

$$\widehat{\Theta} = \arg\min_{\Theta} \left\{ -\log P(\{\mathbf{x}\}, \{\mathbf{y}\}) + \lambda_1 \|A\|_1 + \lambda_2 \|C\|_2^2 \right\} \tag{3.2.3}$$

where λ_1 and λ_2 are tuning parameters and $\| \cdot \|_p$ represents the p-norm of a vector.

3.3 Parameter Estimation

The motivating application requires parameter estimation in model 3.2: given only an observed sequence (or multiple sequences in some applications) of outputs $\{\mathbf{y}\} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$, find the parameters $\Theta = \{A, C, R, \pi_0\}$ that maximize the likelihood of the observed data.

Parameter estimation for LDS has been investigated extensively by researchers from control theory, signal processing, machine learning and statistics. For example, in machine learning, the exact and variational learning algorithms are developed for general Bayesian networks. In control theory, the corresponding area of study is known as system identification, which identifies parameters in continuous state models.

Specifically, one way to find the maximum likelihood solution is through iterative techniques such as expectation maximization (EM) Shumway and Stoffer (1982). The detailed EM steps for a generic LDS can be found in Zoubin and Geoffrey (1996) Ghahramani and Hinton (1996). An alternative approach is to use subspace identification methods such as N4SID and PCA-ID to compute an asymptotically unbiased solution in closed form Van Overschee and De Moor

(1994) Doretto et al. (2003). In practice, determining an initial solution with subspace identif cation and then ref ning the solution with EM is a worthwhile approach Boots (2008).

However, the above solutions can not be directly applied to model 3.2 due to the introduced penalty terms. A new algorithm is then developed and detailed as follows.

By the chain rule, the likelihood in model 3.2 is

$$P(\{\mathbf{x}\},\{\mathbf{y}\}) = P(\mathbf{x_0})\prod_{t=1}^{T} P(\mathbf{x_t}|\mathbf{x_{t-1}})\prod_{t=1}^{T} P(\mathbf{y_t}|\mathbf{x_t}) = \prod_{t=1}^{T} P(\mathbf{x_t}|\mathbf{x_{t-1}})\prod_{t=1}^{T} P(\mathbf{y_t}|\mathbf{x_t})\mathbb{1}_{\pi_0}(\mathbf{x}_0)$$

where $\mathbb{1}_{\pi_0}(\mathbf{x}_0)$ is the indicator function and conditional likelihoods are

$$P(\mathbf{y_t}|\mathbf{x_t}) = \exp\left\{-\frac{1}{2}[\mathbf{y_t} - C\mathbf{x_t}]' R^{-1}[\mathbf{y_t} - C\mathbf{x_t}]\right\}(2\pi)^{-p/2}|R|^{-1/2}$$
$$P(\mathbf{x_t}|\mathbf{x_{t-1}}) = \exp\left\{-\frac{1}{2}[\mathbf{x_t} - A\mathbf{x_{t-1}}]'[\mathbf{x_t} - A\mathbf{x_{t-1}}]\right\}(2\pi)^{-d/2}.$$

Then the log-likelihood is just a sum of quadratic terms

$$\begin{aligned}\log P(\{\mathbf{x}\},\{\mathbf{y}\}) = &-\sum_{t=1}^{T}\left(\frac{1}{2}[\mathbf{y_t} - C\mathbf{x_t}]' R^{-1}[\mathbf{y_t} - C\mathbf{x_t}]\right) - \frac{T}{2}\log|R| \\ &-\sum_{t=1}^{T}\left(\frac{1}{2}[\mathbf{x_t} - A\mathbf{x_{t-1}}]'[\mathbf{x_t} - A\mathbf{x_{t-1}}]\right) - \frac{T}{2}\log|\mathbf{I}| + \log(\mathbb{1}_{\pi_0}(\mathbf{x}_0)).\end{aligned} \tag{3.3.1}$$

Replace $\log P(\{\mathbf{x}\},\{\mathbf{y}\})$ with equation 3.3, model 3.2 is

$$\begin{aligned}\widehat{\Theta} = \arg\min_{\Theta}\Big\{ &\sum_{t=1}^{T}\big(\frac{1}{2}[\mathbf{y_t}-C\mathbf{x_t}]'R^{-1}[\mathbf{y_t}-C\mathbf{x_t}]\big)-\frac{T}{2}\log|R| \\ &+\sum_{t=1}^{T}\big(\frac{1}{2}[\mathbf{x_t}-A\mathbf{x_{t-1}}]'[\mathbf{x_t}-A\mathbf{x_{t-1}}]\big)-\frac{T}{2}\log|\mathbf{I}|-\log(\mathbb{1}_{\pi_0}(\mathbf{x}_0)) \\ &+\lambda_1\|A\|_1+\lambda_2\|C\|_2^2\Big\}.\end{aligned} \tag{3.3.2}$$

Denote the target function in the parenthesis as $\mathbf{\Phi}(\Theta,\{\mathbf{y}\},\mathbf{x})$, then $\mathbf{\Phi}$ can be optimized with an Expectation-Maximization (EM) algorithm.

3.3.1 E Step

The E step of EM requires computing the expected log likelihood,

$$\Gamma = E[\log P(\{\mathbf{x}\},\{\mathbf{y}\}|\{\mathbf{y}\})].$$

This quantity depends on three expectations: $E[\mathbf{x}_t|\{\mathbf{y}\}]$, $E[\mathbf{x}_t\mathbf{x}_t'|\{\mathbf{y}\}]$ and $E[\mathbf{x}_t\mathbf{x}_{t-1}'|\{\mathbf{y}\}]$. We denote them by the symbols:

$$\begin{aligned}\widehat{\mathbf{x}}_t &\equiv E[\mathbf{x_t}|\{\mathbf{y}\}] \\ P_t &\equiv E[\mathbf{x}_t\mathbf{x}_t'|\{\mathbf{y}\}] \\ P_{t,t-1} &\equiv E[\mathbf{x}_t\mathbf{x}_{t-1}'|\{\mathbf{y}\}].\end{aligned} \tag{3.3.3}$$

Expectations 3.3.3 are calculated with a Kalman filter/smoother, which is detailed in Appendix to Chapter 3.

3.3.2 M Step

The parameters are $\Theta = \{A, C, R, \pi_0\}$. Each of them is re-estimated by taking the corresponding partial derivatives of $\mathbf{\Phi}(\Theta, \{\mathbf{y}\}, \mathbf{x})$, setting to zero and solving.

Denote estimations from previous step as $\Theta^{\text{old}} = \{A^{\text{old}}, C^{\text{old}}, R^{\text{old}}, \pi_0^{\text{old}}\}$ and current estimations as $\Theta^{\text{new}} = \{A^{\text{new}}, C^{\text{new}}, R^{\text{new}}, \pi_0^{\text{new}}\}$. Estimation for output noise covariance R has closed form solution,

$$\frac{\partial \mathbf{\Phi}}{\partial R^{-1}} = \frac{T}{2}R - \sum_{t=1}^{T}\big(\frac{1}{2}\mathbf{y}_t\mathbf{y}_t' - C\widehat{\mathbf{x}}_t\mathbf{y}_t' + \frac{1}{2}CP_tC'\big) = 0$$

$$R = \frac{1}{T}\sum_{t=1}^{T}(\mathbf{y}_t\mathbf{y}_t' - C^{\text{new}}\widehat{\mathbf{x}}_t\mathbf{y}_t')$$

$$R^{\text{new}} = \text{Diag}\left\{\frac{1}{T}\sum_{t=1}^{T}(\mathbf{y}_t\mathbf{y}_t' - C\widehat{\mathbf{x}}_t\mathbf{y}_t')\right\}$$

At the bottom line, diagonal of the estimated R is taken, as we constrain R to be diagonal in Constraint 4.

Estimation for initial state also has closed form. The relevant term $\log(\mathbb{1}_{\pi_0}(\widehat{\mathbf{x}}_0))$ is minimized only when

$$\pi_0^{\text{new}} = \widehat{\mathbf{x}}_0$$

Estimation for transition matrix C also has closed form solution, but the solution can

only be derived by rearranging the terms properly. Terms relevant to C in equation 3.3 are

$$\sum_{t=1}^{T}\left(\frac{1}{2}[\mathbf{y}_t - C\mathbf{x}_t]' R^{-1}[\mathbf{y}_t - C\mathbf{x}_t]\right) + \lambda_2\|C\|_2 = f_{\lambda_2}(C;\{\mathbf{x}\},\{\mathbf{y}\}). \tag{3.3.4}$$

In $f_{\lambda_1}(C;\{\mathbf{x}\},\{\mathbf{y}\})$, C is a matrix and need to be vectorized for optimization. Here we follow the methods of Turlach et al Turlach et al. (2005). Without loss of generality, assume R is the identity matrix in equation 3.3.4; otherwise, one can always write equation 3.3.4 as

$$\sum_{t=1}^{T}\left(\frac{1}{2}[R^{-\frac{1}{2}}y_t - R^{-\frac{1}{2}}Cx_t]'[R^{-\frac{1}{2}y_t} - R^{-\frac{1}{2}}Cx_t]\right) + \lambda_2\|R^{-\frac{1}{2}}C\|$$

Let

$$\mathbf{w} = (y_{11},\ldots,y_{T1},y_{12},\ldots,y_{T2},\ldots,y_{1p},\ldots,y_{Tp})'$$

be a $Tp\times 1$ vector from rearranging $\{\mathbf{y}\}$. In addition, let

$$\mathbf{W} = \begin{pmatrix} W' & & \\ & \ddots & \\ & & W' \end{pmatrix}_{pT\times pd}$$

where $W = \begin{pmatrix}\mathbf{x}_1,\ldots,\mathbf{x}_T\end{pmatrix}$. Finally, vectorize C^{old} as

$$\mathbf{c}^{\text{old}} = (C_{11}^{\text{old}},\ldots,C_{1d}^{\text{old}},C_{21}^{\text{old}},\ldots,C_{2d}^{\text{old}},C_{p1}^{\text{old}},\ldots,C_{pd}^{\text{old}})' \tag{3.3.5}$$

where C_{ij} is the element at row i and column j of C. With these new notations, the equation 3.3.4 is equivalent to

$$f_{\lambda_2}(C; \{\mathbf{x}\}, \{\mathbf{y}\}) = \|\mathbf{w} - \mathbf{W}\mathbf{c}\|_2^2 + \lambda_2 \|\mathbf{c}\|_2^2. \tag{3.3.6}$$

With the Tikhonov regularization Tikhonov (1943), equation 3.3.6 has closed form solution

$$\begin{aligned} \mathbf{c}^{\text{new}} &= (\mathbf{W}'\mathbf{W} + \lambda_2 \mathbf{I})^{-1}\mathbf{W}'\mathbf{w} \\ C^{\text{new}} &= \text{Rearrange } \mathbf{c}^{\text{new}} \text{ by equation } 3.3.5 \end{aligned} \tag{3.3.7}$$

Last but not least, let's look at parameter A. Terms involving A in equation 3.3 are,

$$\sum_{t=1}^{T} \big(\frac{1}{2}[\mathbf{x}_t - A\mathbf{x}_{t-1}]'[\mathbf{x}_t - A\mathbf{x}_{t-1}]\big) + \lambda_1 \|A\|_1 = f_{\lambda_1}(A; \{\mathbf{x}\}, \{\mathbf{y}\}). \tag{3.3.8}$$

Similar to what we have done to C, equation 3.3.8 is equivalent to

$$f_{\lambda_1}(A; \{\mathbf{x}\}, \{\mathbf{y}\}) = \|\mathbf{z} - \mathbf{Z}\mathbf{a}\|_2^2 + \lambda_2 \|\mathbf{a}\|_1. \tag{3.3.9}$$

where $\mathbf{z}$ is a $Td \times 1$ vector from rearranging $\{\mathbf{x}\}$ and $\mathbf{Z}$ is a block diagonal matrix with diagonal component $Z' = (\mathbf{x}_0, \ldots, \mathbf{x}_{T-1})'$. Unfortunately, equation 3.3.9 does not have closed form solution due to the $L - 1$ term.

