MODEL BASED PRINCIPAL COMPONENT ANALYSIS WITH APPLICATION TO FUNCTIONAL MAGNETIC RESONANCE IMAGING

by
Magnus O. Ulfarsson

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2007

Doctoral Committee:

Professor Alfred O. Hero III, Co-Chair
Professor Victor Solo, Co-Chair
Professor Jeffrey A. Fessler
Professor Douglas C. Noll

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor Prof. Victor Solo for his support, valuable advise, and constant interest in my work. Many thanks to the Phd committee members Prof. Hero, Prof. Fessler, and Prof. Noll for their input and suggestions. I wish to thanks my friends, and colleagues: Joonki Noh, Valur Olafsson, Dr. Luis Hernandez, and Kiran Pandey for discussions about fMRI that greatly enhanced my understanding. Finally I would like to thank my wife, parents, and family for their patience and support.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

LIST OF ACRONYMS

fMRI Functional Magnetic Resonance Imaging

PCA Principal Component Analysis

nPCA Noisy PCA

BOLD Blood Oxygen Level Dependency

MR Magnetic Resonance

LTI Linear Translation Invariant

ASL Arterial Spin Labeling

IFFT Inverse Fast Fourier Transform

RV Random Variable

SNR Signal to Noise Ratio

EM Expectation Maximization

STD Standard Deviation

CNS Contrast to Noise Ratio

voxel Volume element

PET Positron Emission Tomography

SPECT Single Photon Emission Computed Tomography

ROI Region Of Interest

kPCA Kernel PCA

ICA Independent Component Analysis

iid independent identically distributed

BIC Bayesian Information Criterion

ML Maximum Likelihood

MLE Maximum Likelihood Estimator

BLUP Best Linear Unbiased Prediction

AIC An Information Criterion

MDL Minimum Description Length

RMT Random Matrix Theory

SURE Stein's Unbiased Risk Estimator

ECD Empirical Cumulative Distribution

MP Marchenko-Pastur

MSE Mean Square Error

DFT Discrete Fourier Transform

DCT Discrete Cosine Transform

LRT Likelihood Ratio Test

AFNI Analysis of Functional Neuro-Imaging

sPCA Sparse PCA

slPCA Sparse Loading PCA

svPCA Sparse Variable PCA

svnPCA Sparse Variable Noisy PCA

SVD Singular Value Decomposition

fPCA Functional PCA

FDA Functional Data Analysis

LASSO Least Absolute Shrinkage and Selection Operator

a.s. Almost Surely

CRB Cramer-Rao Bound

EPI Echo Planar Imaging

CCA Canonical Correlation Analysis

SCotLASS Simplified Component Technique LASSO

LARS Least Angle Regression

GLOSSARY

A^T: Transpose of a matrix A.

$Y = [y_{t,v}]$: $T \times M$ fMRI data matrix where T is # time points and M is # of voxels.

y_t^T: A row vector of Y representing brain image observed at time t.

$y_{(v)}$: A column vector of Y representing voxel timeseries observed at voxel # v.

S_y: Data covariance matrix.

L: $r \times r$ diagonal matrix of sample eigenvalues.

Q: Matrix of eigentimeseries.

P: Matrix of eigenimages.

Ω: Modeled covariance matrix.

G: $M \times r$ loading matrix.

F: $M \times r$ orthonormal loading matrix.

λ_j: The variance of noisy principal component number j.

Φ: $T \times m$ matrix of m smooth basis functions.

A^+: The Moore-Penrose inverse of a matrix A.

r: The number of principal components.

$I(x)$: The indicator function $I(x) = 1$ if $x \in S$ zero else.

1_T: A T vector of 1's.

e_v: Is a vector containing zeros except that element v is one.

X: $T \times p$ regression matrix.

ϵ_t: Noise.

σ^2: Noise variance.

γ: Ratio of the # of time points to the # of voxels.

$\hat{\theta}$: An estimate of parameter θ.

$l_\theta(y)$: The log-likelihood function.

$\dim(\theta)$: The degrees of freedom for parameter θ.

$\text{diag}(a_1, ..., a_M)$: $M \times M$ diagonal matrix with $a_1, ..., a_M$ in its diagonal.

dg(A): Is a matrix same size as A with the off-diagonal elements zeroed out.

$|A|$: The determinant of a matrix A.

$\text{tr}(A)$: The trace of a matrix A.

dA: The first differential of a matrix A.

h: Sparseness tuning parameter.

$(a)_+$: Equal to zero if $a \leq 0$, else equal to a.

$\text{sgn}(a)$: The sign of a.

$x \sim N(\mu, \Sigma)$: x is a sample from a Gaussian dist. of mean μ, and covariance Σ.

$\|x\|_p$: The p-norm of x defined by $\left(\sum_{t=1}^{T} x_t^p\right)^{1/p}$.

$\odot$: The Hadamard product $[A \odot B]_{ij} = a_{ij} b_{ij}$.

CHAPTER I

INTRODUCTION

Functional Magnetic Resonance Imaging (fMRI) refers to the use of a Magnetic Resonance (MR) scanner to measure and map brain function by means of rapid acquisition of brain state images. Just like the older Positron Emission Tomography (PET) technique, fMRI is noninvasive and makes it possible to indirectly observe brain activity while a subject performs particular task. However, unlike PET, fMRI does not require injection of radioactive tracer and provides relatively good spatial and temporal resolution. For these reasons, fMRI has proven to be an invaluable tool for Neuroscientists and Psychologists to better understand human brain organization and function.

Blood Oxygen Level Dependency (BOLD) fMRI is the dominant fMRI technique. BOLD fMRI is based on the fact that the magnetic susceptibilities of oxyhemoglobin and deoxyhemoglobin differ slightly, i.e., the signal decay rate of deoxyhemoglobin is more rapid than its oxygenated counterpart. Following neural activation, oxygen rich blood flows to the area of activation which leads to a local changes in the MR decay parameter T_2^*. This subsequently leads to a local rise in the intensity of the observed signal. The intensity rise can be detected [110] and used to generate a BOLD-weighted image.

The BOLD imaging described here is by no means the only way to image functional activity. Other methods exist, e.g., Arterial Spin Labeling (ASL) which allows for the weighting of the MR signal by cerebral blood flow. A detailed comparison of BOLD and ASL is performed in [33].

It is important to keep in mind that the neural activity is indirectly observed through the BOLD response. The details of this forward problem, i.e., the relationship between the stimulus, neural firing, and the BOLD response is still largely unknown, see [108] for a diagram that summarizes the current knowledge.

The BOLD response is usually modeled as a hump like function that reaches maximum in about 5 seconds and then dies out in 10 seconds [23]. On a practical signal processing level it is typically assumed that it is related to the neural activity through a Linear Translation Invariant (LTI) system. This assumption has been shown to be accurate to first order [18, 161].It provides framework to solve the inverse problem, i.e., infer about the neural activity by observing the BOLD response. Buckner [19] discusses current advances and the limits of this inverse problem.

A small initial decrease or dip in the BOLD signal has been observed in some fMRI experiments. These experiments suggest that the dip occurs close to the source of neural activity which hints that higher spatial resolution can be obtained. For example, a spatial map of the iso-orientation columns in the visual cortex requiring sub-millimeter accuracy has been produced [155]. In comparison, the BOLD response can provide 3-5 mm resolution. However, the existence of this dip is still controversial due to lack of reproducibility of these experiments.

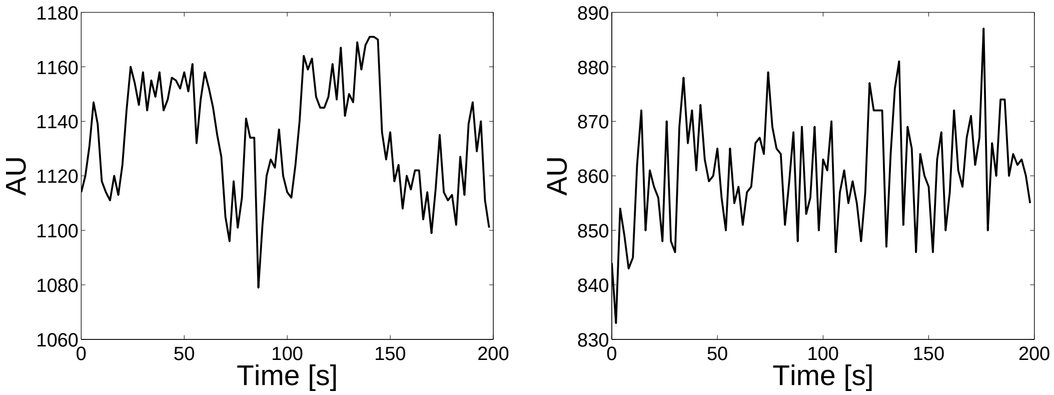
The main type of experimental design in fMRI is the so-called blocked experimental design. A typical blocked experiment usually consists of two states; the control state and a functional state. The functional state could for example involve finger tapping or a visual fixation on a flickering image, while the rest state involves no motor action or visual fixation on non-flickering image. The blocked experimental design to detect activation is a heritage from a time when Single Photon Emission Computed Tomography (SPECT) and PET were the dominant methods to do functional studies. In those method a particular functional state had to be applied for up to a minute to detect activation. In contrast, activation is detectable seconds after a stimuli onset in fMRI. Therefore, there is an interest in event-related experiments where a multiple brief stimulus is applied during a single experiment [123, 22].

Preprocessing of the data is necessary after the MR signal is recorded. It usually involves solving number of ill-conditioned inverse problems. The first step is reconstruction which in the simplest setting is a two-dimensional Inverse Fast Fourier Transform (IFFT). The result is a time series of brain images which generally consist of complex numbers. The phase is almost always discarded in subsequent statistical analysis by taking the magnitude of the images. This is justified by the fact that the Rician Random Variable (RV) resulting from taking the magnitude of a complex Gaussian RV is very well approximated by a Gaussian RV for Signal to Noise Ratio (SNR) greater than 2 [61]. This magnitude of SNR is much less than commonly seen in fMRI, except outside the brain, and more importantly in regions affected by signal dropout. With increasing spatial resolution, and consequently decreasing SNR, methods that incorporate the phase will become important. Anticipating this, researchers have started to develop models for low SNR data; [125] developed estimation method for complex data, and [139] developed an Expectation Maximization (EM) algorithm for Rician distributed fMRI magnitude data.

Other standard preprocessing steps are: physio-correction to correct for cardio and respiratory signals [58], which depends on the availability of external measurements of the signals; motion correction to correct for unexpected movement of the subject during the experiment [47]; slice time correction that corrects for slice acquisition error; spatial normalization to register brain images from different individuals into a standard brain template, e.g., Talairach [146] or MNI [41]; and high pass filtering [47]. Sometimes spatial smoothing is applied, however, this is often poorly motivated. There are many issues related to preprocessing that are unanswered, for instance in what order should they be performed. The correct approach would be to incorporate them into a statistical model, but currently this seems computationally unfeasible.

The resulting sequence of images is four dimensional, three spatial dimensions and time. A particular spatial location in a 3-D grid is called a voxel and a two dimensional cross-sectional grid is called a brain slice. There are two plots that can be used to visualize the data; the time series associated with a voxel, and an image of a brain slice. Figure 1.1 shows an example of two voxel timeseries: activated, and noisy. There are three things to notice here. Firstly, the y-axis is in arbitrary units, i.e., the voxel timeseries do not have a physical meaning. Secondly, the SNR which is defined by the ratio of the baseline amplitude to the noise Standard Deviation (STD) is clearly very high. Thirdly, the so-called Contrast to Noise Ratio (CNS) which is defined as the ratio between the amplitude of the activated signal (the square wave looking signal on Figure I.1(a)) and noise STD is relatively low.

Figure 1.2 shows an example of a brain image slice at a particular time. This is a horizontal slice (axial) where the back of the head is in the lower part of the image. As with the voxel timeseries the voxel intensities are measured in arbitary units. fMRI images are usually interpreted in relative terms, e.g., for our example voxels in the grey matter of the brain are of higher intensity than white matter voxels. Voxels outside the brain are of lowest intensity. The artifacts noticeable outside the brain



(a) Activated voxel timeseries

(b) Noisy voxel timeseries

Figure 1.1: Plots of timeseries observed at two different voxels.

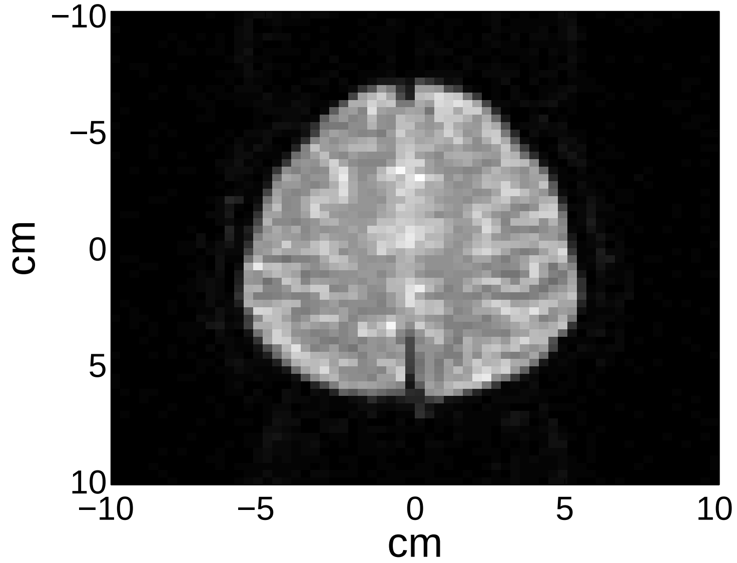


Figure 1.2: A picture of a brain slice observed at a particular time.

are due to the Fourier based reconstruction method used in this example.

Another important aspect of fMRI data is that the number of voxels greatly exceeds the number of time points, e.g., a typical data set has for example 21 brain slices, 64×64 voxels per brain slice, and 100 time points. The number of data points is therefore $21 \times 64 \times 64 \times 100 = 8601600$ which is a very large number.

Even after preprocessing the fMRI data is extremely complex. For instance, Weisskoff [166] showed for a slice in the visual cortex, obtained by fast 7Hz sampling, that the spectra displayed white noise in white matter, cardiac noise in cerebrospinal fluid, and cardiac, respiratory and low-frequency noise in grey matter. Slower sampling rates mean that relatively high frequency processes such as the cardiac rhythm are aliased, further adding to the complexity [90]. To complicate the story even further the fMRI data is also spatially correlated, e.g., due to motion correction algorithms and reconstruction. For these reasons no universally accepted model for fMRI data currently exists.

An interesting feature of fMRI time series is the presence of a low frequency noise. The underlying source is largely unknown. Its main characteristics are [109] 1) it occurs without stimulus, 2) it can be differentiated from the cardiac and respiratory signals, 3) and it can be altered by pharmacological and pathological conditions. An especially interesting aspect of this is the investigation of functional connectivity of resting state data [17, 60].