Though not having a closed form solution, $f_{\lambda_1}(A; \{\mathbf{x}\}, \{\mathbf{y}\})$ can be solved numerically with a Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) Beck and Teboulle (2009)

Hoerl and Kennard (1970). FISTA is an accelerated version of the Iterative Shrinkage-Threshholding Algorithm (ISTA). ISTA is linearly convergent while FISTA is quadratic convergent. Steps of a general FISTA algorithm can be found in Appendix to Chapter 3.

FISTA requires calculating the Lipschitz constant L for $\nabla \mathbf{g}(\mathbf{z}) = \mathbf{Z}'(\mathbf{Z}\mathbf{a} - \mathbf{z})$, where $\mathbf{g}(\mathbf{z}) = \|\mathbf{Z}'\mathbf{a} - \mathbf{z}\|_2^2$. Denote $\|Z\|$ as the induced norm of matrix Z, then L is

$$L = \sup_{x \neq y} \frac{\|\mathbf{Z}'(\mathbf{Z}x - \mathbf{Z}y)\|}{\|x - y\|} = \sup_{x \neq 0} \frac{\|\mathbf{Z}'\mathbf{Z}x\|}{\|x\|} \leq \|\mathbf{Z}'\|\|\mathbf{Z}\| = \|Z'\|\|Z\|.$$

With FISTA and L, matrix A can be update:

$$A^{\text{new}} = \text{FISTA}(\|\mathbf{Z}'\mathbf{a}^{\text{old}} - \mathbf{z}\|_2^2, \quad \lambda_2) \tag{3.3.10}$$

3.3.3 The Complete EM

The complete EM algorithm for PLDS is addressed as follows.

Algorithm EM Algorithm for PLDS

M Step

1. $R^{\text{new}} = \text{Diag}\left\{ \frac{1}{T} \sum_{t=1}^{T} (\mathbf{y}_t \mathbf{y}_t' - C^{\text{old}} \widehat{\mathbf{x}}_t \mathbf{y}_t') \right\}$

2. $\pi_0^{\text{new}} = \widehat{\mathbf{x}}_0$

3. Update C^{new} as in equation 3.3.7

4. Update A^{new} with FISTA, as in equation 3.3.10

E Step

0. Initialize $\Theta = \{A, C, R, \pi_0\}$ if first loop

1. Update the expectations in 3.3.3 with the Kalman filter smoother in **??**

Notice that all the terms involving $\{\mathbf{x}\}$ in the M-step are approximated with the conditional expectations calculated in E-step.

Denote $\mathbf{Y} = [\mathbf{y_1}, \cdots, \mathbf{y_T}]$, a $p \times T$ matrix. The singular value decomposition (SVD) of $\mathbf{Y}$ is

$$\mathbf{Y} = \mathbf{U}\mathbf{D}\mathbf{V}' \approx \mathbf{U}_{p\times d}\mathbf{D}_{d\times d}\mathbf{V}'_{d\times T} = \mathbf{U}_{p\times d}\mathbf{X}_{d\times T} \tag{3.3.11}$$

where $\mathbf{U}_{p\times d}$ is the first d columns of $\mathbf{U}$ and $\mathbf{D}_{d\times d}$ is the upper left block of $\mathbf{D}$. This notation also applies to $\mathbf{V}'_{d\times T}$.

C is then initialzed as $\mathbf{U}_{p\times d}$, while the columns of $\mathbf{X}_{d\times T}$ are used as input for a vector autoregressive (VAR) model to estimate the initial value for A.

The major factors that affect the efficiency and scalability of the above EM algorithm involve the storage and computations of covariance matrix R. The following computational techniques are utilized to make the code highly efficient and scalable.

First, a sparse matrix is used to represent R. When dimension p gets higher, the size of R increase quadratically, which will easily exceed the memory capacity of a computer. Fortunately, with Constraint 4, R is sparse and can be represented with a sparse matrix. For example, when $p = 10,000$, the full R matrix takes over 100 Gigabyte memory, while the sparse matrix takes less than 1 Megabyte.

In addition, to update R in the M step, directly calculate its diagonal without calculating the full matrix R.

Finally, in the E-step, the following term K_t involving R need to be calculated,

$$K_t = V_t^{t-1}C'(CV_t^{t-1}C' + R)^{-1}$$

which involves the inverse of a large square matrix of dimension p by p. As stated previously, such a matrix exceeds available memory when p is high. The Woodbury Matrix Identity is employed to turn a high dimensional inverse to low dimensional problem:

$$(CV_t^{t-1}C' + R)^{-1} = R^{-1} - R^{-1}C[(V_t^{t-1})^{-1} + C'R^{-1}C]^{-1}C'R^{-1}$$

With the above three techniques, the EM algorithm can scale to very high dimensions in terms of p, d and T, without causing any computational issues.

3.4 Result

3.4.1 Parameter Estimations

Two simulations of different dimensions are performed to demonstrate the model and its parameter estimations.

In the low dimensional setting, $p = 300$, $d = 10$ and $T = 100$. The A matrix is generated such that the conditional number is no less than some threshold, 50 being used. Elements with small absolute values are then truncated such that 20 percent of elements are zeros. Eigen values of A are controlled within $[-1, 1]$ to avoid diverging time series. Matrix C is generated as follows. Each column contains random samples from a standard Gaussian distribution. Then the sample is sorted in ascending order. Covariance Q is the identity matrix and covariance R is a multiple of the identity matrix. At time 0, a zero vector $\mathbf{0}$ is used as the value of $\mathbf{x_0}$.

In the high-dimensional setting, p = 10,000, d = 30 and T = 100. The parameter are generated in a similar manner.

To evaluate the accuracy of estimations, some distance measure should be first defined. Here the distance between two matrices A and B is defined as follows

$$d(A, B) = \max_{P \in P(n)} \mathrm{Tr}\big(P \times C_{A,B}\big)$$

where $C_{A,B}$ is the correlation matrix between columns of A and B, $P(n)$ is the collection

of all the permutation matrices of order n and P is a permutation matrix.

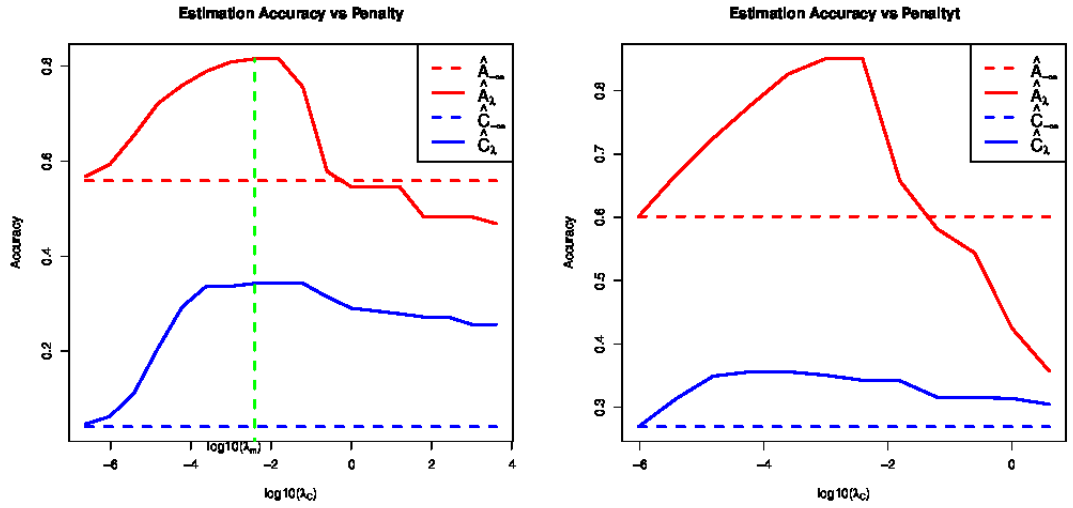
The calculation of $d(A, B)$ is essentially a linear assignment problem and can be solved in polynomial time with the Hungarian algorithm Kuhn (1955).

Both the generic LDS and the penalized LDS are applied to the simulation data. As the true parameters are sparse, we expect that the penalized algorithms would yield better estimations with some proper penalty parameters. When the penalties are approaching 0, the penalized algorithm should converge to the generic model. In addition, when the penalties are getting larger, the penalized algorithm's estimations should become worse.

A sequence of tuning parameters λ_C are utilized, ranging from 10^{-6} to 10^4. $\lambda_A = k\lambda_C$ is set to increase proportionally with λ_C, where k is a constant.

Estimation accuracies are plotted against penalty size λ_C in Figure 3.1. Results from LDS and PLDS are overlayed in one plot for comparison. As the f gure shows, PLDS converges to the LDS when the penalties are approaching zero. Estimation accuracies f rst increase with penalty size and then decrease due to over-shrinkage.

As a concrete example, estimations from both methods are compared to the true values of parameters in Figure 3.2. One can see that true values in each column of C matrix are decreasing smoothly. $\widehat{C}_{\lambda_m}$, which is estimated with optimal penalties $\lambda_C = \lambda_m$ and $\lambda_A = k\lambda_m$, shows similar pattern. In terms of A, the true value is sparse with many 0 (blue) values. PLDS estimation $\widehat{A}_{\lambda_m}$ is also sparse, denoted by the off-diagonal blue values. However, LDS estimation $\widehat{A}_{\lambda_{-\infty}}$ is not sparse, with many yellow and red off-diagonal values.



(a) Low dimensional setting (b) High dimensional setting

Figure 3.1: x axis is tuning parameter λ_C under log scale and y axis is the distance between truth and estimations; λ_A is increasing proportionally with λ_C

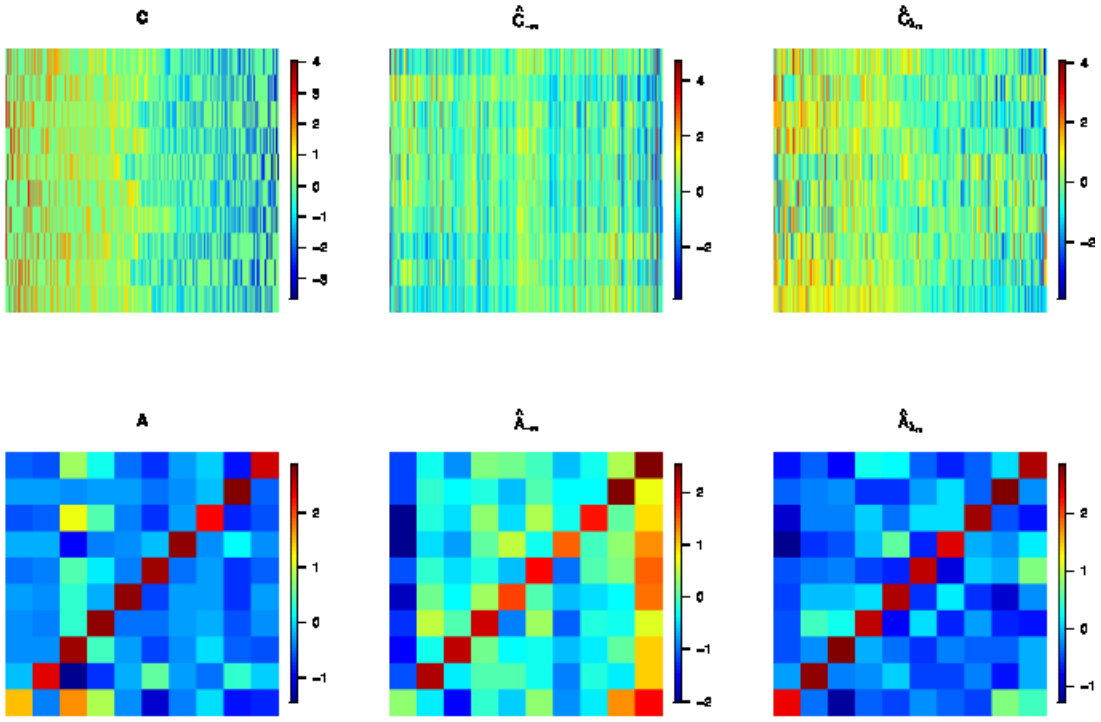


Figure 3.2: Row 1: A truth; non-penalized estimation of A; optimally penalized estimation of A. Row 2: C truth; non-penalized estimation of C; optimally penalized estimation of C.