The two most common approaches to analyze fMRI data are univoxel and multivoxel methods. Univoxel approaches assume spatial independence and consequently model each voxel separately. The voxels are assumed to comprise of BOLD response and noise and possible some nuisance signals. On the other hand multivoxel approaches are generally exploratory and use all the voxels (or some subset) to produce few spatial maps and corresponding temporal signals that (hopefully) capture the essence of the data set well. They can be broadly divided into methods that do not recognize stimulus such as Principal Component Analysis (PCA) [53], Independent Component Analysis (ICA) [96], Self Organizing Maps (SOM) [113], and methods that recognize stimulus such as Partial least squares (PLS) [95] and Canonical Correlation Analysis (CCA) [50].

An underlying theme in this thesis is the realization that there is an underlying model behind PCA [81, 5], which we call the noisy PCA (nPCA), where the PCs can be derived by the Maximum Likelihood (ML) method. This fact gives us an access to a range of statistical machinery such as inference and model selection methods, and a framework to build upon.

In this dissertation we develop three novel features

1. Chapter II develops two Noisy PCA (nPCA) based spatio-temporal models that recognizes temporal smoothness and spatial localization of the fMRI data. Unlike univoxel method these methods do not assume stationary noise. In addition, we demonstrate how to construct a Likelihood ratio test statistic, and importantly, show how to obtain a spatial decomposition of it. The tuning parameters associated with these models are selected jointly by using the BIC criterion.

2. In Chapter III we propose a new svPCA algorithm, which we call svnPCA. Our method has several novel features. Firstly, it is based on a statistical model. Secondly, it uses a penalized likelihood formulation that is able to zero out variables rather than just loadings. Thirdly the optimization problems is nonstandard because it involves an orthogonality constraint and we resolve this by using the geodesic steepest descent and geodesic Newton algorithms. Finally, to select the number of sparse principal components and the sparseness tuning parameter, we propose to use a novel form of the BIC criterion. In addition, we discuss an alternative svnPCA approach using the EM algorithm.

3. Chapter IV introduces a novel method to select the number of principal components based on the nonlinear SURE technique with some help from Random Matrix Theory (RMT). This method is able to handle the case where the num-

ber of observations are of similar order as number of variables. For practical use it is necessary to estimate the noise variance and we have developed a reliable estimator based on RMT.

Chapter V draws conclusions and discusses possible future research directions.

1.1 Univoxel Methods

The goal of univoxel methods is to construct so-called activation maps from the fMRI data. Activation maps are constructed by comparing the voxel timeseries to the experimental stimulus. Voxels that correspond closely to the stimulus are considered to be activated. In univoxel modeling the voxels are considered separately and assumed independent of each other. Spatial correlation is usually ignored, notable exceptions are [136, 116].

In the following, to fix ideas imagine that $y_{(v)}$ is the signal depicted on Figure I.1(a). An observed voxel timeseries (after preprocessing) at voxel number v is typically modeled as follows

$$\begin{aligned} y_{(v)} &= Z_1\gamma_v + Z_2\delta_v + \eta_v \qquad (1.1) \\ &= X\beta_v + \eta_v, \quad v = 1, ..., M \end{aligned}$$

where Z_1 is a $T \times q_1$ matrix that accounts for the BOLD response, Z_2 is a $T \times q_2$ matrix to account for nuisance parameters such as drift, $X = [Z_1, Z_2]$, $\beta_v = [\gamma^T, \delta^T]^T$, and $\eta_v \sim N(0, \Omega_v)$ is a noise term.

1.1.1 The BOLD Response Model

The BOLD response is considered the part of the observed voxel timeseries that reacts to the experimental stimulus. A good model for the BOLD response is necessary in order to get a good tradeoff between bias and variance in the estimate. Too simple model will lead to a biased estimate and too complex model will lead to a high variance estimate. The main issue concerning the construction of the BOLD response model is the relationship of the experimental stimulus to the BOLD response. The papers [56], [27] and [138] discuss the case where the stimulus is assumed to be linearly related to the BOLD signal while e.g., [136] consider the non-linear case.

1.1.2 The Noise Model

Due to the relatively short length of the voxel timeseries the noise models usually require a low order stationary parametric models. The paper [20] uses AR(p) model, i.e., the covariance matrix Ω_v is a Toeplitz matrix. The paper [138] suggests ARMA(1,1) model for the noise.

1.1.3 Estimation

The generalized least square estimate can be used to estimate the parameters β_v. It is given by

$$\hat{\beta}_v = (X^T \Omega_v^{-1} X)^{-1} X^T \Omega_v^{-1} y_{(v)}, \quad v = 1, ..., M.$$

The solution does require the covariance Ω_v, which is generally not available, so an estimate needs to be used. This complicates the problem since now β_v and Ω_v need to be estimated jointly, which is usually done using iterative methods.

1.1.4 Inference and Activation maps

Activation maps are constructed by doing hypothesis test at each voxel. Lets assume that $\beta_v = [\beta_{1,v}, \beta_{2,v}]^T$, $\beta_{1,v}$ controls the baseline, and $\beta_{2,v}$ is the parameter that controls the amplitude of the BOLD response signal, then an appropriate hypothesis test would be

$$\begin{aligned} H_0 : \beta_{2,v} &= 0 \\ H_1 : \beta_{2,v} &\neq 0. \end{aligned}$$

Assuming white noise ($\Omega_v = \sigma_v^2 I_T$), the Likelihood Ratio Test (LRT) [15] is given by

$$\text{LRT}_v = \frac{1}{2}\beta_{2,v}^2 \|s\|^2 / \hat{\sigma}_v^2, \quad v = 1, ..., M. \tag{1.2}$$

where s is a vector modeling the BOLD response, and $\hat{\sigma}_v^2$ is of course an estimate of the noise variance. The LRT_v is a very sensible quantity, measuring the signal to noise ratio at voxel number v. Whether the voxel is activated or not is determined from this number. Correlations, multiple comparison problems greatly complicate the story in this section, but such issues are not discussed in this thesis.

1.2 Multivoxel Methods

1.2.1 Principal Component Analysis (PCA)

The best known multivoxel method is probably PCA [74]. It is based on finding uncorrelated linear combinations of the data that maximize variance. Its classical use in multivariate analysis is exploratory, i.e., to decompose relatively low dimensional data set, where the number of observations greatly exceed the number of variables, in hope that the PCs reveal something about the underlying process. References [74, 70] include many good examples. More theoretical references include [79, 129, 6].

PCA is based on determining a $M \times 1$ vector m, $M \times T$ matrices C and B that solve the following minimization problem:

$$\min_{m,B,C} \sum_{t=1}^{T} \|y_t - m - CB^T y_t\|^2. \tag{1.3}$$

Again to fix ideas imagine that y_t is the brain image shown on Figure 1.2 stacked in a vector, and that $M > T$.

The solutions to (1.3) are given by [129]:

$$\begin{aligned}\hat{m} &= \bar{y} = \frac{1}{T}\sum_{t=1}^{T} y_t \\ C &= B = P_r \\ Y - 1_T\bar{y}^T &= QL^{1/2}P^T.\end{aligned}$$

The last expression is the Singular Value Decomposition (SVD) of the mean corrected $T \times M$ fMRI data matrix, $Q = [q_{(1)}, q_{(2)}, ..., q_{(T)}]$ is a $T \times T$ orthonormal matrix, $P = [p_{(1)}, p_{(2)}, ..., p_{(M)}]$ is an $M \times M$ orthonormal matrix, $L^{1/2}$ is a $T \times M$ diagonal matrix of singular values $l_1^{1/2}, l_2^{1/2}, .., l_T^{1/2}$, and $P_r = [p_{(1)}, p_{(2)}, ..., p_{(r)}]$. It is sometimes useful to use these estimates and write the fMRI data matrix in the following way:

$$Y = 1_T\bar{y}^T + QL^{1/2}P^T.$$

A little algebra shows that

$$S_y = \frac{1}{T}(Y^T(I_T - \frac{1_T 1_T^T}{T})Y) = \frac{1}{T}PLP^T \tag{1.4}$$

i.e., P contains the eigenvectors of the covariance matrix S_y of the voxel timeseries. Since the columns $p_{(1)}, ..., p_{(M)}$ of the matrix P can be plotted spatially we call them the eigenimages of Y. By similar argument we call the columns $q_{(1)}, ..., q_{(T)}$ of Q the eigentimeseries of Y.

There is another way to view PCA, namely by the following maximization problem:

$$\max_{p_{(i)} \in R^M} p_{(i)}^T S_y p_{(i)} \tag{1.5}$$

subject to

$$p_{(i)}^T p_{(i)} = 1, p_{(i)}^T p_{(j)} = 0, i < j, j \geq 2.$$

The first eigenimage $p_{(1)}$ can be interpreted as the direction in R^M of maximum variance. The second eigenimage is orthogonal to the first and points in the direction in R^M of second most variance and so on. If we plot an eigenimage $p_{(j)}$ spatially then the spatial point of largest absolute amplitude represents the point of largest variation when $l_j^{1/2} q_{t,j} p_{(j)}^T$ is observed as a movie over time. If the eigenimage has high amplitude in the gray matter, e.g., in the motor cortex, it is of particular interest and should be looked more closely at. It is also of interest to look at the eigentimeseries $q_{(j)}$, which describe how the eigenimages evolve in time. For example if a eigentimeseries looks like the stimulus signal we are on to something.

The most common exploratory use of PCA in fMRI and medical imaging is to analyze functional connectivity. For example, if two brain regions show up highlighted on the same eigenimage they are said to be functionally connected.

Probably the first use of PCA to investigate functional connectivity in medical imaging was [53] which applied it to a PET data set coming from a verbal fluency experiment, [54] describes a similar fMRI experiment, in both cases the two first eigenimages explained most of the data variance and were biologically interpretable. Bullmore et al. [21] performs PCA on fMRI data from a experiment involving visual and sematic processing of words. They specialized to a Region Of Interest (ROI) selected via univoxel analysis of the data. The results were interpreted in terms of functional relationship between activated regions. Worsley et al. [170] compares PCA to conventional correlation methods to detect functional connectivity.

Other notable papers on the use of PCA in fMRI are firstly Mitra et al. [102] which advocates the use of spatio-frequency PCA, in addition the paper gives a nice signal processing insight in medical image processing. Secondly, Behzadi et al. [13] extracts PCs from regions of the brain in which neural activation is unlikely, such as in CSF and the white matter of the brain. These components are called significant PCs and are subsequently used to act as nuisance parameter regressors in SPM analysis. This method compared favorably to the RETROICOR method [58].

Finally, we point out the standard criticism of using PCA in fMRI. First, as noted by many of the above mentioned papers, there is no guarantee that the PCs are interpretable. In other words there is no biological reason why interesting brain processes should be orthogonal, let alone of high variance. For example the activation could be diffused over many components. Second, as noted by [170] there is still no principled way to obtain p-values for the eigenimages.

1.2.2 Selection of the Number of PCs

A crucial problem is to design a rule to decide how many eigentimeseries/eigenimages should be retained. To motivate this we write the fMRI data in the following way

$$Y = 1_T \bar{y} + \sum_{j=1}^{r} l_j^{1/2} q_{(j)} p_{(j)}^T + N \tag{1.6}$$

where $r < T$. In this case the $1_T \bar{y} + \sum_{j=1}^{r} l_j^{1/2} q_{(j)} p_{(j)}^T$ is considered the signal part and the matrix of residual N is considered to be the noise part. So the selection of the number of PCs is equivalent to selecting the r in Equation (1.6).

A popular ad hoc rule is to use the Scree plot [94], which plots the eigenvalues $l_1, ..., l_T$ in a decreasing order, and looks for an elbow where the signal eigenvalues $l_1, ..., l_r$ are on the left side and the noise eigenvalues $l_{r+1}, ..., l_T$ are on the right. Another approach is to compute the cumulative percentage of the total variation of the PCs:

$$\frac{\sum_{j=1}^{r} l_j}{\mathrm{tr}(S_y)}$$

and retain the number of PCs that represent, say 70% or 80%. A number of other similar methods exist [74], that often work well in practice, but their disadvantage is that they need a subjective decision from the user. This decision is often hard to make, e.g., there can be multiple elbows in the Scree plot.

More objective methods for choosing PCs have been proposed by several authors. The references [168, 37], in different context than fMRI, propose cross-validation to select the number of PCs so that a good prediction model is obtained. However, for large data sets such as for fMRI, the computations for cross-validation become prohibitive. Hansen et al. [63] split their fMRI data set into test and training data and use the test set to select the number of PCs that minimize prediction error. However, fMRI scanning time is expensive so it is questionable how useful this method is. The remarkable [115] was way ahead of its time in applying Random matrix theory (RMT) to PCA in the context of Meteorology and Oceanography. It develops simple selection rule based on keeping all components that lie above the 95% level of the of the cumulative distribution function expected from RMT for all noise data.

1.2.3 Rotated PCA

Starting from the SVD view of PCA $Y = Q_r L_r^{1/2} P_r^T$ it can be seen that this representation is invariant to rotation by a $r \times r$ rotation matrix R, i.e.,

$$Y = AS^T + N$$

where

$$\begin{aligned} A &= Q_r L_r^{1/2} R \\ S &= P_r R \end{aligned}$$

where N is a matrix of residuals, $A = [a_{(1)}, a_{(2)}, ..., a_{(r)}]$ is a $T \times r$ matrix of rotated eigentimeseries, and $S = [s_{(1)}, s_{(2)}, ..., s_{(r)}]$ is a $M \times r$ matrix of rotated eigenimages.

In the case where PCA does not yield interpretable eigenimages $p_{(1)}, p_{(2)}, ..., p_{(r)}$ it is sometimes possible to select a specific rotation matrix R to get more informative results. A two step procedure that consists of first doing PCA and then determine the rotation matrix is called rotated PCA. Perhaps the most well known approach to do this is varimax procedure [76] that consists of maximizing the quantity

$$\sum_{j=1}^{r} \sum_{t=1}^{T} [s_{(j)}^T (y_t y_t^T - S_y) s_{(j)}]^2 \tag{1.7}$$

subject to the orthogonality constraints $S^T S = I_r$. This can be interpreted as maximizing the time centered fourth moment of the eigenimages relative to the basis S [115]. More intuitively, this procedure looks for eigenimages with heavy tailed histogram, i.e., most voxels close to zero and a few larger ones. For an algorithm to solve (1.7) see [91, 115].

The increased interpretability of the rotated PCs comes at a cost. The rotated eigentimeseries A lose their uncorrelatedness property. This is easily seen by computing

$$S^T S = R^T (A^T A) R = R^T L R$$

which is not diagonal.

Another class of methods that we briefly mention is so-called oblique rotation PCA. In that case the rotation matrix R is not orthogonal but invertible. These methods rotate each eigenimage individually and judge the interpretability either by some cost function or by the eye. Examples of oblique rotation are the Promax technique [83] which is based on the Procrustes transformation and the Projection Pursuit [48] which is based on non-Gaussian projection where the non-Gaussianity is measured by so-called projection indices.