In addition to the improved estimation accuracy, the proposed algorithm is also computational eff ciency and highly scalable. As a demonstrate, we measure the running times of multiple simulation scenarios and summarize them in Table 3.1. When both p and d are high dimensional, the algorithm can still solve the problem in a reasonable time.

Table 3.1: PLDS Running Time

p	100	1000	10000	100000	100000
d	10	30	50	100	500
T	100	300	500	1000	1000
Time (min)	0.04	0.50	51.28	208.82	1801.00

3.4.2 Making Predictions

Another perspective when considering the PLDS model is its ability to make predictions. When the parameters Θ and the latent states x_T are estimated, one can f rst use estimated x_T to predict x_{T+1} and use x_{T+1} to predict y_{T+1}. Similarly, more predictions $y_{T+2}, \ldots, y_{T+k}$ can be made. Intuitively, properly chosen penalties give better estimations and good estimations should give more accurate predictions. This idea is demonstrated with a simulation. The parameter settings for this simulation follow Section 3.4.1. The correlation between the predicted signal and true signal is used as a measure of prediction accuracy. The prediction accuracy over penalty size is shown in Figure 3.3.

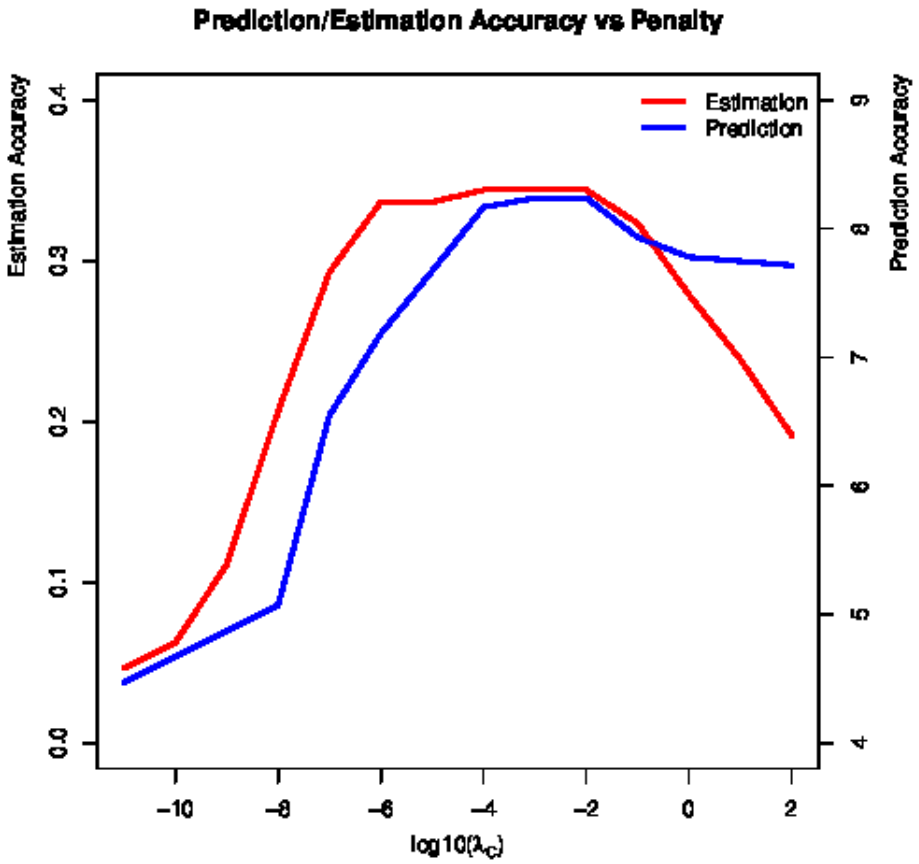


Figure 3.3: Estimation and prediction accuracies.

Observations and findings from these plots include:

- The prediction accuracy first improves then drops when the penalties increase
- The prediction accuracy peaks when the penalty coefficient λ_A and λ_C are around 10^{-3}. This make sense as the same λ pair also gives the best estimation for coefficients A and C, as in Figure 3.1.

This second observation provides us a way to pick tuning parameters in real applications, as detailed in Section 3.5.

3.5 Application

When applied to fMRI data analysis, the model has very good interpretability. Each $\mathbf{y}_t$ is a scan of the brain. Each column of the C matrix is interpreted as a time-invariant brain network. At each time point, the observed brain image, $\mathbf{y}_t$, is a linear mixture of these networks and $\mathbf{x}_t$ contains the mixing coeff cients. Matrix A describes how $\mathbf{x}_t$ evolves over time. A can also be viewed as a directed graph if each network is treated as a vertex. Brain networks are spatially smooth and connectivities among them are empirically sparse. This naturally f ts into the sparsity and smoothness assumptions in PLDS.

The PLDS is applied to analyze the motor cortex of human brains from the KIRBY 21 Data. These data are resting-state fMRI scans consisting of a test-retest dataset previously acquired at the FM Kirby Research Center at the Kennedy Krieger Institute, Johns Hopkins University Landman et al. (2011). Twenty-one healthy volunteers with no history of neurological disease each underwent two separate resting state fMRI sessions on the same scanner: a 3T MR scanner utilizing a body coil with a 2D echoplanar (EPI) sequence and eight channel phased array SENSitivity Encoding (SENSE; factor of 2) with the following parameters: TR 2s; 3mm$\times$3mm in plane resolution; slice gap 1mm; and total imaging time of 7 minutes and 14 seconds.

In this application, test-retest scans from two subjects are analyzed. The imaging data are f rst preprocessed with FSL, a comprehensive library of analysis tools for fMRI, MRI and DTI brain imaging data Smith et al. (2004). FSL is used for spatial smoothing with Gaussian kernel. Then PLDS is applied on the smoothed data.

The following are basic descriptions of the data and model parameters.

- Number of voxels, p = 7396
- Number of scans, T = 210
- Number of latent states, d = 11
- Tuning parameters: $\lambda_A = 0.00001$, $\lambda_C = 0.00001$
- Max number of iterations: EM 30 steps, L-1/L-2 regularized subproblems, 30 steps

The number of latent states, d, can be manually selected based on related research that maps the primary motor region to human activities. For instance, Meier et. al mapped the motor region to 9 human organs: tongue, lips, squint, fingers, wrist, forearm, elbow, foot and saccade Meier et al. (2008).

A more flexible technique to choose the number of latent states involves the profile likelihood method proposed by Zhu et. al Zhu and Ghodsi (2006). As a first step, eigen values of the data matrix are calculated with Principal Component Analysis (PCA). The cumulative eigen values as a percentage of the sum of all eigen values are then plotted - see Figure 3.4. Visually one notes that the first 10 eigen values take over 80% of all variations. The number of latent states can be selected as the smallest number (of eigen values) that explains over 80% of total variation in the data. However, the drawback of this method is clear: the choice of threshold percentage (here 80%) is highly subjective. The profile likelihood method overcomes this problem and could pick the dimension automatically.

The above method assumes that the first k eigen-values are samples from a Gaussian distribution $N(\mu_1, \sigma^2)$, while the rest are from a different Gaussian distribution $N(\mu_2, \sigma^2)$. Then the profile likelihood can be calculated given k, for all $k = 1, \cdots, T$ and selecting the optimal k as the one with the highest profile likelihood. As shown in Figure 3.4, when the profile likilihood method is applied to the first scan of subject one, d = 11 is selected. Apply the method to all four scans, the numbers of latents states selected are 6, 11, 14 and 15 respectively. Their average, $d = 11$, is used.

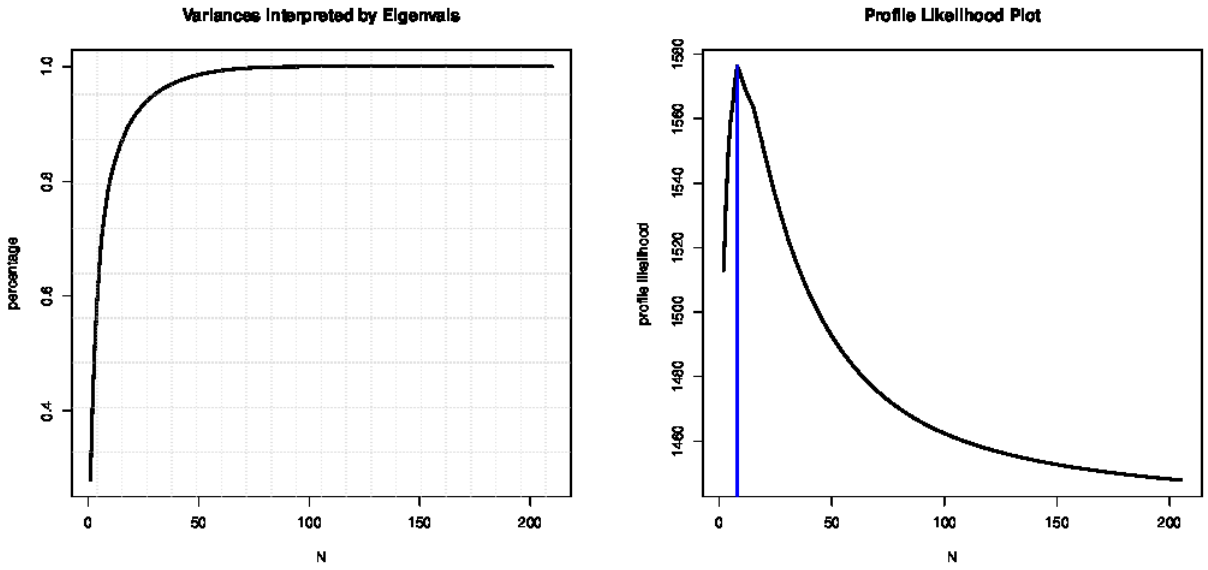


Figure 3.4: Eigen-values and Corresponding Profile Likelihood Plot

The A matrices as connectivity graphs are first plotted in Figure 3.5. One can group the scans correctly with the A matrices. Specifically, denote the A matrix estimation for the first scan of subject one as A_{11}. Similary notations apply to the other scans. Then

the canonical correlations among the four matrices are summarized in Table 3.2. Another permutation invariant measure of square matrix similarity, the Amari error, is also provided in the table Amari et al. (1996). Notice a higher $d(A, B)$ or a smaller Amari error means more similarity. From both measures, one can group the four scans correctly. This implies that the graph contains subject-specif c information.