Notice that we have discussed these rotation methods in term of rotation of the eigenimages. It is also possible to rotate the eigentimeseries instead. Same discussion as above would follow.

There are a few examples of the use of rotated PCA in fMRI. Backfrieder et al. [7] uses oblique rotation to search for components representing brain activation. Andersen et al. [3] investigated projection pursuit in a fMRI experiment involving pharmacological stimulation in primates and showed that it allowed for more interpretable and lower dimensional representation of the data than PCA. Thomas et al. [149] discussed how Independent Component Analysis (ICA) and PCA with and without varimax rotation treat noise. PCA and rotated PCA were found to be better in separating random noise from brain activation than ICA. On the other hand, ICA was found to be better in separating structured noise from brain activation than the PCA based methods.

There is a close connection between rotated PCA and the Fast ICA method [69] that is worth mentioning since Fast ICA is very popular in fMRI research [25, 12]. Fast ICA looks for rotations of the PCs by optimizing the following cost function

$$\sum_{j=1}^{r}\sum_{t=1}^{T} G(s_{(j)}^T y_t) \tag{1.8}$$

subject to $S^T S = I_r$, and G is called a contrast function. Note that the varimax method is a special case of (1.8). The idea is that the contrast function is a measure of the non-Gaussianity of the inner product $s_{(j)}^T y_t$. We do not want Gaussian random variable since it has the largest entropy among all random variable of equal variance [28] and therefore it is least structured. A popular contrast function in practice is

$$G(u) = \log\cosh(u).$$

For alternative contrast functions and further discussion about ICA see [68, 71].

1.2.4 Kernel PCA

Kernel PCA (kPCA) [127] is a generalization of PCA where unlike PCA it used nonlinear functional of the data. The fMRI data can be written in terms of the orthogonal kPCs $s_{(1)}, s_{(2)}, ..., s_{(r)}$ in the following way:

$$\begin{aligned} Y &= AS^T + N \\ &= \sum_{j=1}^{r} a_{(j)} s_{(j)}^T + N. \end{aligned}$$

The kPCs are obtained by diagonalizing a generalized $M \times M$ covariance matrix $S_K = [d(y_{(i)}, y_{(j)})]$, which is called the kernel matrix, where $d()$ is some distance norm such that S_K is a positive definite matrix.

The paper [148] introduced kPCA into fMRI research, in that paper the kernel matrix was chosen as

$$S_K = [y_{(i)}^T y_{(j)} e^{\frac{corr(y_{(i)}, y_{(j)})-1}{\sigma_d}}],$$

where $corr$ stands for correlation, and σ_d controls the amount of nonlinearity in the kernel. For example $\sigma_d \to \infty$ corresponds to the classical PCA, while decreasing σ_d leads to more emphasis on highly correlated signals. An interesting consequence of picking low σ_d is that negatively correlated timeseries are treated as almost orthogonal and therefore will most likely not appear on the same spatial map. This is in sharp contrast with conventional PCA.

The temporal components are easily computed by linear regression from the kPCs

$$a_{(j)} = Y s_{(j)}, j = 1, ..., r.$$

In fMRI the number M of voxels is usually very high, this poses a significant practical problem since this method requires a diagonalization of $M \times M$ matrix. As a remedy, Thirion [148] suggested to preselect interesting voxels, e.g., from thresholded activation maps.

The parameters r and σ_d need to be selected. An independent identically distributed (iid) regression framework was assumed in [148] and employed the BIC criterion to select r. The selection of σ_d seems to be a significant problem. The kPC need to be recomputed for different values of σ_d which is a huge task.

kPCA is a promising technique in fMRI but much more work has to go into how to choose the kernel or equivalently the non-linear functionals, and how it relates to the biological processes we are after. The paper [52] which focuses on non-linear PCA could serve as a starting point.

1.2.5 Functional PCA

Functional PCA (fPCA) is a basic tool from Functional Data Analysis (FDA) [119]. The basic premise of FDA and fPCA is that the observations are not random vectors of independent observations but rather smooth functions. In the fMRI case, we view the signal observed at voxel v, as a continuous function $y_v(t)$. The fPCA finds an orthonormal set of eigenfunctions $q(t)$ that optimally explain the variance of the fMRI data set.

There are a few examples of the use of fPCA in fMRI. Solo et al. [140] and Long et al. [87] use it in multisubject settings to estimate spatially varying non-stationary noise covariance kernel. Viviani et al. [163] view the signals observed at each voxel as continuous function and perform fPCA. They show that fPCA has advantage over traditional PCA in capturing the BOLD response. Interestingly, they allow the number of basis function to vary over voxels by using generalized cross-validation [30] to select it at each voxel.

There are two basic ways to incorporate smoothness into the eigentimeseries. The first method is to express the original data as a linear combination of known basis functions such as Fourier, B-splines, and wavelets [162, 144]. Now we discuss the formulation.

Lets assume we have a basis ϕ and expand the voxel timecurves

$$y(t) = \phi^T(t)C$$

where $y(t)$ is $1 \times M$ vector, $\phi^T(t)$ is $1 \times m$ vector of basis components, and $C = (\Phi^T\Phi)^{-1/2}\Phi^T(Y - 1_T m^T)$ is a $m \times M$ matrix of basis coefficients. The corresponding time covariance kernel is given by

$$S_\phi(s,t) = \phi^T(s)CC^T\phi(t).$$

Lets further assume that the smooth eigenfunctions can be written in terms of the basis as

$$q(t) = \phi^T(t)B.$$

The eigenequation can be written as (cf. Equation (1.4))

$$\begin{aligned} \int S_\phi(s,t)q(t)dt &= \phi^T(s)CC^TWB \\ &= \phi^T(s)BD \end{aligned}$$

where $W = \int \phi(t)\phi(t)^T dt$. This holds for all s so the following matrix equation holds

$$C^TCWB = BD.$$

Notice that if the basis is orthonormal then $W = I_m$, and therefore the problem reduces to doing a spectral decomposition on the smooth sample covariance

$$(\Phi^T\Phi)^{-1/2}\Phi^T S_y \Phi(\Phi^T\Phi)^{-1/2}.$$

If the basis is not orthogonal an integration has to be performed to compute W. This is usually done by numerical quadrature methods. An important problem is to select the number of basis function and PCs. The most common methods used in FDA context seem to be visualization and cross-validation.

The second method to enforce smoothness is to incorporate it into the PCA itself. Since this method has not been applied in fMRI research we refer to Chapter 9 in [119] for a detailed discussion.

1.3 Canonical Correlation Analysis

The idea behind CCA is to analyze the relationship between two vectors of variables. Suppose we have the observations of two zero mean random vectors, i.e., the p dimensional vector x and the M dimensional vector y. The goal is then to find linear combinations Xa and Yb such that they have the largest possible correlation

$$\rho(a, b) = \frac{a^T S_{xy} b}{\sqrt{a^T S_x a b^T S_y b}}.$$

That is

$$\max_{a,b} \rho(a, b) \text{ subject to } a^T S_x a = b^T S_y b = 1.$$

where S_x and S_y are the sample covariance matrices of X and Y, S_{xy} is the cross-covariance between X and Y. It is possible to find k pairs $(a_{(j)}, b_{(j)}), \quad j = 1, ..., k, k = \min(M, p)$ of the canonical correlations vectors that satisfy the condition above given that $a_{(i)}^T S_x a_{(j)} = 0, \quad i \neq j$, and the same for the b vectors. The solutions of the optimization problem above are called the canonical correlations. It can be shown that [79] that the canonical correlation vectors can be found by first performing SVD on the cross correlation matrix R_{xy} between X and Y

$$R_{xy} = S_x^{-1/2} S_{xy} S_y^{-1/2} = UDV^T$$

where $D = \text{diag}(\rho_1, ..., \rho_r)$ is a diagonal matrix of the canonical correlations. Then the canonical correlation vectors $a_{(1)}, ..., a_{(k)}, b_{(1)}, ..., b_{(k)}$ can be found by

$$\begin{aligned} A &= S_x^{-1/2} V \\ B &= S_y^{-1/2} U \end{aligned}$$

where $A = [a_{(1)}, ..., a_{(k)}]$ and $B = [b_{(1)}, ..., b_{(k)}]$. If either S_x of S_y are singular then the equations above can be modified by exchanging the generalized inverse [120] for the inverse.

Similar to the other multivariate methods discussed above one hopes that only few canonical variates YB show effect of stimulus and the loading map B show where the activation is.

CCA has only recently received attention in the fMRI literature. Friman et al [49] uses CCA to extract temporal and spatial signals that are maximally autocorrelated and claims that all signals of interest are autocorrelated. In [50] the x-set is a signal subspace that contains for example a prototype of the BOLD-response and other signals of interest. The y-set contains a voxel and its neighborhood, for example 9×9 region around the pixel. Friston [55] uses CCA to develop test statistics about the whole brain volume using the Wilks statistic [129].

1.4 Noisy PCA

Since fMRI data is spatio-temporal there are two possible nPCA models for it. Firstly, temporal nPCA where we view the brain scans or images as independent observations. Secondly, spatial nPCA where the voxels are viewed as independent observations. Both formulations are important to this thesis. For instance, Chapter 3 focuses on the spatial model and Chapter 4 on the temporal model. In this section we first develop temporal nPCA, then the spatial nPCA model is discussed.

1.4.1 Temporal nPCA

Like previously discussed the traditional PCA is exploratory. Lawley [82] demonstrated that there is a statistical model behind it which has the following sample function description

$$\begin{aligned} y_t &= \mu_t + \epsilon_t \qquad (1.9) \\ &= m + Gu_t + \epsilon_t, \quad t = 1, ..., T \end{aligned}$$

where

y_t :	$M \times 1$ brain image at time t
m :	The mean brain image
$G = (g_{(1)}, g_{(2)}, ..., g_{(r)})$:	$M \times r$ loading matrix
$u_t \sim N(0, I_r)$:	r vector of nPCs
$\epsilon_t \sim N(0, \sigma^2 I_M)$:	Isotropic noise
ϵ_t, u_t :	Independent random vectors.

The problem is to estimate $\theta = (m, G, \sigma^2)$. This model bears some resemblance with the standard array processing model [78], except in array processing, the loading matrix G is known except for few unknown parameters. So array processing methods are of no help here.

The log-likelihood is given by

$$l_\theta(y) = -\frac{T}{2}\mathrm{tr}(S_y\Omega^{-1}) - \frac{T}{2}\log|\Omega|$$

where $\Omega = GG^T + \sigma^2 I_M$. If M is large this log-likelihood expression is not useful for computation since a $M \times M$ matrix has to be inverted. We proceed to simplify it. By using the matrix inversion lemma we can write

$$\Omega^{-1} = \frac{1}{\sigma^2}(I_M - GW^{-1}G^T)$$

where $W = G^T G + \sigma^2 I_r$. By using the properties of the determinant we can write

$$\begin{aligned} \log|\Omega| &= \log|GG^T + \sigma^2 I_M| \\ &= \log|\sigma^{2M}(\frac{G^T G}{\sigma^2} + I_r)| \\ &= (M-r)\log\sigma^2 + \log|W|. \end{aligned}$$

By using the above expressions we can write the log-likelihood in the following way

$$\begin{aligned} l_\theta(y) &= -\frac{T}{2\sigma^2}\mathrm{tr}S_y + \frac{T}{2\sigma^2}\mathrm{tr}(W^{-1}G^T S_y G) \\ &- \frac{T(M-r)}{2}\log\sigma^2 - \frac{T}{2}\log|W|. \end{aligned} \tag{1.10}$$

The ML estimates [81, 147, 152] are given by (a derivation given in Appendix A for completeness)

$$\begin{aligned} \hat{m} &= \bar{y} = \frac{1}{T}\sum_{t=1}^{T} y_t \\ \hat{G} &= P_r(L_r - \hat{\sigma}^2 I_r)^{1/2} R \\ \hat{\sigma}^2 &= \hat{\sigma}_r^2 = \frac{1}{M-r}\sum_{j=r+1}^{M} l_j. \end{aligned} \tag{1.11}$$

Here $L_r = \mathrm{diag}(l_1, ..., l_r)$, where $l_1 > l_2 >, ..., > l_r$, contains the r largest eigenvalues of the data covariance matrix S_y, R is an arbitrary orthogonal rotation matrix, and

$$S_y = \frac{1}{T}\sum_{t=1}^{T}(y_t - \bar{y})(y_t - \bar{y})^T = \frac{1}{T}(Y^T - \bar{y}\mathbf{1}^T)(Y - \mathbf{1}\bar{y}^T)$$

where $\mathbf{1}$ denotes an M-vector of 1's, and the r columns of P_r are the corresponding eigenvectors, so $S_y P_r = P_r L_r$. The rotation matrix in the ML estimate is basically reflecting the important property that the log-likelihood is invariant to rotation of the loading matrix G, i.e., the loading matrix is not identifiable. This is usually resolved by imposing an identifiability constraint such as setting $R = I_r$. This particular choice is very appealing since in this case the loading matrix is orthogonal and the sample principal components are uncorrelated (see Equation (1.12)). In fact one of the defining properties of PCA is that the PCs are uncorrelated so it might be argued that the constraint $R = I_r$ is necessary.

Importantly, [147] proved that the ML estimates are the global optimizers of the likelihood, this was rediscovered by [152]. Also note that the asymptotic distribution of the ML estimators was derived in [4].

For given data and the ML estimate of θ, the estimated noisy principal components have to be estimated. A good estimate of it is the Best Linear Unbiased Prediction (BLUP) estimate [122], it and its associated covariance matrix is given by

$$\begin{aligned} \hat{u}_t &= E_{\hat{\theta}}(u_t|y_t) = \hat{W}^{-1}\hat{G}^T(y_t - \hat{m}) \\ S_{\hat{u}} &= \text{var}_{\hat{\theta}}(u_t|y_t) = \hat{\sigma}_r^2\hat{W}^{-1} \\ \hat{W} &= \hat{G}^T\hat{G} + \hat{\sigma}^2 I_r \end{aligned} \tag{1.12}$$

The estimate for μ_t in Equation (1.9) is given by

$$\begin{aligned} \hat{\mu}_t &= \hat{G}\hat{u}_t = \bar{y} + \hat{G}\hat{W}^{-1}\hat{G}^T(y_t - \bar{y}) \\ &= \bar{y} + \sum_{j=1}^{r} p_{(j)}\frac{l_j - \hat{\sigma}^2}{l_j}p_{(j)}^T(y_t - \bar{y}) \end{aligned} \tag{1.13}$$

this is a nonlinear function of y_t.

1.4.2 Noisy PCA Model Selection

An issue that needs to be resolved is to determine how many nPCs should be retained, i.e., r needs to be selected. The methods introduced Section 1.2.2 can of course be used, but the ML framework suggests new techniques. The first solution approach based on the ML framework was to use a nested sequence of hypothesis tests to test if the smallest $M-r$ eigenvalues are equal [10, 82]. Since the hypothesis testing approach depends on subjective choice of a threshold, the paper [165] suggested to use methods based on the application of the information theoretic criteria such as An Information Criterion (AIC) [1], Bayesian Information Criterion (BIC) [128], and Minimum Description Length (MDL) [121]. They are given by (MDL=BIC in this case)

$$\begin{aligned}
\mathrm{AIC}_r &= -2l_{\hat{\theta}}(y) + 2\dim(\theta) \\
\mathrm{BIC}_r &= -l_{\theta}(y) + \frac{1}{2}\dim(\theta)\log T
\end{aligned}$$

where (derived in Appendix B)

$$l_{\hat{\theta}}(y) = -\frac{MT}{2} - \frac{T}{2}\sum_{j=1}^{r}\log(l_j) - \frac{T(M-r)}{2}\log\hat{\sigma}^2$$

and $\dim(\theta)$ is the degree of freedom in θ, i.e., number of parameters that can be independently adjusted. It is given by

$$\dim(\theta) = Mr - \frac{r(r-1)}{2} + 1 + M.$$

By discarding terms that do not depend on r, and using the fact that the sum of the log of the eigenvalues is constant, the AIC criterion can be written as

$$\mathrm{AIC}_r = -2\log\left(\frac{\prod_{j=1}^{\rho} l_j^{1/(M-r)}}{\frac{1}{M-r}\sum_{r=j+1}^{\rho} l_j}\right)^{(M-r)T} + 2\dim(\theta)$$

where ρ is the number of non-zero eigenvalues. The BIC can of course be written similarly. Interestingly, this formulation reveals that these criteria only depend on the $\rho - r$ smallest eigenvalues.

Both AIC and BIC have been found to be useful in practice. BIC penalizes complex models more than AIC and therefore always selects fewer PCs. In addition, it can be shown that AIC is not a consistent estimator of the true order [77].

Hansen [62] derived a criterion which is equivalent to the AIC presented here and applied it to a fMRI data set [63]. Another notable paper that derives a method closely related to BIC is [130]. Surprisingly, many authors incorrectly use the AIC/BIC criteria presented here for other problems than nPCA. For instance, the papers [24, 12] used them for selection of the number of independent components in fMRI data; [165] for selection of the number of signals in array processing (that paper, however, acknowledged the incorrect usage).

1.4.3 Spatial nPCA

Since spatial nPCA is very similar to the temporal nPCA we will only state the model and give the ML solutions. There will also be some recycling of notation. The model is given by

$$y_{(v)} = m_v 1_T + G u_v + \epsilon_v, \quad v = 1, ..., M \tag{1.14}$$

where m_v is the baseline amplitude at voxel v, G is the $T \times r$ loading matrix, and $u_v \sim N(0, I_r)$ and $\epsilon_v \sim N(0, \sigma^2 I_T)$ are independent random vectors. The ML solutions are given by

$$\begin{aligned} \hat{m}_v &= \bar{y}_v = \frac{1}{T}\sum_{t=1}^{T} y_{v,t} \\ \hat{G} &= Q_r(L_r - \hat{\sigma}_r^2 I_r)^{1/2} R \\ \hat{\sigma}^2 &= \hat{\sigma}_r^2 = \frac{1}{T-r}\sum_{j=r+1}^{T} l_j. \end{aligned} \tag{1.15}$$

where $L_r = \text{diag}(l_1, ..., l_r)$, $l_1 > l_2 >, ..., > l_r$, contains the r largest eigenvalues of the data covariance matrix S_y, R is an arbitrary orthogonal rotation matrix, and

$$S_y = \frac{1}{M}\sum_{v=1}^{M}(y_{(v)} - \bar{y}_v)(y_{(v)} - \bar{y}_v)^T$$

where the r columns of Q_r are the eigenvectors, so $S_y Q_r = Q_r L_r$.

1.4.4 Deterministic PCA

There is another PCA model based on ML which is due to Whittle [167] that we now briefly discuss. This model was also considered in Anderson [5] where it was called linear functional relationships. In addition Anderson compared it to nPCA which he called structural linear relationships. The deterministic PCA model is given by

$$y_t = m + Gv_t + \epsilon_t, \quad t = 1, ..., T.$$

Now v_t is assumed to be deterministic and has to satisfy $\sum_{t=1}^{T} v_t = 0$. The log-likelihood is given by

$$l_\theta(y) = -\frac{TM}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{t=1}^{T} \|y_t - m - Gv_t\|^2.$$

The ML solutions are given by [5, 147]

$$\begin{aligned} \hat{G} &= P_r \\ \hat{v}_t &= P_r^T (y_t - \hat{m}), \quad t = 1, ..., T \\ \hat{\sigma}^2 &= \frac{1}{M} \sum_{j=r+1}^{\rho} l_j. \end{aligned}$$

Notice that if we assume that v_t has been determined the log-likelihood to be optimized is equivalent to Equation (1.3). This model has not been used as much in practice as nPCA, perhaps since theoretical analysis are more difficult, e.g., it is harder to derive asymptotic results.

CHAPTER II

TEMPORALLY SMOOTH AND SPATIALLY LOCAL nPCA MODEL

As discussed in Section 1.1 the observed voxel timeseries are often modeled as a sum of BOLD response, nuisance signals such as drift, and stationary noise. The drift term is supposed to account for the low-frequency noise, head movement and various other effects of physiological or hardware related origin. Since the drift term is supposed to account for such wide variety of effects, researchers have not agreed on a satisfactory model for it.

Probably the most common drift model is to use a linear combination of few low-frequency terms from the discrete cosine set [51] or polynomials [169]. Meyer [99] modeled the drift nonparametrically using wavelets within the framework of partial linear models, Fadili et al. [43] further extended this method. Bazargani et al. [11] suggested to model the drift nonparametrically using the so-called MDL denoising principle [124].

In the following, we propose two models that recognize temporal smoothness and allow for non-stationary noise model. The idea is to let the noise model, based on nPCA, take care off the nuisance/drift signals. The first model models the noise globally, while the second model relaxes the global assumption. Both models use regression matrix to model the mean, which is supposed to account for the BOLD response and the baseline. The nPCA model is used to model the noise part. In the first model a global nPCA model is used but in the second model local log-likelihood is used to relax that assumption. Part of this chapter has previously been published in our own conference papers [156, 157].

2.1 Temporally Smooth Global nPCA (XnPCA)

We propose a spatio-temporal model based on (spatial) nPCA which we call XnPCA. Its main properties are that it 1) models the BOLD response deterministically 2) allows for temporal smoothness 3) is able to handle non-stationary noise. The XnPCA model is given by

$$y_{t,v} = x_t^T \beta_v + g_t^T u_v + \epsilon_{t,v}, \quad t = 1, ..., T, v = 1, ..., M. \tag{2.1}$$

Here $X = [x_t^T]$ is a $T \times p$ regression matrix that includes the mean, and any standard model for the BOLD response, and $u_v \sim N(0, I_r)$, $\epsilon_v \sim N(0, \sigma^2 I_T)$ orthogonal random vectors, u_v and ϵ_v are mutually independent, and g_t^T is t-th row of the loading matrix G.

Now we impose temporal smoothness on the loadings, $g_t, t = 1, ..., T$. There are two main ways to do that. Firstly, by imposing a roughness penalty on the loadings. Secondly, by expanding the loadings in a smooth basis. We will focus on the second method. Lets assume $\Phi = [\phi_t^T]$ is a predetermined basis, e.g., Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) or B-splines. We expand the loadings in this basis

$$g_t = \phi_t^T B$$

where B is a $m \times r$ matrix of basis coefficients. By using this expression we can rewrite the XnPCA model in the following form

$$y_{(v)} = X\beta_v + \Phi B u_v + \epsilon_v, \quad v = 1, ..., M. \tag{2.2}$$

Now we proceed to show that this model does not assume (wide sense) stationary noise. An equivalent stochastic description of the Model (2.2) is given by

$$y_{(v)} \sim N(X\beta_v, \Omega)$$

where $\Omega = \Phi B B^T \Phi^T + \sigma^2 I_M$. Now notice that

$$\Omega_{t,s} = \phi_t^T B B^T \phi_s + \sigma^2.$$

But for the noise to be stationary we need Ω to be a Toeplitz matrix or equivalently a diagonal-constant matrix, which is clearly not the case.

The log-likelihood function for the XnPCA model is given by

$$\begin{aligned} l_\theta(y) &= -\frac{M}{2}\mathrm{tr}(S_e\Omega^{-1}) - \frac{M}{2}\log|\Omega| \\ &= -\frac{M}{2\sigma^2}\mathrm{tr}(S_e) + \frac{M}{2\sigma^2}\mathrm{tr}(W^{-1}B^T\Phi^T S_e \Phi B) \\ &- \frac{M(T-r)}{2}\log\sigma^2 - \frac{M}{2}\log|W| \end{aligned}$$

where $W = B^T B + \sigma^2 I_r$ and

$$S_e = \frac{1}{M}\sum_{v=1}^{M}(y_{(v)} - X\beta_v)(y_{(v)} - X\beta_v)^T.$$

2.1.1 ML Estimation for XnPCA

The task at hand is to estimate $\theta = (\beta, B, \sigma^2)$. Differentiating the likelihood and solving the Euler equations leads to

$$\hat{\beta}_v = (X^T\Omega^{-1}X)^{-1}X^T\Omega^{-1}y_{(v)}, \quad v = 1, ..., M \tag{2.3}$$

$$\hat{B} = K_r(D_r - \hat{\sigma}^2 I_r)^{1/2} R \tag{2.4}$$

$$\hat{\sigma}^2 = \frac{\mathrm{tr}(S_e) - \mathrm{tr}(D_r)}{T - r} \tag{2.5}$$

where K_r is the $m \times r$ matrix of unit eigenvectors of $S_\Phi = (\Phi^T\Phi)^{-1/2}\Phi^T S_e \Phi(\Phi^T\Phi)^{-1/2}$,$D_r = \text{diag}(d_1, ..., d_r)$, and R is an arbitrary rotation matrix. Now we make few observations. The matrix S_Φ is of rank less than or equal to m so $m \geq r$. In addition, since the maximal rank of S_e is $T - p$ we get the condition $r \leq T - p$. And finally, since the mean is always included in the regression matrix X it is not included in Φ so Φ is of maximal rank $m \leq T - 1$.

The Euler equations are linked nonlinearly together so the solution is not trivial. We suggest the following cyclic ascent algorithm to maximize the likelihood

Algorithm II.1 (XnPCA)**.** The XnPCA algorithm consists of the following steps, where subscript 0 means current iteration index, and subscript 1, means new iteration index.

1. Given B_0, σ_0^2 compute $l_{\theta_0}(y)$.

2. Compute $\beta_{v,1}$ via Equation (2.3).

3. Compute B_1 and σ_1^2 via Equations (2.4) and (2.5) (set $R = I_r$).

4. Compute $l_{\theta_1}(y)$.

5. If $\frac{l_{\theta_1}(y) - l_{\theta_0}(y)}{l_{\theta_0}(y)} < k$ stop, else set $\theta_0 = \theta_1$ and return to step 1.

We can use the result of the algorithm to estimate the voxel time series at voxel v

$$\hat{y}_{(v)} = X\hat{\beta}_v + \Phi\hat{B}\hat{u}_v \tag{2.6}$$

where the sample nPCs and their corresponding covariance are given by

$$\begin{aligned} \hat{u}_v &= \hat{W}^{-1}\hat{B}^T\Phi^T(y_{(v)} - X\hat{\beta}) \\ S_{\hat{u}} &= \hat{\sigma}^2\hat{W}^{-1}. \end{aligned}$$

It can be shown that this cyclic descent algorithm is globally convergent. We refer to Section 3.5.1 for a discussion.

2.1.2 Model Selection for XnPCA

To finish the model fitting we need to choose the number of basis functions m and the number of PCs r. Here we present the BIC criterion, which was discussed in Section 1.4.2. However, in this case the usage is unusual because we need to choose two tuning parameters, i.e., the number of basis functions m, and the number of principal components r. The BIC is a function of both tuning parameters and is given by

$$\mathrm{BIC}(m,r) = \frac{MT}{2} + \frac{M(T-r)}{2}\log\hat{\sigma}^2 + \frac{M}{2}\sum_{j=1}^{r}\log d_j + \frac{1}{2}\mathrm{dim}(\theta)\log M \qquad (2.7)$$

where $\hat{\theta}$ is the maximum likelihood estimate of the parameter vector θ, and $\mathrm{dim}(\theta)$ is the number of free parameters given by

$$\mathrm{dim}(\theta) = mr - r(r-1)/2 + r + 1.$$

To select the tuning parameters m and r we look for a minimum of the BIC function. We find it useful to be able to decompose BIC spatially in order to examine the local fit of the voxel timeseries. The BIC can be spatially decomposed in the following way

$$\begin{aligned}\mathrm{BIC}(m,r) &= \sum_{v=1}^{M}\mathrm{BIC}_v(m,r)\\ &= \frac{1}{2}\sum_{v=1}^{M}\Bigg(\frac{1}{\hat{\sigma}^2}(y_{(v)} - X\hat{\beta}_v)^T(I_T - \Phi\hat{B}\hat{W}^{-1}\hat{B}^T\Phi^T)(y_{(v)} - X\hat{\beta}_v)\\ &+ \quad (T-r)\log\hat{\sigma}^2 + \log|\hat{W}|\Bigg) + \frac{1}{2}\,\mathrm{dim}(\theta)\log M. \qquad (2.8)\end{aligned}$$

2.1.3 Likelihood Ratio Test (LRT)

To compare, for instance, two nPCA based models the ML framework provides a nice way to do it, namely LRT [15]. Lets assume that H0: $y_{(v)} \sim N(m, GG^T + \sigma^2 I_T)$ and H1: $y_{(v)} \sim N(m, G_1 G_1^T + \sigma_1^2 I_T)$ represent two nPCA based models, respectively. Then the LRT can be written as

$$\mathrm{LRT}(\theta, \theta_1) \;\; = \;\; -\frac{1}{2}\mathrm{det}|\Omega| - \frac{1}{2}\mathrm{tr}(S_y \Omega^{-1}) + \frac{1}{2}\mathrm{det}|\Omega_1| + \frac{1}{2}\mathrm{tr}(S_y \Omega^{-1}).$$

Just like Friston's [55] CCA test statistic this LRT statistic can in current form only be used to make inference about whole brain volumes, i.e., construction of activation maps is not possible. However, a very important observation is that the LRT can be decomposed spatially.

Spatial Decomposition of the Log-Likelihood:

$$\mathrm{LRT} = \sum_{v=1}^{M} \mathrm{LRT}_v$$

where

$$\mathrm{LRT}_v = \frac{T}{2\sigma^2} \sum_{j=1}^{r} \frac{(\hat{g}_{(j)}^T (y_{(v)} - m))^2}{g_{(j)}^T g_{(j)} + \sigma^2} - \frac{T}{2\sigma_1^2} \sum_{j=1}^{r} \frac{(g_{(j),1}^T (y_{(v)} - m))^2}{g_{(j),1}^T g_{(j),1} + \sigma_1^2}.$$

The LRT can then be plotted spatially and tested.