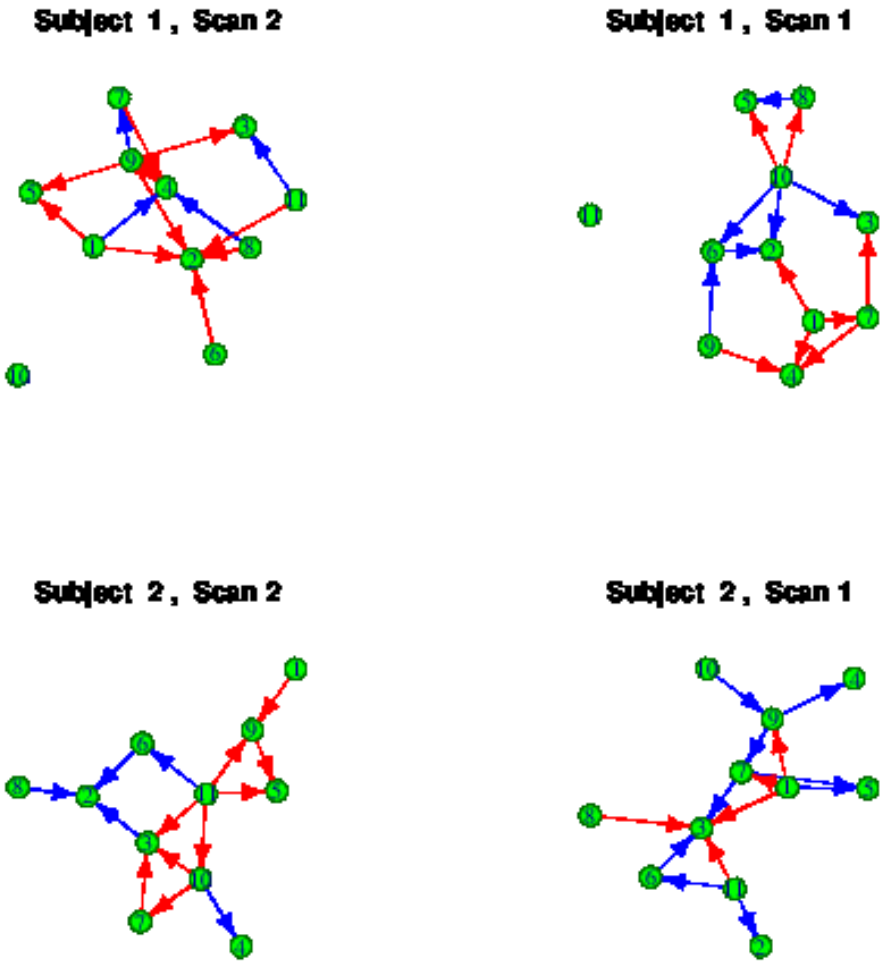


Figure 3.5: Connectivity Graph: The wider edge means stronger connectivity; the red edge means negative connectivity and blue edge means positive connectivity.

Table 3.2: Similarities Among Estimated A Matrices

Pairs	A_{11}, A_{12}	A_{11}, A_{21}	A_{11}, A_{22}	A_{12}, A_{21}	A_{12}, A_{22}	A_{21}, A_{22}
$d(\cdot, \cdot)$	**10.2**	9.9	10.0	10.0	10.0	**10.1**
Amari Error	**0.88**	1.05	1.02	1.08	1.09	**0.98**

As an example, the 3D renderings of the columns of matrix C from the first scan of subject one are shown in Figure 3.6 (after thresholding). The biological meaning of those regions need to be further validated. It is helpful to compare those regions to other existing parcellations of the mortor cortex. As an example, the blue region above accurately matches the DM (dorselmedical) parcel of the five-region parcellation proposed by Nebel MB et al. Nebel et al. (2014b).

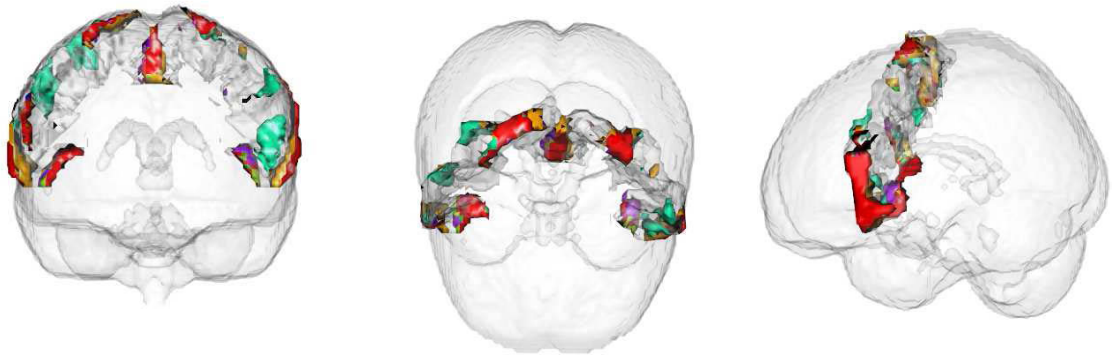


Figure 3.6: 3D Rendering of Columns of Matrix C

Another application of the algorithm is predicting brain signals. To demonstrate this, the algorithm is applied to the Human Connectome Project (HCP) data. Using the profile likelihood method, $d = 149$ is picked. The data has $T = 1200$ time points. The first $N = 1000$ are picked as train data, while the rest are used as test data. Then both the

SVD method in Equation and the PLDS algorithm are used for prediction. The prediction accuracies are shown in Figure 3.7. The first observation is that, the PLDS algorithm is giving significantly better predictions for the first 150 predictions compared to the SVD method. As the SVD method is also used to intialize the PLDS algorithm, this shows that the PLDS algorithm improves estimations from the SVD method in terms of short-term predictions. Another observation is, the PLDS algorithm's performance get worse when one predicts into the "long" future (> 150 steps). This is reasonable, as there is no way that we can predict the two noise terms in the model, therefore the prediction errors from each step will accumulate and yields deteriorating predictions.

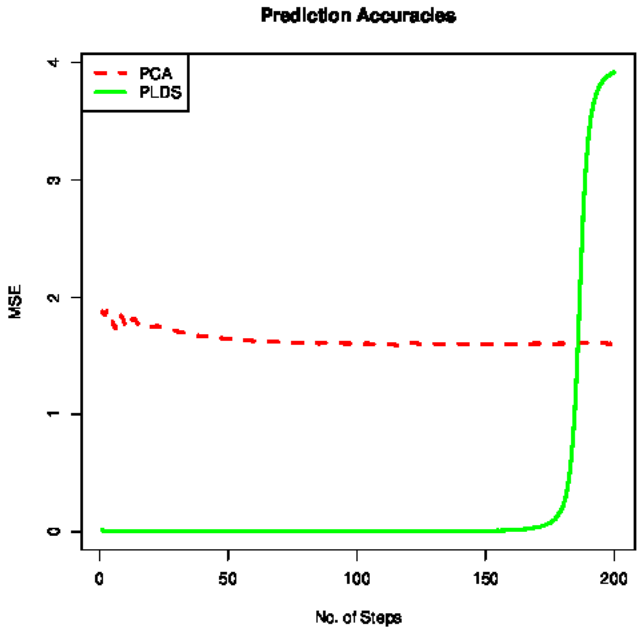


Figure 3.7: Prediction accuracies comparison on HCP data

3.6 Discussion

By applying the proposed model to fMRI scans of the motor cortex of healthy adults, we identify limited sub-regions (networks) from the motor cortex. A statistical procedure should be further developed to match these regions to existing parcellations of the motor cortex.

In the future, this work could be extended in two important directions. First, assumptions on the covariance structures in the observation equation could be generalized. Prior knowledge could be incorporated to covariance R. The general rule is that R should be general enough to be f exible while suff ciently restricted to make the model useful. A lot of other platforms such as tridiagnol and upper triangular could also be considered. Mohammad et. al have discussed the impact of auto correlation on functional connectivity, which also provides us a direction for extension Arbabshirani et al. (2014).

Finally, the work can also be extended on the application side. Currently, only data from a single subject is analyzed. As a next step, the model can be extended to a group version and be used to analyze more subjects. The coeff cients from the algorithm could be used to measure the reproducibility of the scans.

Chapter 4

fMRI Based Biomarker for Physical Pain

Abstract

Physical pain is most often measured by self-report. This sole reliance on individual recollection hampers diagnosis and treatment. In addition, self report is not calibrated across individuals. Accurate biomarkers of pain would revolutionize pain management. However, the development of accurate biomarkers remains a difficult and unsolved problem in this area. Functional magnetic resonance imaging (fMRI) is widely used in neuroscience for quantifying brain function in vivo. Its excellent spatial resolution and ability to measure time varying cognitive function via BOLD signals make it a promising tool as a biomarker for pain. In this article, a functional regression model is built to extract features from high-dimensional time series and then a support vector machine (SVM) is trained with the features for physical pain prediction with fMRI data. It was found that with fMRI signal, one can predict the physical pain with a accuracy of over 78%.

keywords: physical pain, biomarker, fMRI, machine learning, functional regression

4.1 Introduction

Functional magnetic resonance imaging (fMRI) is a neuroimaging procedure using MRI technology that measures brain activity by detecting associated changes in blood f ow. This technique relies on the fact that celebral blood f ow and neuronal activation are coupled: when an area of the brain is in use, blood f ow to that region also increases.

Functional MRI has been widely used in medical research. Notably, people have explored the potential of fMRI as a biomarker for perception and thinking. For example, Nishimoto et al. (2011) used fMRI to predict what a subject has seen when watching a movie.

Physical pain is an aff iction associated with enormous cognitive, social and economic costs (Simon, 2012). It is mainly assessed via self-report, an imperfect, uncalibrated and subjective measure. In addition, the capacity to reliably report pain is limited in some populations, such as the elderly, young children and people with cognitive impairments. Moreover, self-report provides little insight into the neurophysiological processes underlying pain.

It is promising to derive a fMRI-based biomarker for pain assessment combining fMRI technology and pain measurement. In fact, Wager et al. (2013) have used fMRI as a biomarker to predict the scale of pain that a subject feels from heat stimuli. The research in this manuscript builds upon their work and aims to improve prediction accuracy by exploring better features and classif ers.

In this study, thermal stimuli of four different intensities are given to the left forearm of

each participant in randomized sequences (trials). Each trial is followed by a fMRI scan of the subject's brain. Each participant underwent 12 trials at each of four intensities. The four intensities consists of an innocuous warmth and three levels of painful heat. The innocuous warmth is def ned with the use of self-report by participants as level 1 on a 9-point visual-analogue scale (VAS), with temperature of $41.0 \pm 1.9°\mathrm{C}$. Similarly, the three levels of painful heat correspond to participant-def ned levels 3, 5 and 7, with mean temperatures of $43.3 \pm 2.1°\mathrm{C}, 45.4 \pm 1.71°\mathrm{C}$ and $47.1 \pm 0.98°\mathrm{C}$ respectively.

In each fMRI scan, the brain is spatially separated into small voxels (a three-dimensional rectangular cuboid; 3d version of a pixel). A subject is scanned multiple times in each scan session, generating a $T \times V$ matrix, where T is the number of time points and V is the number of voxels. Denote the number of subjects as I. The number of stimuli given to subject i is N_i. Each stimulus is followed by a scan session, which contains T times of scan. Let Y_{in} be the pain scale reported by subject i after the n-th stimulus. The variable C_{in} is the corresponding pain category, and $C_{in} = 1, 2, 3, 4$ corresponds to the innocuous warmth and 3 painful heat stimulus respectively. The variable $\mathbf{X}_{in}$ is a $T \times V$ matrix, denoting the scan session following stimulus Y_{in}.

The goal is to predict Y_{in} from $\mathbf{X}_{in}$. The biomarker development procedure consists of two steps:

Feature Extraction: determine which voxels to include; determine which time courses to include and how to combine them.

Statistical Learning: train a classif er with the selected feature for pain prediction.

4.2 Feature Extraction

Feature extraction consists of two steps. As a first step, domain-specific information is used to pick out the most relevant voxels. After that, a time-weighted average of the vector time series is used as the feature.