2.2 Example: Temporally Smooth Global nPCA

Now we analyze the AFNI fMRI data set presented in Section 4.4.1. We focus on brain slice number 5 which is known to include the motor cortex. Only voxels inside the brain were considered. In addition, each voxel timeseries was normalized to unit variance. For the BOLD response we assume a simple model due to Cohen

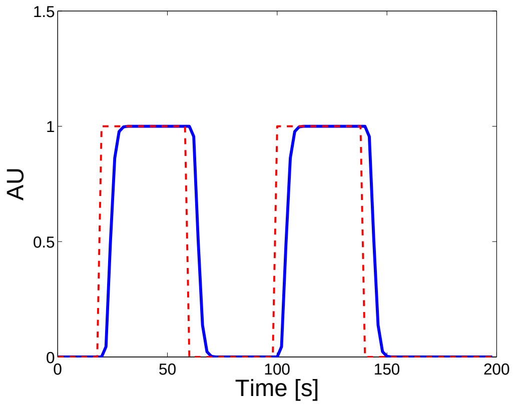
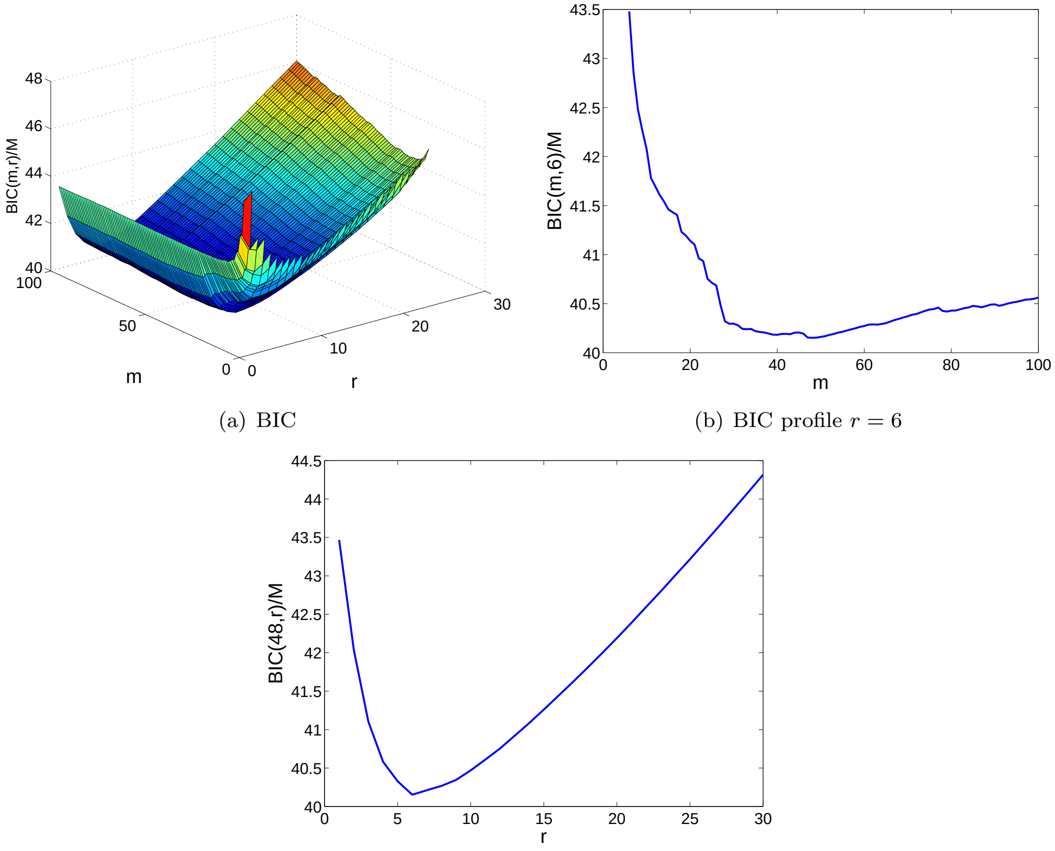


Figure 2.1: The stimulus signal used in the example in Section 2.2 (dashed) and its corresponding BOLD response. Stimulus equal to one means right hand finger thumb opposition, and stimulus equal to zero means rest.

[27]. In this model it is assumed that the brain has fixed impulse response, so the the BOLD response is a convolution between it and the right hand finger-thumb opposition stimulus signal. Figure 2.1 shows the BOLD response (solid line) and the corresponding stimulus signal (dashed line). The regression matrix $X = [1_T, s]$ contains two columns; the BOLD response s that we just discussed, and a constant signal 1_T for the baseline. We chose the real harmonic Fourier basis for Φ which is given by a $T \times T - 1$ matrix

$$\Phi = [\sin(\frac{2\pi}{T}t), \cos(\frac{2\pi}{T}t), ..., \sin(k\frac{2\pi}{T}t), \cos(k\frac{2\pi}{T}t), ..., \cos(\frac{T-1}{T}\pi t)], \quad t = 1, ..., T.$$

Figure 2.2 shows the BIC plots for this example; (a) shows the two dimensional BIC surface, (b) the minimum BIC profile for fixed r, (c) the minimum BIC profile for fixed m. The BIC picked $r = 6$ principal components and $m = 48$ basis functions which corresponded to the overall minimum of the BIC. Note that the BIC along the $r = 6$ profile is rather flat between $m = 25$ and $m = 50$ so an alternative is to pick $m = 25$. Figure 2.3 shows plots of XnPC loadings or the eigentimeseries, i.e., the columns of the ΦB matrix, and Figure 2.4 shows a spatial plots of sample XnPCs or the eigenimages, i.e. $u_{v,j}, v = 1, ..., M, j = 1, ..., 6$. We see from the eigentimeseries



(a) BIC

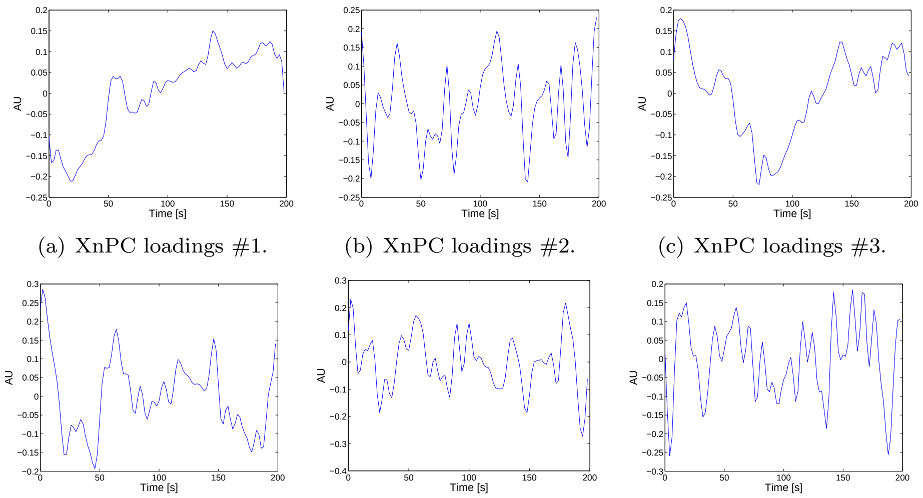
(b) BIC profile $r = 6$

(c) BIC profile $m = 48$

Figure 2.2: The BIC plots for the example in Section 2.2.

that they are relatively low frequency signals. The first eigentimeseries is a drift signal that increases with time, a phenomenon usually seen in fMRI timeseries. Eigenimage 2 is clearly strongly representing the saggital sinus. But perhaps the most important takeaway point is that these eigentimeseries are probably not something that one would a priori put in a regression matrix when doing standard univoxel analysis.

Figure 2.5 shows BIC plotted spatially using Equation (2.8). This plot gives us interesting local information of the quality of fit. Specifically, the motor areas have lowest BIC scores, and the white matter the highest. Interestingly, the voxels near the edges of the brain and in the saggital sinus (see Figure 2.6) also seem to be represented well. Figure 2.7 shows voxel timeseries along with their XnPCA fits



(a) XnPC loadings #1. (b) XnPC loadings #2. (c) XnPC loadings #3.

(d) XnPC loadings #4. (e) XnPC loadings #5. (f) XnPC loadings #6.

Figure 2.3: Plots of the XnPC loadings (the columns of the ΦB matrix) for the example in Section 2.2.

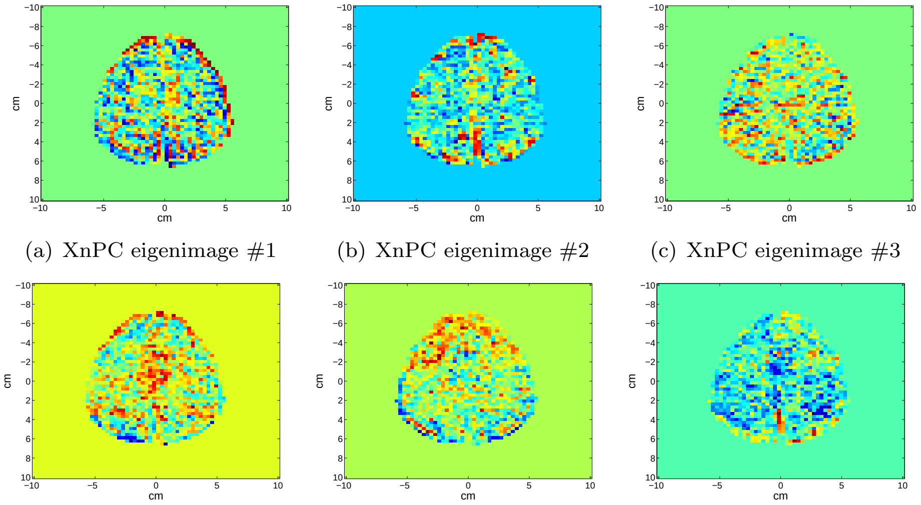
(using Equation (2.6)) for four brain regions defined on Figure 2.6: MC for motor cortex, SS for saggital sinus, EB for edge of brain, and WM for white matter. The observed voxel timeseries are fitted very nicely, it seems that we are getting nice tradeoff between bias and variance. Notice that the method does not appear to try to fit the white matter voxel timeseries which is not surprising since we expect it only to contain white noise.

2.2.1 LRT

Consider the hypothesis tests

$$\begin{aligned} H_0 : \beta_{2,v} &= 0 \\ H_1 : \beta_{2,v} &\neq 0 \end{aligned}$$

where $\beta_{2,v}$ is the parameter that controls the amplitude of the BOLD response signal s (see Figure 2.1) at voxel number v. The spatial decomposition of the LRT for this



(a) XnPC eigenimage #1 (b) XnPC eigenimage #2 (c) XnPC eigenimage #3

(d) XnPC eigenimage #4 (e) XnPC eigenimage #5 (f) XnPC eigenimage #6

Figure 2.4: Spatial plots of the XnPC eigenimages (the $u_{v,j}$) for the example in Section 2.2.

hypothesis test is given by

$$\begin{aligned} \text{LRT} &= \sum_{v=1}^{M} \text{LRT}_v \\ &= \frac{1}{2}\sum_{v=1}^{M} \beta_{v,2}^2 (s^T \hat{\Omega}^{-1} s) \\ &= \frac{1}{2}\sum_{v=1}^{M} \beta_{v,2}^2 s^T \frac{(I_T - \Phi\hat{B}\hat{W}^{-1}\hat{B}^T\Phi^T)}{\hat{\sigma}^2} s. \end{aligned} \tag{2.9}$$

The Ω was only estimated under H_1. Figure 2.8 shows a spatial plot of $2 \cdot \text{LRT}$ along with a standard univoxel activation plot (see Equation (1.2)). We see that the XnPCA LRT test statistic reflects activation both in the primary and supplementary motor cortices. It is slightly surprising that we seem to get activation in the both sides of the brain since this was a right hand-finger tapping experiment.

2.3 Temporally Smooth and Spatially Local nPCA (XlnPCA)

In the XnPCA model described above the noise model is global, i.e., the covariance matrix is the same for all voxels. Here we propose to relax that by developing a spatial local approach by using the local log-likelihood [151]. For another example of

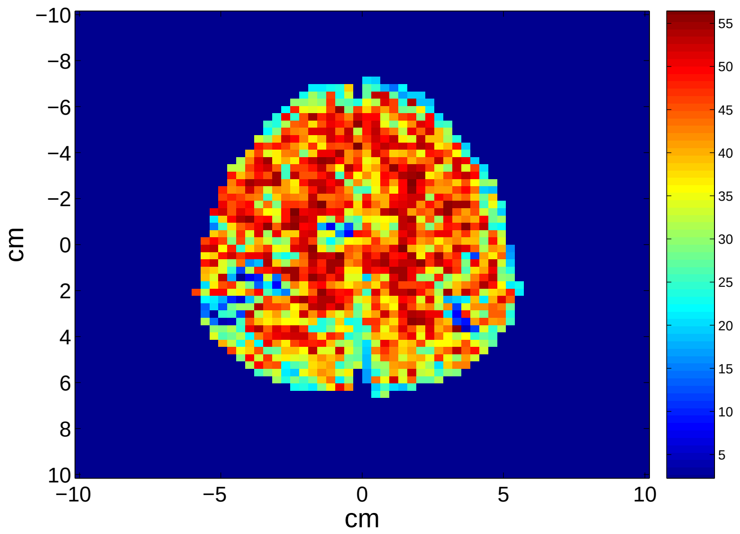


Figure 2.5: The BIC statistic plotted spatially for the example in Section 2.2.

an application of local log-likelihood in fMRI see [136]. In this case the model which we call XlPCA for voxel timeseries is given by

$$y_{(v)} = X\beta_v + \Phi_v B_v u_v + \epsilon_v, \quad v = 1, ..., M. \tag{2.10}$$

The difference from the model in previous section is that Φ_v, B_v and σ_v^2 and consequently the covariance matrix are allowed to vary spatially. Notice that the same basis Φ_v is used for all voxels, but the number of basis function can change.