To elaborate on the first step, voxels within pain-related regions were selected based on a meta-analysis of 224 previous studies. They were selected with the automated meta-analysis toolbox Neurosynth (www.neurosynth.org) based on previous studies that frequently use the word "pain" (Yarkoni et al., 2011; Wager et al., 2013). The voxels are based on regions showing consistent results across 224 published studies out of $4,393$ studies in the database. As a result, $22,379$ positive voxels whose activity positively predicted pain and $10,940$ negative voxels that negatively predicted pain are selected (2 x 2 x 2 mm, resliced to 3 x 3 x 3 mm for analysis). These voxels account for 9.45% of the total in-brain volume.

In the second step, the vector time series are averaged over time. Unlike common problems in machine learning, in this problem, each observation (pain Y_{in}) corresponds to a vector time series, instead of a feature vector. For instance, the covariates correponding

to Y_{11} is in the following format:

$$\mathbf{X_{11}} = \begin{pmatrix} x_{11,11} & \cdots & x_{11,V1} \\ x_{11,12} & \cdots & x_{11,V2} \\ \vdots & \ddots & \vdots \\ x_{11,1T} & \cdots & x_{11,VT} \end{pmatrix}.$$

One could vectorize the full time series matrix and use them as a feature. However, this approach has at least two drawbacks. First this feature fails to take advantage of the internal structure of the time series. In addition it is computationally ineff cient, even intractable. A weighted average of the time series (over time) might be a good alternative, as long as the weights could capture the most relevant time points.

Wager et al. (2013) uses a straight-forward box weight for averaging over time. An alternative is to use Gaussian kernel. These two approaches are driven by some domain-specif c information about the experiment, as explained below. In this research we designed a data driven, functional data analysis (FDA) model to estimate the weights. The three different weights are plotted in the Figure 4.1.

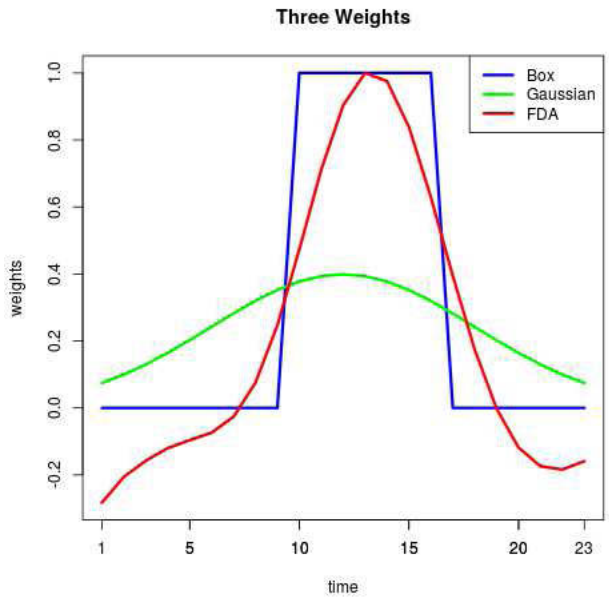


Figure 4.1: Three weights for multivariate time series averaging

The idea for the box weight is very simple: use a equally weighted average of time points 10 to 16 as the feature. The time points 10 to 16 were picked out based on prior expert knowledge about the experiments. In fact, the 23 time points lie into f ve stages: cue, pain, ISI 1, rating and ISI 2, where stage 3 (ISI 1) is the major stage that the pain stimulus takes effect and leads to brain activity. The box weight is fast, but lacks f exibility by using equal weights.

Gaussian weights are more f exible, as they can be tuned by changing the mean and variance of the underlying Gaussian kernel. Yet they are still not ideal, since the same weights are used across all voxels. In addition, its power is also limited by the symmetry property of Gaussian distribution.

The last approach is driven by the data. It is assumed that each voxel has its own weight

and the weights are learned from the data with a Functional Data Analysis (FDA) model (Goldsmith et al., 2010):

$$y_{ij} = \alpha_0 + \int_0^1 \beta_v(t) x_{ijv}(t) dt + \epsilon_{ij}. \tag{4.2.1}$$

Here, the dependent variable, y_{ij} is expressed as a integral over time. The continuous $x_{ijv}(t)$ function is estimated by smoothing the time series $x_{ij,vt}$, $t = 1, \cdots, T$, while $\{\beta_v(t)\}_{v=1}^V$ is the smooth weight function. Then $\{\beta_v(t)\}_{v=1}^V$ are used to average the time series as the feature.

In this functional regression model, one has a f nite number of observations with which to determine the inf nite dimensional $\beta_v(t)$. This is not identif ed, as it is almost always possile to f nd a $\beta_v(t)$ such that each $\epsilon_{ij} = 0$. More importantly, there are always an inf nite number of solutions for $\beta_v(t)$ that produces the same predictions (Graves et al., 2009).

One strategy to deal with identif ability is to use a basis expansion of $\beta_v(t)$:

$$\beta_v(t) = \sum_{k=1}^{K} c_k \phi_k(t), \tag{4.2.2}$$

where $\phi_k(t), k = 1, \cdots, K$ is a set of basis functions. In this application, the popular B-splines basis system is used (Catmull and Clark, 1978). A B-spline system is determined by its order, interior knots and two end points. Aside from the two end basis functions, each basis function begins at zero and rises to a peak at a certain knot location before falling back to zero and remaining there until the right boundary. The f rst and last functions rise from

the first and last interior knot to the value of one on the right and left boundary respectively.

In this work, the order and knots of B-splines are determined manually. Two sets of B-splines basis are used, one for data smoothing and one for the β coefficients. The number of basis should be big enough to catch the structure of the data, but not too big to cause overfitting. The model is implemented with R package *fda* (Febrero-Bande and Oviedo de la Fuente, 2012). The package has good support for functional data smoothing and functional regression. The prediction is implemented with the *e1071* package (Suykens and Vandewalle, 1999; Dimitriadou et al., 2008). Details are given in Section 4. A visual plot of $\{\beta_v\}_{v=1}^{V}$ is shown in Figure 4.2.

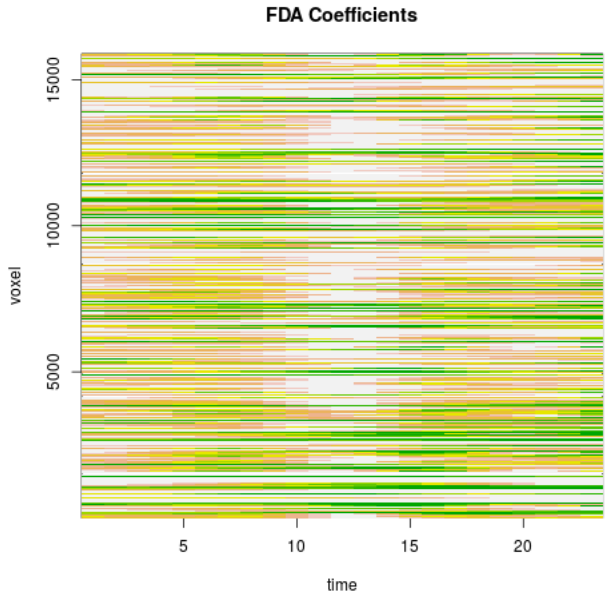


Figure 4.2: Weights estimated with functional regression: white color is more weight while green color is less weight

This plot validated the previous two approaches for feature averaging. In the heat map, whiter regions indicates higher weight. One could notice that the time points 10 to 16 has

higher weights almost across nearly all voxels.

4.3 Support Vector Regression

With the feature extracted above, we can proceed to find a good classifier for pain prediction. In this research the ϵ-Support Vector Regression (SVR) is adopted (Drucker et al., 1997; Gunn et al., 1998; Smola and Schölkopf, 2004).

The ϵ-Support Vector Regression (SVR) (Vapnik et al., 1997), is a Support Vector Classifier (SVC), which tries to maximize the margin between two classes. It seeks to find a function $f(x)$ that has at most ϵ deviation from the actually observed outcome, y_i, for all the training data, and at the same time is as "flat" as possible. In this section, we will introduce this powerful regression algorithm and explain how it was applied in our prediction problem.

4.3.1 Linear Regression

In the case that $f(x)$ is a linear function, suppose x is a p-dimensional vector representing the predictor, we have:

$$f(x) = \omega^T x + b, \text{ where } \omega \in \mathbb{R}^p, x \in \mathbb{R}^p, \text{ and } b \in \mathbb{R}^1.$$

The "flatness" is this case, means small values of ω, i.e. we want to minimize the l_2 norm: $||\omega||^2 = \omega^T\omega$. Ideally, the regression problem transforms to a convex optimization problem:

$$\begin{aligned} &\text{minimize } ||\omega||^2 \\ &\text{subject to } |y_i - \omega^T x_i - b| \leq \epsilon \end{aligned}$$

The above constraint indicates that a function $f(x)$ approximates the training data with at most ϵ absolute error. However, such function may not actually exist. In order to cope with such a situation, slack variables ξ and ξ^* can be introduced to the constraints. Then we have:

$$\begin{aligned} &\text{minimize } \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ &\text{subject to } \forall i, \begin{cases} y_i - \omega^T x_i - b \leq \epsilon + \xi_i \\ \omega^T x_i + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} . \end{aligned}$$

The constant $C > 0$ is called the cost parameter, which determines how much we want to penalize on the sum of outlying errors. Intuitively speaking, if C is small, then the function will tend to be flat, since our tolerance for ξ_i and ξ_i^* are large. As C gets larger, the $\|\omega\|^2$ will get larger, but we will have smaller training error. Thus we need to decide how to trade

off between model complexity and goodness-of-fit.

Note that, in this formulation, the optimization problem corresponds to dealing with a so called ϵ-insensitive loss function $|\xi|_\epsilon$ defined by:

$$|\xi|_\epsilon := \begin{cases} |\xi| - \epsilon & \text{if } |\xi| > \epsilon \\ 0 & \text{if } |\xi| \leq \epsilon. \end{cases}$$

In practice, this loss function can be defined in other forms, such as using squared loss instead of absolute loss.

4.3.2 Non-linear Regression and the Kernel Trick

The power of SVR comes with its capability to deal with the non-linear regression problem. Suppose we have an arbitrary basis function, Φ, that maps x from $\mathbb{R}^p$ to $\mathbb{R}^q$, where in most cases $q > p$. For example, $x = (a, b), \Phi(a, b) = (a^2, b^2, \sqrt{ab})$. Then by replacing x_i with $\Phi_i = \Phi(x_i)$, we are able to formulate some complex non-linear regression problem

as below:

$$\text{minimize } \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

$$\text{subject to } \forall i, \begin{cases} y_i - \omega^T\Phi_i - b \leq \epsilon + \xi_i \\ \omega^T\Phi_i + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} .$$

This type of optimization problem is often easier to solve in its dual form. Thus by applying the Lagrange Multiplier, $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$ (all non-negative), the primal Lagrangian is:

$$\begin{aligned} L :=& \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) - \sum_{i=1}^{n}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ &+ \sum_{i=1}^{n}\alpha_i(\omega^T\Phi_i + b - y_i - \epsilon - \xi_i) \\ &+ \sum_{i=1}^{n}\alpha_i^*(y_i - \omega^T\Phi_i - b - \epsilon - \xi_i^*). \end{aligned}$$

By taking the f rst derivative of L wrt to the primal variables, $\omega, b, \xi_i, \xi_i^*$, and setting them to zero, we will have the KKT conditions. We then plug them back to the Lagrangian, which

yields the dual optimization problem:

$$\begin{aligned}\text{maximize } & -\frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\Phi_i^T\Phi_j- \\ & \epsilon\sum_{i=1}^{n}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{n}y_i(\alpha_i-\alpha_i^*) \\ \text{subject to } & \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)=0 \text{ and } \alpha_i,\alpha_i^*\in[0,C].\end{aligned}$$

Since one the KKT conditions is that $\omega = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\Phi_i$. Thus by solving the dual problem, ω can be completely described as a linear combination of Φ_i. In a sense, the complexity of a functions representation by the support vectors (SVs) is independent of the dimensionality of the input space $\mathbf{X}$, and depends only on the number of SVs. This property gives us the power to estimate complex non-linear relation between x and y.