2.3.1 Estimation for XlnPCA

The estimation is based on optimizing the local log-likelihood which is given by

$$l_{\theta_v}(y) = \sum_u K_{u,v}\left(-\frac{1}{2}(y_{(u)} - X\beta_v)^T\Omega_v^{-1}(y_{(u)} - X\beta_v) - \frac{1}{2}\log|\Omega_v|\right)$$

where $K_{u,v}$ is a spatial weighting kernel centered at v that sums to one. In this work we pick uniform weighting

$$K_{u,v} = \frac{1}{k^2}I(y_{(u)} \in L_v)$$

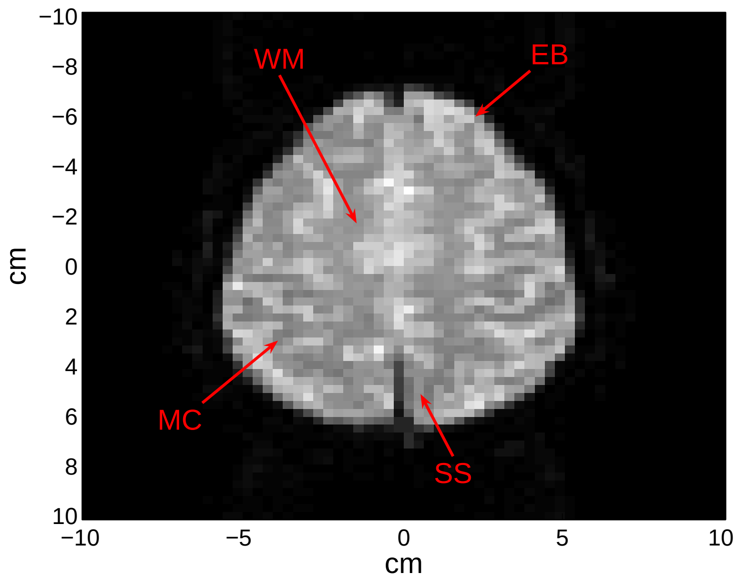


Figure 2.6: The brain regions used in the example in Section 2.2 defined.

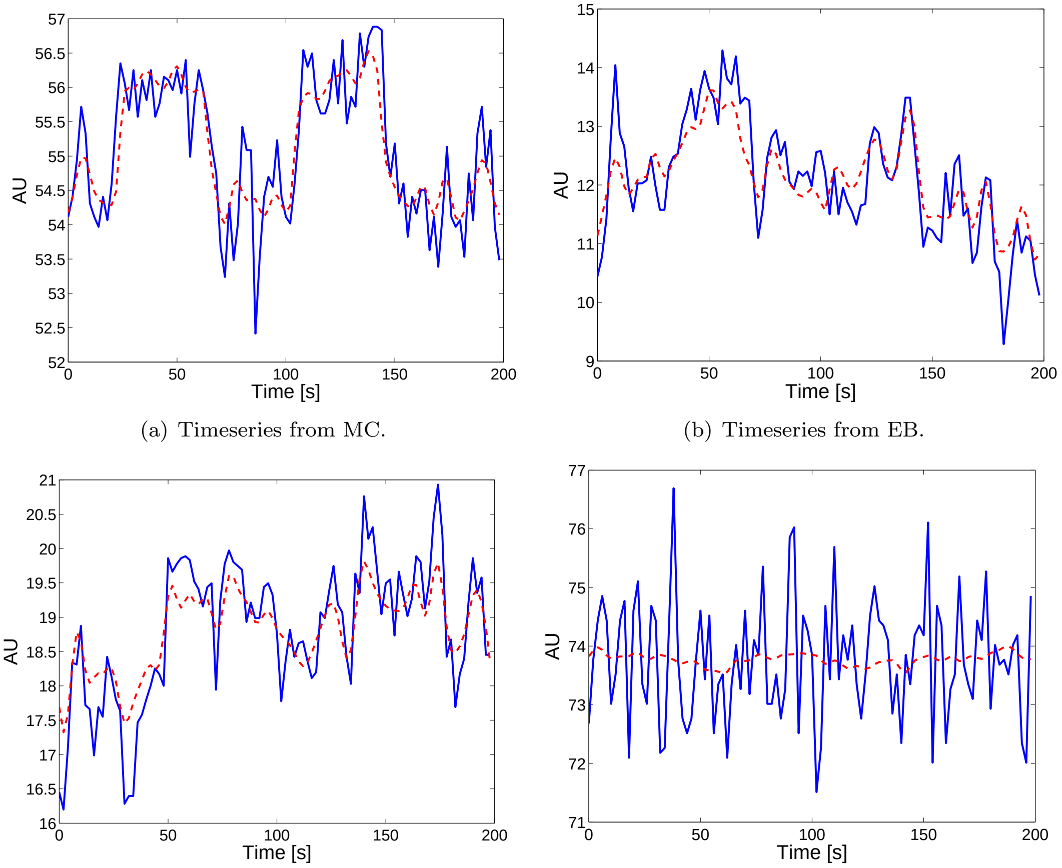
where L_v is a $k_v \times k_v$ window centered at voxel v. In this case the local log-likelihood can be written as

$$l_{\theta_v}(y) = -\frac{k^2}{2}\mathrm{tr}(\Omega_v^{-1}S_{e,v}) - \frac{k^2}{2}\log|\Omega_v|$$

where $S_{e,v} = \frac{1}{k^2}\sum_{u\in L_v}(y_{(u)} - X\beta_v)(y_{(u)} - X\beta_v)^T$ is the sample local covariance and $\theta_v = (\beta_v, B_v, \sigma_v^2)$ is a vector of the local parameters to be estimated. The local log-likelihood is maximized when

$$\begin{aligned}\hat{\beta}_v &= (X^T\Omega_v^{-1}X)^{-1}X^T\Omega_v^{-1}\bar{y}_v \\ \hat{B}_v &= K_{r,v}(D_{r,v} - \hat{\sigma}_v^2 I_r)^{1/2}R \\ \hat{\sigma}_v^2 &= \frac{\mathrm{tr}(S_{e,v}) - \mathrm{tr}(D_r)}{T-r}\end{aligned}$$

where $K_{r,v}$ is the $m_v \times r_v$ matrix of unit eigenvectors of $S_{\Phi,v} = (\Phi_v^T\Phi_v)^{-1/2}\Phi_v^T S_{e,v}\Phi_v(\Phi_v^T\Phi)^{-1/2}$, $D_{r,v}$ is a $r_v \times r_v$ diagonal matrix that contains the corresponding eigenvalues, R is an arbitrary rotation matrix, and $\bar{y}_v$ is a average of voxel timeseries over the local window L_v. As in the previous section the parameters are intertwined so we use cyclic ascent to find their values.



(a) Timeseries from MC.

(b) Timeseries from EB.

(c) Timeseries from SS.

(d) Timeseries from WM.

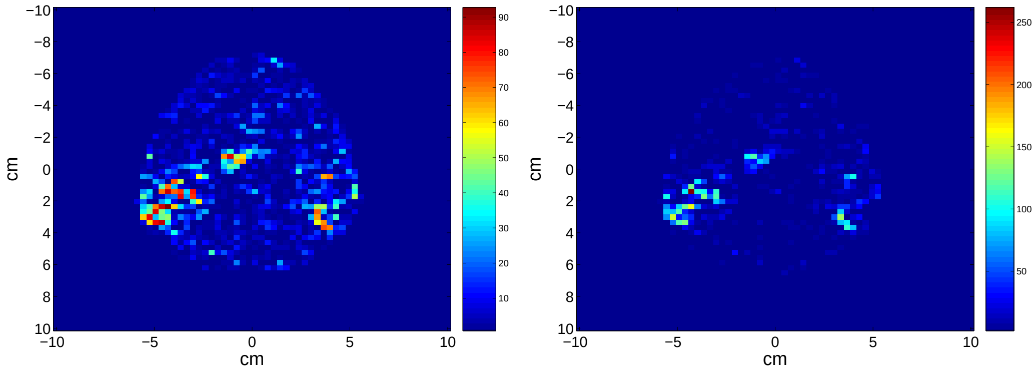
Figure 2.7: Voxel timeseries vs fitted timeseries for XnPCA for four different brain regions defined on Figure 2.6 for the example in Section 2.2.

2.3.2 Model Selection for XlnPCA

As before we use the BIC criterion to complete the model fitting. In this case we are selecting three tuning parameters for each voxel; number of local PC r_v, the number of basis functions m_v, and the size of the ROI k_v.

$$\mathrm{BIC}_v(m_v, r_v) = -l_{\hat{\theta}_v}(y) + \dim(\theta)\log k_v$$

where $\hat{\theta}$ is the MLE of the parameter vector θ, and $\dim(\theta)$ is the number of free parameters given by $\dim(\theta) = m_v r_v - r_v(r_v - 1)/2 + r_v + 1$. Notice that when $r_v = 0$



(a) The XnPCA test statistic

(b) A standard univoxel test statistic

Figure 2.8: The XnPCA LRT statistic plotted spatially for the example in Section 2.2 along with a standard univoxel LRT statistic.

the model (2.10) is simply

$$y_{(v)} = X\beta_v + \epsilon_v.$$

That is the standard (white noise) univoxel model.

2.4 Example: Temporally Smooth and Spatially Local nPCA

Here we use the XlnPCA model for the AFNI data, we use the same regression matrix X, and the same basis of harmonic Fourier basis functions Φ as in the example for XnPCA. However, in this case we do not normalize the voxels to unit variance. Since the model selection is computationally demanding the activation signal was regressed out from the data prior to model selection. This simplification cuts out the cyclic descent step in the estimation algorithm. Empirical evidence suggests that this does not change the final model in a significant way. Figure 2.9 is a spatial plot of the size of ROI $k_v, v = 1, ..., M$ chosen by the BIC criterion over the set $k_v = 1, 3, 5, 7, 9$. This reflects automatic smoothing since larger ROI size implies that the method is pooling more voxels together to estimate parameters at a particular voxel. Figure 2.10 depicts a spatial map of the number of basis functions $m_v, v = 1, ..., M$. Interestingly, there seems to be that voxels near the edges of the brain need the

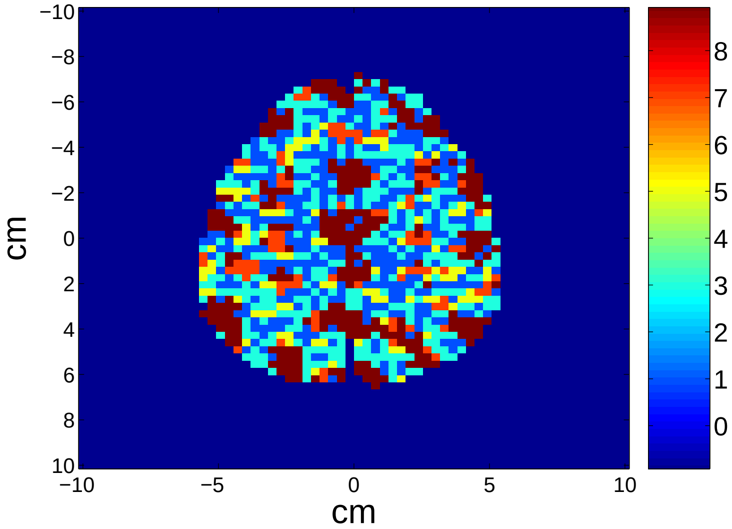


Figure 2.9: The size of ROI $k_v, v = 1, ..., M$ plotted spatially for the example in Section 2.4.

largest number of basis functions. Figure 2.11 shows a spatial map of the number of local PCs $r_v, v = 1, ..., M$. For most voxels the number of lPCs is from 2-6. Figure 2.12 shows a spatial map of the BIC statistic.

Figure 2.13 shows a spatial plot of the following LRT test statistic constructed similarly to (2.9)

$$\mathrm{LRT}_v = \frac{\hat{\beta}_{2,v}^T (X_2^T (I_T - \Phi_v \hat{B}_v \hat{W}^{-1} \hat{B}_v^T \Phi^T) X_2 \hat{\beta}_{2,v}}{\hat{\sigma}_v^2}.$$

This test statistic is much smoother that the one depicted on Figure 2.8. The activation in the motor cortices is still clearly apparent.

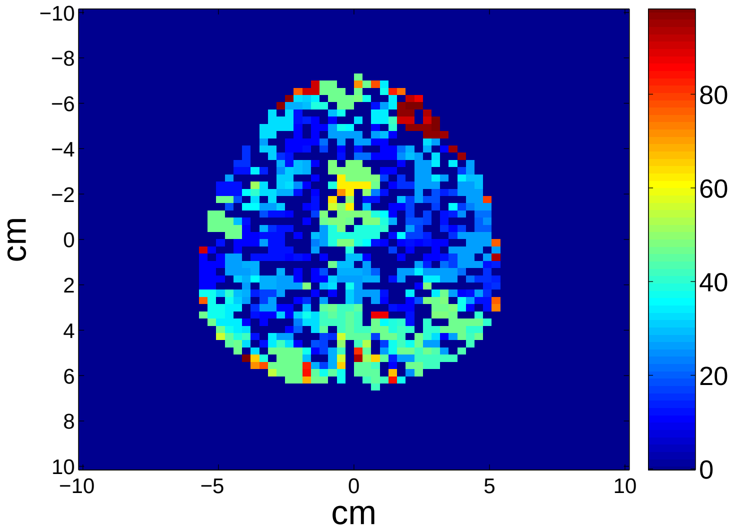


Figure 2.10: The number of basis functions $m_v, v = 1, ..., M$ plotted spatially for the example in Section 2.4.

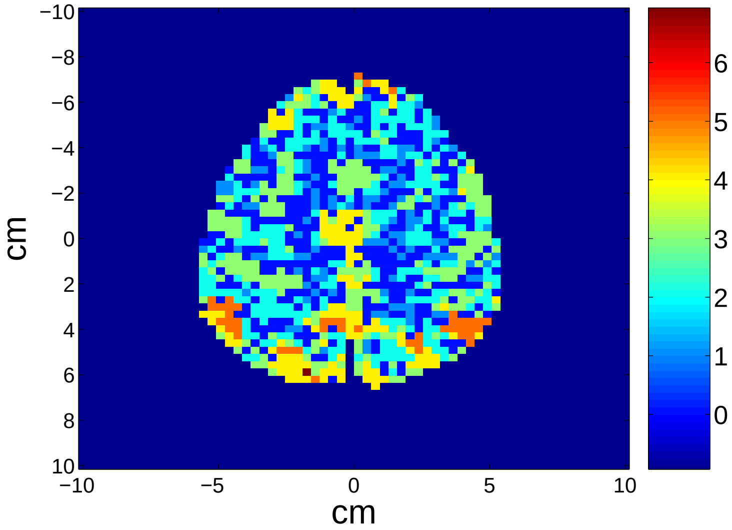


Figure 2.11: The number of local PCs $r_v, v = 1, ..., M$ plotted spatially for the example in Section 2.4.

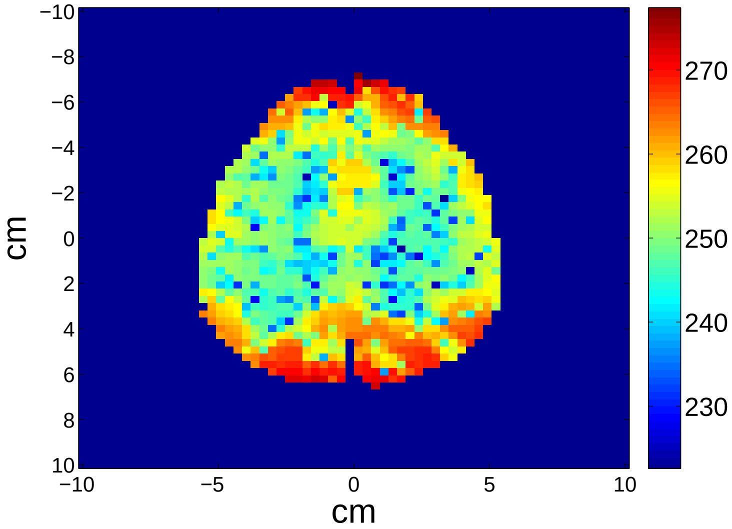


Figure 2.12: The BIC statistic plotted spatially for the example in Section 2.4.