Moreover, note that to solve the dual problem, we need is the dot product form: $\Phi_i^T\Phi_j$. By pluging in $\omega = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\Phi_i$ to $f(x) = \omega^T\Phi(x) + b$, the representation of $f(x)$ also only depends on the the dot product form: $\Phi(x_i)^T\Phi(x)$. Hence it is suff cient to know $K(x, x') := \Phi(x)^T\Phi(x')$ rather than Φ explicitly. In theory, Mercer's Theorem rigorously characterizes the kernel function. In practice, there are 4 types kernel functions that are most commonly used:

1. linear kernel: $K(x_i, x_j) = x_i^T x_j$
2. polynomial kernel: $K(x_i, x_j) = (\gamma x_i^T x_j + c_0)^d, \gamma > 0$
3. radial basis function(RBF): $K(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2), \gamma > 0$

4. sigmoid kernel: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + c_0)$

4.4 Application to Pain Prediction

The SVM method is applied to predict pain intensity. The analysis consists of three parts.

In part one, SVM models are trained for pain prediction, which show that the functional regression based features improve prediction accuracy compared to box weights. In section two, by further averaging the features within a pain category for a subject, prediction accuracies for both functional regression weights and box weights are further improved. The former still gives higher prediction accuracy. In part three, we explored why the functional regression weights give better prediction accuracy by clustering voxels in the pain related regions are clustered according to their weights. The three parts are detailed as below.

4.4.1 Prediction Acurracy

In this analysis, the dataset is first divided into train and test data by subject. To be specific, 12 (60%) of the 20 subjects are used for training, while the remaining 8 are testing subjects. Then the weight matrix $W_{V\times T}$ is estimated with model (4.2.1) using the training data only. Apply $W_{V\times T}$ to both train and test data, we get corresponding features. Finally a SVM model is trained from training data and used for prediction on test data. The correlation between the predicted pain intensities and the real intensities is calculated as a measure

of prediction accuracy.

A leave-1-subject-out cross validation procedure was applied to search for the optimal prediction parameter setting. Leave-1-subject-out means that we did a 12-fold cross-validation, but instead of randomly permuting the data set, at iteration i, the 4 rows from subject i were picked out as the testing set and the rest were used as the training set.

In the functional regression, a B-spline basis system of order 12 is used for data smoothing, while another B-spline basis system of order 3 with breaks at $1, 10, 13, 16$ and 23 is used for the regression coeff cient. A grid search is performed, where the data smoothing basis order ranges from 4 to 20 and the beta coeff cient basis order ranging from 1 to 6. The orders combination with the best prediction accuracy is picked.

We did a grid search for all the four kernels introduced above, while the RBF had the best performance. For the RBF kernel, the searching grid involves three tuning parameters: (ϵ, C, γ). In the grid, $\epsilon \in [0.005, 1]$, $C \in [1, 300]$ and $\gamma \in (10^{-8}, 1]$, and by running the leave-1-subject-out cross-validation on each combination of parameters in the grid, we obtained the set of optimal parameters in terms of MSE and correlation$(y_i, \widehat{y_i})$.

The analysis is repeated 20 times using different training and test data. The same procedure is performed for box weight.

A comparison of the prediction accuracies with functional regression weights and box weights is shown in Figure 4.4. Performing a t-test to test the prediction accuracy difference, we get a mean difference of 0.041 with a p-value of 0.0002.

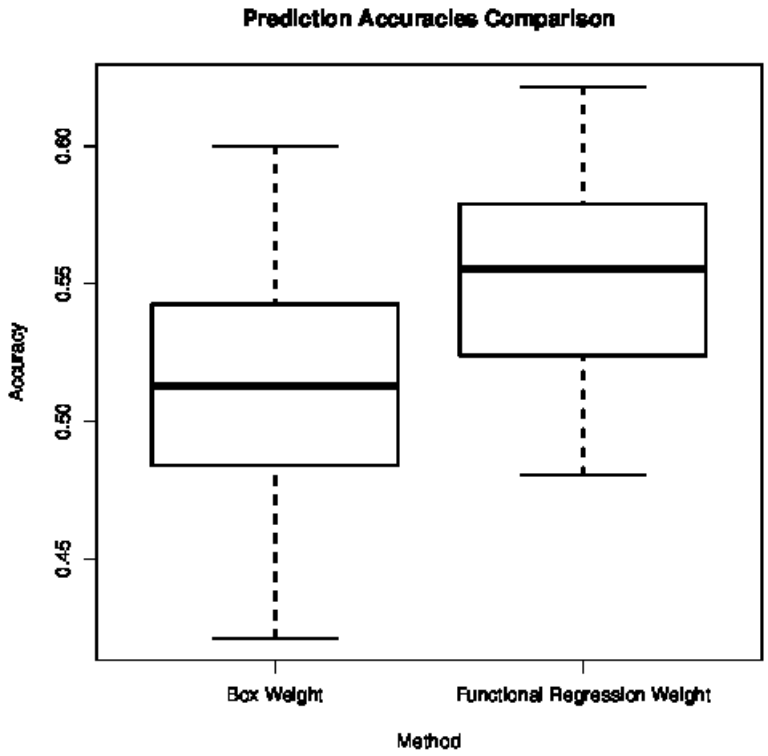


Figure 4.3: Prediction accuracies comparison. The mean correlations for functional regression weights and box weights are 0.554 and 0.513 respectively.

Another set of analysis is performed to further show the validity of the result. We changed the proportion of training data set to be 70%. The comparison is shown in Figure 4.4. Similarly, we get a mean difference of 0.04 with a p-value of 0.0009.

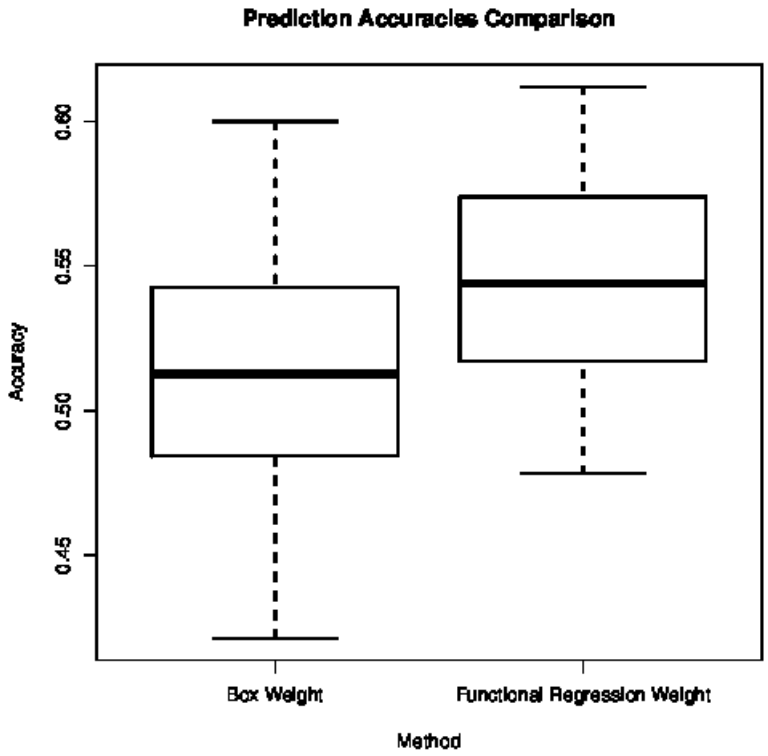


Figure 4.4: Prediction accuracies comparison. The mean correlations for functional regression weights and box weights are 0.55 and 0.51 respectively.

These shows that the functional regression based method has better prediction power than the simple box-weight method.

4.4.2 Further Improving Prediction Accuracy

This analysis differs from the above one in that the features are further averaged over pain categories for each subject. Essentially in this analysis, average intensities of four pain categories for each subject are averaged and predicted. In the above analysis, pain intensities for each stimulus given to a subject is used predicted. As expected, higher accuracy is

achieved in predicting the average intensities and the functional regression weights again yields higher accuracy. Similar to above, a random sample of 60% of subjects are used for training and the rest are used for testing. The analysis is repeated 30 times.

A comparison of the prediction accuracies in predict stimulus-level intensities and average intensities is shown in 4.5. The prediction accuracies has a mean difference of 0.03 and a p-value of 0.00004. Notice that though the accuracies here are very high, the analysis has its limitations, as it could only predict average pain intensities in a pain category for a subject.

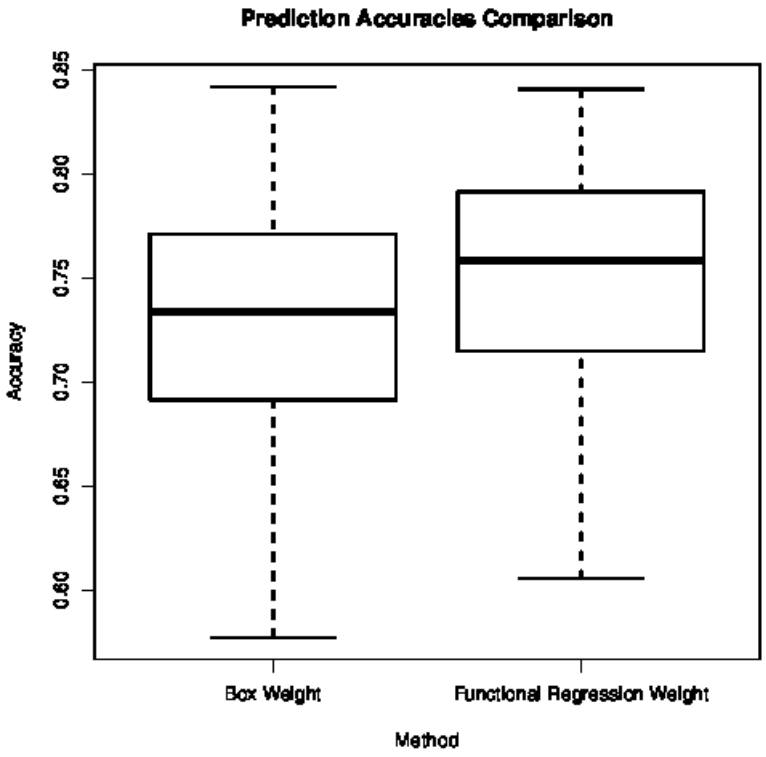


Figure 4.5: Prediction accuracies comparison. The mean correlations for functional regression weights and box weights are 0.75 and 0.72 respectively

4.4.3 Clustering Voxels

As shown above, the functional regression based weights give better prediction accuracies in different analysis scenarios. A naturally question to ask is: whether the weights are biologically meaningful? To explore this question, we clustered the voxels according to their estimates weights over time. The K-Means algorithm is adopted for clustering.

The number of clusters 5 is selected from the following sum of squared error (SSE) plot.