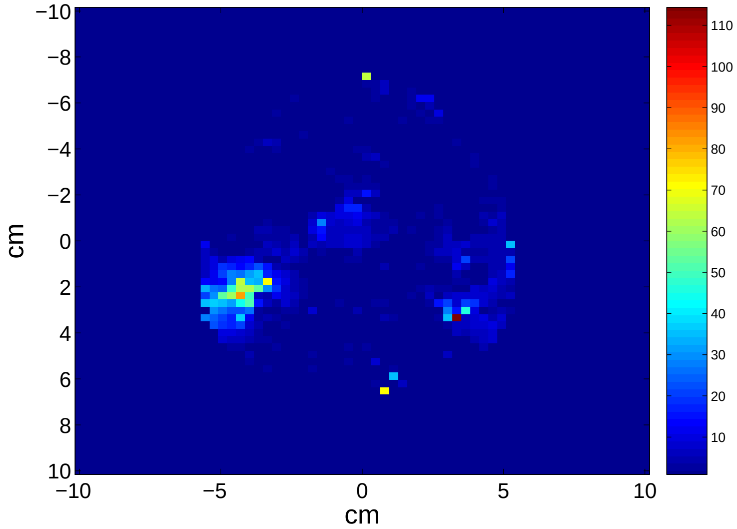


Figure 2.13: The activation test statistic plotted spatially for the example in Section 2.4.

CHAPTER III

SPARSE VARIABLE nPCA USING GEODESIC OPTIMIZATION METHODS

The nPCs are a linear combination of all the variables of the data set and the loadings are usually nonzero. In many application, e.g., nPCA of imaging data sets where pixels are regarded as variables, some of the variables measured are just noise. It is then attractive to zero them out to get less noisy and more interpretable results.

A desirable way is to incorporate automatic thresholding into the estimation process. This kind of automatic variable selection has been a very active research topic in statistics and signal processing over the last decade in the form of penalized least squares optimization where the penalty is nonlinear [2, 35, 150, 154, 172]. Especially interesting is the paper by Alliney et al. [2] where they lay out the theory for this kind of penalized optimization and use their methods to fit AR and ARMA models. Basically, they anticipated the LASSO method, discussed below, by two years.

3.1 Sparse PCA

Sparse PCA (sPCA) is a newly introduced class of methods to get automatic thresholding in the PCA context. It aims to zero out some of the loadings on some of the variables. We identify two sub-classes; Sparse Loading PCA (slPCA) that aims to zero out only some of the PC loadings associated with a variable, and Sparse

Variable PCA (svPCA) that zeros out whole variables, i.e., zeros out all loadings associated with a variable. To clarify this terminology, we look at traditional PCA

$$Y \approx \sum_{j=1}^{r} l_j^{1/2} q_{(j)} p_{(j)}^T.$$

In the usual PCA terminology the rows of Y are called observations and the columns are called variables, of course in the fMRI case, $q_{(j)}$ is eigentimeseries number j, and $p_{(j)}$ is eigenimage number j. We can extract a vector of variables

$$y_{(v)} \approx \sum_{j=1}^{r} l_j^{1/2} q_{(j)} p_{v,j}$$

the $p_{v,j}, j = 1, ..., r$ are called the loadings for the variable v. If all the loadings $p_{v,j}, j = 1, ..., r$ are zeroed out then the whole vector $y_{(v)}$ is in effect zeroed out. We call this svPCA. If only some of the loadings $p_{v,j}, j = 1, ..., r$ are zeroed out we call it slPCA.

Perhaps the first explicit slPCA method was Simplified Component Technique LASSO (SCotLASS) introduced in Jolliffe et al. [75]. Later, Zou et al. [175] formulated slPCA as a regression-type optimization problem and obtained sparse loadings by using the Least Absolute Shrinkage and Selection Operator (LASSO) [150]. D'Aspremont [31] presented a direct formulation for slPCA using semidefinite programming. Johnstone et al. [72] proposed an svPCA method. In detail, his algorithm first wavelet transforms the data. Then discards low variance variables and computes reduced PCA based on the variables left over. Finally, we note that there is considerable literature on variable selection methods for classical PCA [74], which can be regarded as svPCA methods. All of the above mentioned methods have in common that they are not based on a statistical model. Therefore, they do not have access to the range of modeling and inferential tools that model based methods provide.

In this chapter, we propose a new svPCA algorithm, which we call svnPCA. Our method has several novel features. Firstly, it is based on a statistical model. Secondly, it uses an amplitude penalized likelihood formulation that is able to zero out variables rather than just loadings. Thirdly, the optimization problem is nonstandard because it involves an orthogonality constraint and we resolve this by using geodesic descent algorithms. Fourthly, to select the number of svnPCs, and the sparseness tuning parameter, we propose to use a novel form of the BIC criterion. Finally, we discuss an alternative svnPCA based on the EM algorithm. An earlier version of some of this work has previously been published in the conference paper [158] but many details were omitted. A journal paper has been submitted [160].

3.1.1 The LASSO and The Elastic Net

The LASSO [150] was proposed as an estimation method for the standard regression model (1.2). The LASSO estimate is given by

$$\beta_{lasso} = \underset{\beta \in R^p}{\operatorname{argmin}} \sum_{t=1}^{T} (y_t - \beta_1 - \sum_{j=2}^{p} x_{tj}\beta_j)^2 + h_1 \sum_{j=2}^{p} |\beta_j|.$$

Notice that if we square the penalty term this LASSO reduces to ridge regression [64]. Although LASSO is a shrinkage method like ridge regression there is an important difference. By setting the tuning parameter h_1 high enough LASSO can produce an exact zero solution. The LASSO solution can be efficiently computed by the Least Angle Regression (LARS) algorithm [39].

Zou et al. [174] pointed out the following limitation for LASSO: a) When $p > T$, the LASSO selects at most T variables. b) If there is a group of variables among which the pairwise correlation is very high, then the LASSO tends to select only one variable from the group. c) For $T > p$ situations, if there are high correlations between predictors, the prediction performance of the LASSO is dominated by ridge regression. Zou proposed the elastic net estimate as a remedy for these problem.

The elastic net estimate is given by:

$$\beta_{en} = (1 + h_2) \left\{ \underset{\beta \in R^p}{\operatorname{argmin}} \sum_{t=1}^{T} (y_t - \beta_1 - \sum_{j=2}^{p} x_{tj}\beta_j)^2 + h_2 \sum_{j=2}^{p} |\beta_j|^2 + h_1 \sum_{j=2}^{p} |\beta_j| \right\}.$$

Zou showed that the LARS algorithm can be used to compute the elastic net estimates efficiently.

3.1.2 Zou's slPCA

Zou's slPCA is based on solving the following optimization problem for $M \times r$ matrices $B = [b_{(1)}, ..., b_{(r)}]$ and $C = [c_{(1)}, ..., c_{(r)}]$:

$$\min_{B,C \in R^{M \times r}} \sum_{t=1}^{T} \|y_t - CB^T y_t\|^2 + h_2 \sum_{j=1}^{r} \|b_{(j)}\|^2 + \sum_{j=1}^{r} h_{1,j} \|b_{(j)}\|_1 \qquad (3.1)$$

subject to $C^T C = I_r$ where h_2 and $h_{1,j}$ are tuning parameters that need to be selected. Notice that if $h_1 = 0$ and $h_{2,j} = 0$ then (3.1) reduces to the traditional PCA cost function (1.3).

The estimation algorithm Zou proposes is a cyclic descent algorithm that alternates between optimizing (3.1) with respect to B and C.

Algorithm III.1 (Zou's slPCA)**.** Zou's algorithm is given by, where e.g., C_0 means estimate of C at current iteration, and C_1 means new estimate of C.

1. At the very first step set $C_0 = P_r$, where $Y = QL^{1/2}P^T$.

2. Optimization with respect to B keeping C fixed leads to the following elastic net problems [174] for $j = 1, ..., r$:

$$b_{(j),1} = \underset{b_{(j)} \in R^{M \times 1}}{\operatorname{argmin}} b_{(j)}^T (Y^T Y + h_2) b_{(j)} - 2c_{(j),0}^T Y^T Y b_{(j)} + h_{1,j} \|b_{(j)}\|_1.$$

3. Optimization with respect to C keeping B fixed is equivalent to solving a reduced rank Procrutes rotation [94]. To solve we do the SVD of $Y^T Y B_1 = UDV^T$, and then update $C_1 = UV^T$.

4. Repeat steps 2-3 until B converges.

5. Normalization:

$$\hat{v}_{(j)} = \frac{b_{(j)}}{\|b_{(j)}\|}, \quad j = 1, ..., r.$$

The $\hat{v}_{(j)}$ are the sparse loadings.

An important problem is how to choose the tuning parameters. Zou picked $h_{1,j}$ which gave a good compromise between variance and sparsity. He did this by using plots of percentage of explained variance of the sparse loadings as a guideline. However, the question of exactly what is a good compromise was left unanswered. The choice of h_2 was not considered as critical as its main purpose is to overcome potential collinearity problems of Y.

When $M >> T$ the computational cost of the above algorithm is very high. A computationally more efficient algorithm is obtained by picking $h = \infty$. Then step 2 of the above algorithm reduces to soft-thresholding

$$b_{(j),1} = \left(|c^T_{(j),0} Y^T Y| - \frac{h_{1,j}}{2} \right)_+ \operatorname{sgn}(c^T_{(j),0} Y^T Y).$$

3.1.3 Jolliffe's SCOTLASS slPCA

Jolliffe's SCotLASS [75] is based on the following optimization problem

$$\underset{a_{(i)} \in R^{M \times 1}}{\operatorname{argmax}} \; a^T_{(i)} S_y a_{(i)} \tag{3.2}$$

subject to

$$\begin{aligned} a^T_{(i)} a_{(i)} &= 1, \quad a^T_{(j)} a_{(i)} = 0, j < i, i \geq 2 \\ \sum_{k=1}^{M} |a_{ik}| &\leq h_1 \quad . \end{aligned}$$

In Jolliffe's paper the problem was stated in terms of the correlation matrix instead of the covariance matrix, but that does not change the algorithm. Notice that if we drop the l_1 constraint on the loadings the cost function reduces to the traditional PCA cost function (1.5).

The tuning parameter h_1 has to be selected by the user. Jolliffe does not give a method to select it, but notes the following properties 1) $h_1 \geq \sqrt{M}$ yields traditional PCA; 2) $h_1 < 1$, there is no solution; 3) $h_1 = 1$, there is exactly one nonzero a_{ik} for each i. SCotLASS is not a convex optimization problem so it needs numerical optimization and suffers from the problem of many local optima. In [75] a projected gradient algorithm which required multiple runs from random initial starting points to get globally optimal solution. Trendafilov [153] et al. later developed a globally convergent algorithm. Despite of that, the lack of guidance for choosing h_1 makes SCotLASS an impractical solution.

3.1.4 Johnstone's svPCA

Johnstone's sparse variable PCA algorithm can be described by the following steps:

1. Select a wavelet basis $\{\psi_{(i)}, i = 1, ..., M\}$ for $R^{M\times 1}$, compute coordinate $\tilde{y}_{t,i}$ for each y_t such that
$$y_{t,v} = \sum_{i=1}^{M} \tilde{y}_{t,i}\psi_{v,i}.$$

2. Compute the sample variances $\hat{\sigma}_i^2 = v\hat{a}r(\tilde{y}_{t,i})$. Let $\hat{I}$ denote the set of indices i corresponding to the largest k variances.

3. Apply traditional PCA to the reduced data set $\{\tilde{y}_{t,i}, i \in \hat{I}, t = 1, ..., T\}$ obtaining $\tilde{\rho} = [\tilde{\rho}_{i,j}], \quad i = 1, ..., M, j = 1, ..., k.$

4. Filter out noise in $\tilde{\rho}_i$ by hard thresholding yielding $\tilde{\rho}_i^*$.

5. Reconstruct: $\hat{\rho}_{v,j} = \sum_{i=1}^{k} \tilde{\rho}^*_{i,j}\psi_{v,i}$.

$\tilde{\rho}_{(j)}, j = 1, ..., k$ are the sparse loadings.

3.2 Sparse Variable Noisy PCA Formulation

In this section we introduce our new svnPCA method. We change the model slightly from what was presented in Section 1.4.1

$$y_t = m + Fu_t + \epsilon_t, \quad t = 1, ..., T. \tag{3.3}$$

In this case, for reasons that will become clear below, we constrain F to be orthonormal i.e., $F^T F = I_r$ and assume that $u_t \sim N(0, \Lambda)$, where $\Lambda = \text{diag}(\lambda_1, ..., \lambda_r)$. In this case, the log-likelihood is given by

$$\begin{aligned} l_\theta(y) &= -\frac{1}{2\sigma^2}\text{tr}(S_y) + \frac{1}{2\sigma^2}\text{tr}(W^{-1}F^T S_y F) \\ &- \frac{(M-r)}{2}\log\sigma^2 - \frac{1}{2}\log|W| - \frac{1}{2}\log|\Lambda| \end{aligned}$$

where $W = I_r + \sigma^2\Lambda^{-1}$. The MLE are in this case

$$\begin{aligned} \tilde{m} &= \frac{1}{T}\sum_{t=1}^{T} y_t \\ \tilde{F} &= P_r \\ \tilde{\sigma}^2 &= \frac{\text{tr}(S_y) - \text{tr}(L_r)}{M-r} \\ \tilde{\Lambda} &= L_r - \sigma^2 I_r. \end{aligned} \tag{3.4}$$

Our svnPCA procedure is based on minimizing the following amplitude penalized negative log-likelihood

$$J_\theta(y) = -\frac{1}{M}l_\theta(y) + h\rho(F)$$

where F is orthonormal. We consider initially the class of $l_p, p \geq 1$ penalties

$$\begin{aligned}\rho(F) &= \frac{1}{M}\sum_{v=1}^{M}\left(\sum_{u=1}^{r}|f_{v,u}|^p\right)^{1/p} \\ &= \frac{1}{M}\sum_{v=1}^{M}\|f_v\|_p.\end{aligned}$$

We think of this penalty and the log-likelihood as obtained by discretizing an integral; this explains the normalization by $1/M$. Notice that when $r = 1$ the penalty reduces to the l_1 penalty which is well known to produce a sparse solution [34]. Also note that for $p = 2, r \geq 1$ this type of penalty is well known in the total variation denoising literature [126, 164]. But there it is used to regularize a gradient, whereas we are regularizing amplitude. We do not suppose any spatial continuity with respect to v.