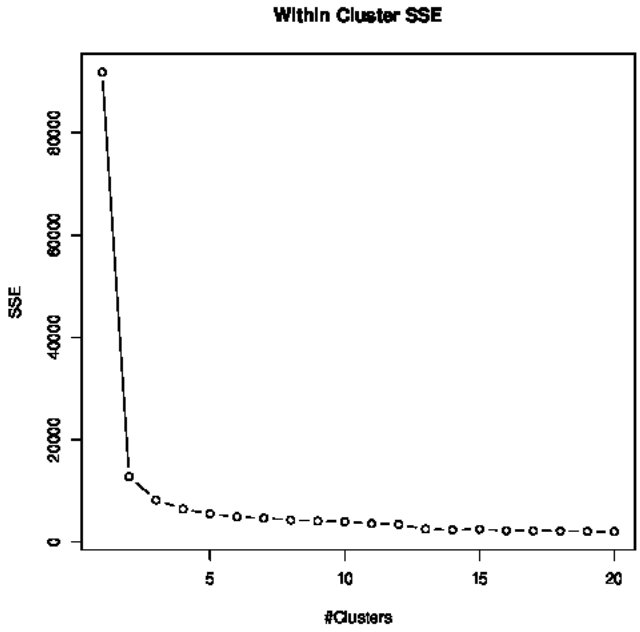
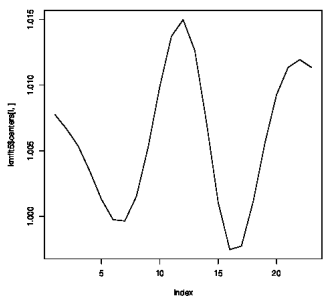
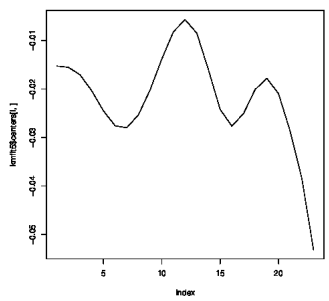
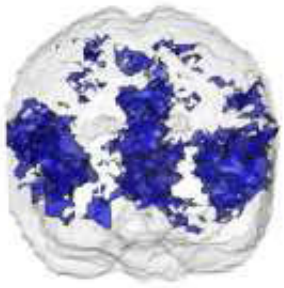


Figure 4.6: Pick number of clusters: the SSE drops very slowly when the number of clusters is over 5.

As a demonstration, the centers and spatial plots of two clusters are shown below.







4.5 Discussions and Future Work

By exploring two different class of features and a SVM classif er, we have developed a reliable fMRI base biomarker for physical pain. We believe there are some more space for improvement. A possible way is to design a deep learning algorithm which combines that feature selection and classif er learning steps. Another direction for improvement is use a penalized functional regression model, as proposed by Goldsmith et al. (2011), to obtain

better weights.

Chapter 5

Discussion and Future Work

The analysis of fMRI covers a wide ranges of topics, from the initial acquisition of raw data to its use in measuring brain activity, making inferences about brain activity and making predictions about physical, psychological and disease states. Statistics has already played a crucial role on many of the important issues. However, there are still areas where statistics has been underutilized and hopefully will have an increased role in the future.

In this research, three novel methods are proposed for fMRI data analysis. The methods have both advantages and shortcomings.

The f rst method, parallel group independent component analysis, or PGICA algorithm, is much faster than the sequential one with the help of parallel computing. Essentially we used parallel computing to turn a time intensive problem into constant time problem. The algorithm is also implemented in R and published on CRAN. For an overview of using R to analyze fMRI data, see Eloyan et al. (2014). Nonetheless, we still have not analyzed

the 1000 Functional Connectome Project dataset in its entirety. The main remaining obstacles are site-specific variations, which plague the quality of results. More specifically, functional imaging data collected in each data collection site have different features, such as population demographics, scanner types, data quality and so on. The data in each site have been collected for addressing specific research questions introducing issues while analyzing the data collectively. The factors for different sites interfere when analyzing data together. Thus we have found a degradation in the quality of results as more data is included. For future work, aggregating methods to properly combine data from different sites are needed.

The second method, penalized linear dynamical system (PLDS) is a very general model that can be applied to many applications. A parameter estimation algorithm is also developed based on the Expectation-Maximization algorithm (EM). It is worth mentioning that when the dimension is very high, the EM algorithm is not as robust, due to loss of precision in large matrix operations. Moreover, in the application, biological meaning of the found regions need to be further validated. It is helpful to compare those regions to other existing parcellations of mortor cortex. For example, we can develop a quantitative procedure to compare them to the five-region parcellation proposed by Nebel et al. (2014b).

The third method, a two-stage procedure to predict physical pain from fMRI scans, aims to connect brain activity to physical feelings. The procedure has higher prediction accuracy, compared to existing methods. The improvement is not huge, but is significant nonetheless. Considering that fMRI images are expensive to acquire, the improvement is

meaningful. In the future, the method should be applied to data collected from different sites. Thus we can evaluate whether it continues to achieve better prediction across different datasets.

A1 Appendix to Chapter 3

Kalman Filter Smoother

Algorithm Kalman Filter Smoother

0. Def ne $\mathbf{x}_t^\tau = \mathrm{E}(\mathbf{x}_t|\{\mathbf{y}\}_1^\tau)$,$\mathbf{V}_t^\tau = \mathrm{Var}(\mathbf{x}_t|\{\mathbf{y}\}_1^\tau)$, $\widehat{\mathbf{x}}_t \equiv \mathbf{x}_t^T$ and $P_t \equiv V_t^T + \mathbf{x}_t^T \mathbf{x}_t^{T\prime}$

1. Forward Recursions:

$$\mathbf{x}_t^{t-1} = A\mathbf{x}_{t-1}^{t-1}$$

$$\mathbf{V}_t^{t-1} = A\mathbf{V}_{t-1}^{t-1} + \mathbf{Q}$$

$$K_t = \mathbf{V}_t^{t-1} C'(CV_t^{t-1}C' + R)^{-1}$$

$$\mathbf{x}_t^t = \mathbf{x}_t^{t-1} + K_t(\mathbf{y}_t - C\mathbf{x}_t^{t-1})$$

$$V_t^t = V_t^{t-1} - K_t C V_t^{t-1}$$

$$\mathbf{x}_1^0 = \pi_0,\ V_1^0 = \mathbf{V}_0$$

2. Backward Recursions:

$$J_{t-1} = V_{t-1}^{t-1} A'(V_t^{t-1})^{-1}$$

$$\mathbf{x}_{t-1}^T = \mathbf{x}_{t-1}^{t-1} + J_{t-1}(\mathbf{x_t^T} - \mathbf{A}\mathbf{x_{t-1}^{t-1}})$$

$$V_{t-1}^T = V_{t-1}^{t-1} + J_{t-1}(V_t^T - V_t^{t-1})J_{t-1}'$$

$$P_{t,t-1} \equiv V_{t,t-1}^T + \mathbf{x}_t^T \mathbf{x}_t^{T\prime}$$

$$V_{T,T-1}^T = (I - K_T C)AV_{T-1}^{T-1}$$

FISTA

In general, FISTA optimize a target function

$$\min_{x \in \mathcal{X}} \quad \mathbf{F}(\mathbf{x}; \lambda) = \mathbf{g}(\mathbf{x}) + \lambda \|\mathbf{x}\|_{\mathbf{1}} \tag{A.1.1}$$

where $\mathbf{g} : R^n \rightarrow R$ is a continuously differentiable convex function and $\lambda > 0$ is the regularization parameter.

A FISTA algorithm with constant step is detailed below

Algorithm FISTA($\mathbf{g}, \lambda$).

---

1. Input an initial guess $\mathbf{x_0}$ and Lipschitz constant $\mathbf{L}$ for $\nabla \mathbf{g}$, set $\mathbf{y_1} = \mathbf{x_0}, t_1 = 1$

2. Choose $\tau \in (0, 1/\mathbf{L}]$.

3. Set $k \leftarrow 0$.

4. **loop**

5. Evaluate $\nabla \mathbf{g}(\mathbf{y_k})$

6. Compute $\mathbf{x_1}$= $\mathbf{S}_{\tau\lambda}(\mathbf{y_k} - \tau \nabla \mathbf{g}(\mathbf{y_k}))$

7. Compute $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$

8. $\mathbf{y_{k+1}} = \mathbf{x_k} + \left(\frac{t_k - 1}{t_{k+1}}\right)\Big(\mathbf{x_k} - \mathbf{x_{k-1}}\Big)$

9. Set $k \leftarrow k + 1$

10. **end loop**

In the above

$$\mathbf{S}_\lambda(\mathbf{y}) = (|\mathbf{y}| - \lambda)_+ \mathbf{sign}(\mathbf{y}) = \begin{cases} y - \lambda & \text{if } y > \lambda \\ y + \lambda & \text{if } y < -\lambda \\ 0 & \text{if } |y| \leq \lambda. \end{cases}$$

Bibliography

Allen, E., Erhardt, E., Damaraju, E., Gruner, W., Segall, J., Silva, R., Havlicek, M., Rachakonda, S., Fries, J., Kalyanam, R., et al. (2011). A baseline for the multivariate comparison of resting-state networks. *Frontiers in systems neuroscience*, 5.

Amari, S.-i., Cichocki, A., Yang, H. H., et al. (1996). A new learning algorithm for blind signal separation. *Advances in neural information processing systems*, pages 757–763.

Amdahl, G. M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference*, pages 483–485.

Andersen, A. H., Gash, D. M., and Avison, M. J. (1999). Principal component analysis of the dynamic response measured by fmri: a generalized linear systems framework. *Magnetic Resonance Imaging*, 17(6):795–815.

Arbabshirani, M. R., Damaraju, E., Phlypo, R., Plis, S., Allen, E., Ma, S., Mathalon, D., Preda, A., Vaidya, J. G., Adali, T., et al. (2014). Impact of autocorrelation on functional connectivity. *NeuroImage*.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202.

Beckmann, C. F. and Smith, S. M. (2005). Tensorial extensions of independent component analysis for multisubject fmri analysis. *Neuroimage*, 25(1):294–311.

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

Biswal, B. B., Mennes, M., Zuo, X.-N., Gohel, S., Kelly, C., Smith, S. M., Beckmann, C. F., Adelstein, J. S., Buckner, R. L., Colcombe, S., et al. (2010). Toward discovery science of human brain function. *Proceedings of the National Academy of Sciences*, 107(10):4734–4739.

Bode, D. (2012). *Rsge: Interface to the SGE Queuing System*. R package version 0.6.3.

Boots, B. (2008). Learning stable linear dynamical systems.

Calhoun, V., Adali, T., Pearlson, G., and Pekar, J. (2001). A method for making group inferences from functional mri data using independent component analysis. *Human brain mapping*, 14(3):140–151.

Calhoun, V. D., Liu, J., and Adalı, T. (2009). A review of group ica for fmri data and ica for joint inference of imaging, genetic, and erp data. *Neuroimage*, 45(1):S163–S172.

Catmull, E. and Clark, J. (1978). Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-aided design*, 10(6):350–355.

Di Martino, A., Yan, C., Li, Q., Denio, E., Castellanos, F., Alaerts, K., Anderson, J., Assaf, M., Bookheimer, S., Dapretto, M., et al. (2014). The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19(6):659–667.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., and Weingessel, A. (2008). Misc functions of the department of statistics (e1071), tu wien. *R package*, pages 1–5.

Doretto, G., Chiuso, A., Wu, Y. N., and Soatto, S. (2003). Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., and Vapnik, V. (1997). Support vector regression machines. *Advances in neural information processing systems*, 9:155–161.

Eavani, H., Satterthwaite, T. D., Gur, R. E., Gur, R. C., and Davatzikos, C. (2013). Unsupervised learning of functional network dynamics in resting state fmri. In *Information Processing in Medical Imaging*, pages 426–437. Springer.

Eloyan, A., Crainiceanu, C., and Caffo, B. (2013). Likelihood-based population independent component analysis. *Biostatistics*, 14(3):514–527.

Eloyan, A. and Ghosh, S. (2011). Smooth density estimation with moment constraints using mixture distributions. *Journal of nonparametric statistics*, 23(2):513–531.

Eloyan, A., Li, S., Muschelli, J., Pekar, J. J., Mostofsky, S. H., and Caffo, B. S. (2014).

Analytic programming with fmri data: A quick-start guide for statisticians using r. *PloS one*, 9(2):e89470.

Febrero-Bande, M. and Oviedo de la Fuente, M. (2012). Statistical computing in functional data analysis: the r package fda. usc. *Journal of Statistical Software*, 51(4):1–28.

Flynn, M. (1972). Some computer organizations and their effectiveness. *Computers, IEEE Transactions on*, 100(9):948–960.

Ghahramani, Z. and Hinton, G. E. (1996). Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science.

Goldsmith, J., Bobb, J., Crainiceanu, C. M., Caffo, B., and Reich, D. (2011). Penalized functional regression. *Journal of Computational and Graphical Statistics*, 20(4).

Goldsmith, J., Feder, J., Crainiceanu, C. M., Caffo, B., and Reich, D. (2010). Penalized functional regression.

Graves, S., Hooker, G., and Ramsay, J. (2009). Functional data analysis with r and matlab.

Gunn, S. R. et al. (1998). Support vector machines for classif cation and regression. *ISIS technical report*, 14.

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

Harman, H. (1976). *Modern factor analysis*. University of Chicago Press.

Havlicek, M., Friston, K. J., Jan, J., Brazdil, M., and Calhoun, V. D. (2011). Dynamic modeling of neuronal responses in fmri using cubature kalman f ltering. *Neuroimage*, 56(4):2109–2128.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hyvärinen, A. and Oja, E. (1997). A fast f xed-point algorithm for independent component analysis. *Neural computation*, 9(7):1483–1492.

Joel, S., Caffo, B., van Zijl, P., and Pekar, J. (2011). On the relationship between seed-based and ica-based measures of functional connectivity. *Magnetic Resonance in Medicine*, 66(3):644–657.

Jutten, C. and Herault, J. (1991). Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal processing*, 24(1):1–10.

Koch, W., Teipel, S., Mueller, S., Buerger, K., Bokde, A., Hampel, H., Coates, U., Reiser, M., and Meindl, T. (2010). Effects of aging on default mode network activity in resting state fmri: does the method of analysis matter? *Neuroimage*, 51(1):280–7.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Landman, B. A., Huang, A. J., Gifford, A., Vikram, D. S., Lim, I. A. L., Farrell, J. A., Bogovic, J. A., Hua, J., Chen, M., Jarso, S., et al. (2011). Multi-parametric neuroimaging reproducibility: A 3-t resource study. *Neuroimage*, 54(4):2854–2866.

Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224.

Lindquist, M. A. et al. (2008). The statistical analysis of fmri data. *Statistical Science*, 23(4):439–464.

Mattson, T., Sanders, B., and Massingill, B. (2004). *Patterns for parallel programming*. Pearson Education.

McKeown, M. J., Jung, T.-P., Makeig, S., Brown, G., Kindermann, S. S., Lee, T.-W., and Sejnowski, T. J. (1998). Spatially independent activity patterns in functional mri data during the stroop color-naming task. *Proceedings of the National Academy of Sciences*, 95(3):803–810.

Meier, J. D., Af alo, T. N., Kastner, S., and Graziano, M. S. (2008). Complex organization of human primary motor cortex: a high-resolution fmri study. *Journal of neurophysiology*, 100(4):1800–1812.

Michael, A., Anderson, M., Miller, R., Adal, T., and Calhoun, V. (2014). Preserving subject

variability in group fmri analysis: performance evaluation of gica vs. iva. *Front Syst Neurosci*, 8:106.

Nebel, M., Eloyan, A., Barber, A., and Mostofsky, S. (2014a). Precentral gyrus functional connectivity signatures of autism. *Frontiers in systems neuroscience*, 8.

Nebel, M. B., Joel, S. E., Muschelli, J., Barber, A. D., Caffo, B. S., Pekar, J. J., and Mostofsky, S. H. (2014b). Disruption of functional organization within the primary motor cortex in children with autism. *Human brain mapping*, 35(2):567–580.

Nishimoto, S., Vu, A. T., Naselaris, T., Benjamini, Y., Yu, B., and Gallant, J. L. (2011). Reconstructing visual experiences from brain activity evoked by natural movies. *Current Biology*, 21(19):1641–1646.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Rabiner, L. and Juang, B.-H. (1986). An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16.

Rauch, H. (1963). Solutions to the linear smoothing problem. *Automatic Control, IEEE Transactions on*, 8(4):371–372.

Roweis, S. and Ghahramani, Z. (1999). A unifying review of linear gaussian models. *Neural computation*, 11(2):305–345.

Shumway, R. H. and Stoffer, D. S. (1982). An approach to time series smoothing and forecasting using the em algorithm. *Journal of time series analysis*, 3(4):253–264.

Simon, L. S. (2012). Relieving pain in america: A blueprint for transforming prevention, care, education, and research. *Journal of Pain and Palliative Care Pharmacotherapy*, 26(2):197–198.

Smith, S., Jenkinson, M., Woolrich, M., Beckmann, C., Behrens, T., Johansen-Berg, H., Bannister, P., De Luca, M., Drobnjak, I., Flitney, D., et al. (2004). Advances in functional and structural mr image analysis and implementation as fsl. *Neuroimage*, 23:S208–S219.

Smith, S. M., Hyvärinen, A., Varoquaux, G., Miller, K. L., and Beckmann, C. F. (2014). Group-pca for very large fmri datasets. *NeuroImage*, 101:738–749.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222.

Suykens, J. A. and Vandewalle, J. (1999). Least squares support vector machine classif ers. *Neural processing letters*, 9(3):293–300.

Tierney, L., Rossini, A. J., Li, N., and Sevcikova, H. (2012). *snow: Simple Network of Workstations*. R package version 0.3-10.

Tikhonov, A. N. (1943). On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, volume 39, pages 195–198.

Turlach, B. A., Venables, W. N., and Wright, S. J. (2005). Simultaneous variable selection. *Technometrics*, 47(3):349–363.

Valdés-Sosa, P. A., Sánchez-Bornot, J. M., Lage-Castellanos, A., Vega-Hernández, M., Bosch-Bayard, J., Melie-García, L., and Canales-Rodríguez, E. (2005). Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1457):969–981.

Van Overschee, P. and De Moor, B. (1994). N4sid: Subspace algorithms for the identif cation of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93.

Vapnik, V., Golowich, S. E., and Smola, A. (1997). Support vector method for function approximation, regression estimation, and signal processing. *Advances in neural information processing systems*, pages 281–287.

Wager, T. D., Atlas, L. Y., Lindquist, M. A., Roy, M., Woo, C.-W., and Kross, E. (2013). An fmri-based neurologic signature of physical pain. *New England Journal of Medicine*, 368(15):1388–1397.

Wang, H. E., Bénar, C. G., Quilichini, P. P., Friston, K. J., Jirsa, V. K., and Bernard, C. (2014). A systematic framework for functional connectivity measures. *Frontiers in neuroscience*, 8.

Yarkoni, T., Poldrack, R. A., Nichols, T. E., Van Essen, D. C., and Wager, T. D. (2011).

Large-scale automated synthesis of human functional neuroimaging data. *Nature methods*, 8(8):665–670.

Zhu, M. and Ghodsi, A. (2006). Automatic dimensionality selection from the scree plot via the use of prof le likelihood. *Computational Statistics & Data Analysis*, 51(2):918–930.

BIBLIOGRAPHY

SHAOJIE CHEN

schen89@jhu.edu

615 N. Wolfe St. E3035

Baltimore, MD 21205

http://www.biostat.jhsph.edu/∼shachen

Date of Birth: Sep 2nd, 1989

Place of Birth: HeBei, China

EDUCATION

2011 - 2015 **Johns Hopkins Bloomberg School of Public Health**, Baltimore, MD

Ph.D. in Biostatistics

Thesis title: *Statistical Methods To Analyze Massive High-Dimensional Neuroimaging Data*

Advisor: Prof. Brian Caffo

2014 **Johns Hopkins University**, Baltimore, MD

M.S.E in Computer Science

2010 **Hong Kong Baptist University**, Hong Kong, China

Summer Researcher

2007 - 2011 **Tsinghua University**, Beijing, China

B.S. in Mathematics

PROFESSIONAL EXPERIENCE

06/2014 - 08/2014	**Quantitative Summer Associate**
	Barclays Capital, New York City, NY
2011 - now	**Level 3 Candidate**
	CFA Institute
2011 - 2012	**Organizer and Member**
	Johns Hopkins University Quantitative Club, Baltimore, MD
2011 Summer	**Independent Contract Analyst**
	Stanford School of Business, Stanford, CA

HONORS AND AWARDS

JOHNS HOPKINS UNIVERSITY

2014	Joseph Zeger Conference Travel Award
2011-15	Department of Biostatistics Graduate Fellowship

Tsinghua UNIVERSITY

2011 Hong Kong PhD Fellowship (1 million HKD, 135 out of over 10,000)

2009 First Class Academic Scholarship (top 5%)

2008 First Class Academic Scholarship (top 5%)

PUBLICATIONS

PUBLISHED/SUBMITTED

Chen S, Huang L, Qiu H, Nebel MB, Mostofsky S, Pekar J, Lindquist M, Eloyan A, Caffo BS (2015). A Parallel Group Independent Component Analysis Algorithm for Massive fMRI Data Analysis. *Submitted to Neuroimage*.

Chen S, Vogelstein J, Lee S, Lindquist M, Caffo BS (2015). A Sparse High Dimensional State-Space Model with an Application to Neuroimaging Data. *Submitted to Journal of the American Statistical Association: Application*.

Yue C, Xu Y, **Chen S**, Goldsmith J, Caffo BS, Zipunnikov V (2014). Multilevel Binary Principal Component Analysis with Application to Brain Activity. *Submitted to Nerual Information Processing Systems*.

Coughlin JM, Wang Y, Munro CA, Ma S, Yue C, **Chen S** et al. (2014) Neuroinf ammation and brain atrophy in former NFL players: An in vivo multimodal imaging pilot study. *Neurobiology of Disease*. doi: 10.1016/j.nbd.2014.10.019.

Yue C, **Chen S**, Sair HI, Arian R, Caffo BS (2014). Estimatin a graphical intra-class correlation coeff cient (GICC) using multivariate probit linear mixed models. *Computational Statistics and Data Analysis*.

Pontone G, **Chen S**, Mari Z, Marsh L, Robins P, Williams J, Bassett S (2014). The Longitudinal Impact of Depression on Disability in Parkinson's disease. *Submitted to The American Journal of Geriatric Psychiatry*.

Li S, **Chen S**, Yue C, Caffo BS (2014). Independent Component Analysis through Fast Nonparametric Density Estimation. *Submitted to Frontiers of Neuroscience*.

WORKING PAPERS

Chen S, Deng D, Caffo BS, Lindquist M. Predicting Human Pain From fMRI Signal for Brain Activities.

PRESENTATIONS

2015 A Sparse High Dimensional State-Space Model with an Application to Neuroimaging Data. ENAR Spring Meeting, Miami, FL

2014 A Sparse High Dimensional State-Space Model with an Application to Neuroimaging Data. Department of Neuroscience. Johns Hopkins Brain Science Institute. November 25, Baltimore, MD.

03/2014 A Parallel Group Independent Component Analysis Algorithm for Massive fMRI Data Analysis. ENAR Spring Meeting, Baltimore, MD.

08/2013 Search for Default Network Using Likelihood-Based Population Independent Component Analysis. JSM, Montreal, Canada

---

TEACHING

2015 Biostatistics For Laboratory Scientists, Graduate, 140.615

2014 Statistical Methods in Public Health **I-II**, Graduate, 140.621-622.

2013 Introduction to SAS Statistical Package, Graduate

2013 Bayesian Methods **I-II**, Graduate, 140.751-752, Prof. Hongkai Ji

2012 Statistical Reasoning in Public Health II, Graduate, Prof. John McGready

2012 Statistical Reasoning in Public Health I, Graduate, Prof. John McGready

2012 Data Science Workshop II, Graduate, Prof. John McGready

2012 Data Science Workshop I, Graduate, Prof. John McGready