To clarify what we mean by a sparse solution we rewrite the nPCA model in the following way

$$y_{t,v} = m_v + f_v^T u_t + \epsilon_{t,v}, \quad t = 1, ..., T, v = 1, ..., M.$$

By concatenating the values at a voxel v in a vector we can write

$$y_{(v)} = m_v 1_T + U f_v + \epsilon_v, \quad v = 1, ..., M$$

where $U = [u_t^T]$ is a $T \times r$ matrix. From this expression we see that if the loadings corresponding to variable v are zero, i.e., $f_v = 0$ the variable $y_{(v)}$ can be dismissed as only noise.

3.2.1 Properties of the l_p Penalty

The properties of the penalty stem from the behavior of its gradient. We can investigating this by calculating the derivative

$$\frac{\partial\|f_v\|_p}{\partial f_{v,u}} = \begin{cases} \frac{\text{sgn}(f_{v,u})|f_{v,u}|^{p-1}}{\left(\sum_{u=1}^{r}|f_{v,u}|^p\right)^{1-p}}, & p > 1 \\ \text{sgn}(f_{v,u}), & p = 1. \end{cases}$$

We thus see that, when $p = 1$, the derivative is discontinuous with respect to each component $f_{v,1}, ..., f_{v,r}$ separately. This means that the l_1 penalty is able to produce zeroing of individual loadings [2, 150].

However, for $p > 1$, we see that the derivative is continuous (since $|f_{v,u}|^{p-1} \to 0$, as $f_{v,u} \to 0$) unless all loadings are simultaneously 0, since then the denominator in the derivative vanishes.

If we recalculate a directional derivative by setting $f_{v,u} = \text{sgn}(g_{v,u})g$ where $g > 0$ and letting $g \to 0$ so all components simultaneously go to zero, although with arbitrary sign, then we find

$$\left.\frac{\partial \|f_v\|_p}{\partial f_{v,u}}\right|_{f_{v,u}=\text{sgn}(g_{v,u})g, g\to 0} = \text{sgn}(f_{v,u}).$$

This shows that the directional derivative is discontinuous at the null r-vector $(0, 0, ..., 0)^T$. So for $p > 1$ the penalty can produce simultaneous zeroing of all loadings at a given variable.

Consider a given set of loadings for variable v; $f_{v,u}, u = 1, ...r$, and let $f_{v,u^*} = \max_{1\leq u\leq r} |f_{v,u}|$. Then

$$\begin{aligned} \|f_v\|_p &= \left(\sum_{u=1}^{r} |f_{v,u}|^p\right)^{1/p} \\ &= |f_{v,u^*}| \left(\sum_{u=1}^{r} |\frac{f_{v,u}}{f_{v,u^*}}|^p\right)^{1/p}. \end{aligned}$$

But now for each u, $|f_{v,u}/f_{v,u^*}| \leq 1$. Thus for $p > p' \geq 1$, $|f_{v,u}/f_{v,u^*}|^p \leq |f_{v,u}/f_{v,u^*}|^{p'}$ so as p decreases to 1, the strength of the penalty weakens. On the other hand l_∞ thus provides the strongest penalty. In our earlier work [158], we used l_∞ and l_4. Here we have chosen l_2 partly for mathematical simplicity, but also because of rotational invariance, i.e., if R is orthogonal then $\|f_v\|_p = \|Rf_v\|_p$ only for $p = 2$.

We note that the l_∞ penalty has been used in a regression setting [154], and a version of l_2 in regression setting [172]. Of course, our problem setting here is totally different to regression; our work [158] and here, was developed independently of these references.

To explain the orthogonality constraint we drop it, and suppose F has SVD, $U\Lambda V^T$. The penalty is

$$\|f_v\|_2 = \sqrt{\sum_{u=1}^{r} u_{v,u}^2 \lambda_u}$$

and so the components are unequally weighted by the eigenvalues. This does not lead to useful results.

3.2.2 Identifiability of the svnPCA Model

The svnPCA cost function is identifiable iff $J_{\theta_1}(y) = J_{\theta_2}(y) \Leftrightarrow \theta_1 = \theta_2$ for all θ_1, θ_2. To investigate identifiability we introduce a $r \times r$ orthonormal matrix R and define

$$\begin{aligned} F_R &= FR \\ \Lambda_R &= R^T \Lambda R. \end{aligned}$$

We clearly have

$$\rho(F) = \rho(F_R).$$

Furthermore, note that the log-likelihood only depends on Ω. We have

$$\begin{aligned} \Omega &= F\Lambda F^T + \sigma^2 I_M \\ &= F_R \Lambda_R F_R^T + \sigma^2 I_M \end{aligned}$$

where $\Lambda_R = R^T \Lambda R$ has to be diagonal. Clearly it is only diagonal if R and Λ are commutative

$$\Lambda R = R\Lambda.$$

These matrices are commutative iff R is diagonal, and since it is orthonormal its diagonal elements must be either equal to 1 or -1. In that case we have $\Lambda = \Lambda_R$ and the columns of F_R and F are equal up to a sign. Therefore we conclude that the svnPCA cost function is identifiable up to the signs of the columns of the loading matrix F.

3.2.3 Smoothed l_2 Penalty

Since the penalty term $\|f_v\|_2$ is not smoothly differentiable the classical optimization theory does not apply. As a remedy, we propose to use the following smooth approximation [135, 164]:

$$\rho_\gamma(F) = \frac{1}{M}\sum_{v=1}^{M}\left((\|f_v\|_2^2 + \gamma^2)^{1/2} - \gamma\right)$$

where γ is a small mollifier parameter. The papers [143, 145] list desirable properties of smoothed penalty functions. The most important ones are convexity; symmetry; allows discontinuity. The paper [135] pointed out that this penalty function obeys all those properties. We now show that we can obtain a result arbitrarily close to the $\gamma = 0$ solution. Let $\hat{\theta}$ be the minimizer of $J_\theta^\gamma(y)$ and $\tilde{\theta}$ be the minimizer of $J_\theta(y)$ then consider that

$$\begin{aligned} |J_{\hat{\theta}}(y) - J_{\tilde{\theta}}(y)| &= |(J_{\hat{\theta}}(y) - J_{\hat{\theta}}^\gamma(y)) + (J_{\hat{\theta}}^\gamma(y) - J_{\tilde{\theta}}^\gamma(y)) \\ &+ (J_{\tilde{\theta}}^\gamma(y) - J_{\tilde{\theta}}(y))| \\ &\leq 2\sup_\theta |J_\theta^\gamma(y) - J_\theta(y)| \end{aligned}$$

since the middle term is negative. Now consider that

$$\begin{aligned}
|J_\theta^\gamma(y) - J_\theta(y)| &= \frac{h}{M}|\sum_{v=1}^{M}\left(\sqrt{\|f_v\|_2^2+\gamma^2} - \gamma - \|f_v\|_2\right)| \\
&= \frac{h}{M}|\sum_{v=1}^{M}\frac{\|f_v\|_2^2+\gamma^2-(\|f_v\|_2+\gamma)^2}{\sqrt{\|f_v\|_2^2+\gamma^2}+\gamma+\|f_v\|_2}| \\
&= \frac{h}{M}|\sum_{v=1}^{M}\frac{-2\gamma\|f_v\|_2}{\sqrt{\|f_v\|_2^2+\gamma^2}+\gamma+\|f_v\|_2}| \\
&\leq h\gamma.
\end{aligned}$$

We thus have

$$|J_{\hat{\theta}}(y) - J_{\tilde{\theta}}(y)| \leq h\gamma$$

which can be made arbitrarily small. Note in the following discussion we set

$$J_\theta(y) = -\frac{1}{M}l_\theta(y) + \rho_\gamma(F).$$

3.3 Sparse Variable Noisy PCA

Due to the non-linear penalty term it is not possible to derive a closed form expression for the MLE, so we resort to iterative algorithms. We propose to use a cyclic descent algorithm for this problem. The basic idea behind cyclic descent algorithms is split the parameter vector into blocks and then to minimize the cost with respect to each of those blocks. The algorithm cycles through these block optimizations until convergence. Now we state the svnPCA algorithm.

Algorithm III.2 (svnPCA)**.** The svnPCA algorithm is given by, where subscript 0 denotes current iteration, and subscript 1 denotes next iteration.

Initialization:

We initialize the iteration at the nPCA MLE solutions (3.4).

Cyclic Iteration:

F-step: Given Λ_0, σ_0^2 get F_1

$$F_1 = \begin{cases} \operatorname{argmin}_{F \in R^{M \times r}} J_{(F,\Lambda_0,\sigma_0^2)}(y) \\ \text{Subject to } F^T F = I_r. \end{cases} \tag{3.5}$$

This step is achieved by the geodesic steepest descent algorithm.

Λ-step: Given F_1, get σ_1^2, Λ_1

$$\sigma_1^2 = \frac{\operatorname{tr}(S_y) - \operatorname{tr}(F_1^T S_y F_1)}{M - r} \tag{3.6}$$

$$\Lambda_1 = \operatorname{dg}(F_1^T S_y F_1) - \sigma_1^2 I_r \tag{3.7}$$

where dg() sets the off-diagonal elements to zero and keeps the diagonal elements.

Stop Condition:

If

$$\frac{|J_{\theta_0}(y) - J_{\theta_1}(y)|}{|J_{\theta_0}(y)|} < \epsilon_1$$

then stop, otherwise set $\theta_0 = \theta_1$ and return to cyclic iteration step.

Equations (3.6) and (3.7) are derived in Appendix F. By dropping all terms that do not play a part in the optimization we substitute the cost function in Equation (3.5) with

$$-\frac{1}{2M\sigma_0^2}\operatorname{tr}(W_0^{-1} F^T S_y F) + h\rho_\gamma(F). \tag{3.8}$$

Interestingly, the log-likelihood (left) term in this expression can be interpreted as the trace of the covariance between y_t and its prediction $\hat{y}_t = F u_{0,t}$. Moreover, notice that if we let $\sigma_0^2 \to 0$ in (3.8), then it becomes

$$-\frac{1}{2M}\operatorname{tr}(F^T S_y F) + h\rho_\gamma(F). \tag{3.9}$$

It is well known [129] that optimization of the left term subject to orthogonality constraints yields traditional PCs. Now we discuss the geodesic steepest descent step in more detail.

3.3.1 The Geodesic Steepest Descent Algorithm

To get the sparse loadings F, an optimization problem with orthogonality constraints has to be solved. Classical approaches to solve constrained optimization problems are primal methods such as the gradient projection methods, and reduced gradients methods, and dual methods that work with the Lagrangian [173, 89, 14].

Recently, there has been great interest in so-called geodesic algorithms that reformulate the constrained problem as an unconstrained optimization problem on the constraint manifold. They are iterative algorithms that 1) guarantee that the update at each step of the algorithm satisfies the constraint 2) the updates move along a geodesic which is a path of shortest distance on the constraint surface.

The geodesic steepest descent algorithm was first suggested by Luenberger in [89]. That paper showed that to provide a guarantee for convergence for steepest descent procedure on a manifold it was necessary to take descent steps along a geodesic. Computational issues were not discussed.

A general geodesic steepest descent algorithm is very computationally demanding since a nonlinear system of ordinary differential equation has to be solved, at each iteration step, to compute the geodesic. However, the orthogonality constraint that we are dealing with has received special attention, e.g., a Procrustes problem on the Stiefel manifold is discussed in [40], optimization on the Stiefel manifold in the context of blind source separation is considered in [36, 26, 46, 107]. These orthogonality constraints define the Stiefel manifold, which is defined as

$$\mathrm{St}(M,r) = \{F \in R^{M\times r} : h(F) = F^T F - I_r = 0\}.$$

Associated with a point F [1] on the Stiefel manifold is a tangent space

$$\begin{aligned} \mathfrak{T}(F) &= \{dF \in R^{M\times r} : dh(F) = 0\} \\ &= \{dF \in R^{M\times r} : dF^T F + F^T dF = 0\} \end{aligned}$$

which is an $Mr - r(r+1)/2$ dimensional vector space. A tangent vector dF can be written in the following forms:

$$\begin{aligned} dF &= FA + F_\perp B \qquad (3.10) \\ &= FA + (I_M - FF^T)C \end{aligned}$$

where A is $r \times r$ skew-symmetric matrix ($A^T = -A$), B is $(M-r)\times r$ arbitrary matrix, C is $M \times r$ arbitrary matrix, and $F_\perp$ is any $M \times (M-r)$ matrix such that $FF^T + F_\perp F_\perp^T = I_M$. The orthogonal complement of the tangent space is the normal space which is a $r(r+1)/2$ dimensional vector space and can be written as

$$\mathfrak{N}(F) = \{N \in R^{M\times r} : N = FS\}$$

where S is $r \times r$ symmetric.

It is customary when working with the Stiefel manifold to assign the following metric to each tangent space [38, 92]:

$$\begin{aligned} \langle X_1, X_2\rangle &= \mathrm{tr}(X_1^T(I_M - \frac{1}{2}FF^T)X_2) \\ X_1, X_2 \in \mathfrak{T}(F) \quad &, \quad F \in \mathrm{St}(M,r). \end{aligned}$$

[1] Actually this is only true for regular points which are points F where the $r^2 \times Mr$ Jacobian matrix $Dh(F)$ has full row rank r^2. It is easy to verify that all points on the Stiefel manifold satisfy this, and are therefore regular.

A geodesic on $\mathrm{St}(M,r)$ is a smooth curve $\tilde{F}(t) \in \mathrm{St}(M,r), 0 \leq t \leq s$ that minimizes the functional

$$\int_0^s \| \frac{d\tilde{F}(t)}{dt} \| dt$$

with respect to all other curves on $\mathrm{St}(M,r)$. The Stiefel gradient $\tilde{\nabla}_F J_\theta(y)$ of the function $J_\theta(y)$ at F is defined to be the tangent vector $\tilde{\nabla}_F J_\theta(y)$ that satisfies

$$\frac{dJ_{(\tilde{F}(t),\Lambda_0,\sigma_0^2)}(y)}{dt}|_{t=0} = \langle \tilde{\nabla}_F J_\theta(y), \Delta \rangle$$

where $\Delta = \frac{d\tilde{F}}{dt} |_{t=0}$. Solving this yields

$$\tilde{\nabla}_F J_\theta(y) = \frac{\partial J_\theta(y)}{\partial F} - F \frac{\partial J_\theta(y)}{\partial F^T} F. \tag{3.11}$$

where

$$\frac{\partial J_\theta(y)}{\partial F_0} = -\frac{S_y F_0 W_0^{-1}}{\sigma_0^2} + h D_\gamma F_0 \tag{3.12}$$

and

$$D_\gamma = \mathrm{diag}\left(\frac{(\|f_1\|^2 + \gamma^2)^{-1/2}}{M}, ..., \frac{(\|f_M\|^2 + \gamma^2)^{-1/2}}{M} \right).$$

The important paper [38] worked out an explicit formula for the geodesic on the Stiefel manifold and developed a Newton and a conjugate gradient procedure. The formula requires a computation of the matrix exponential, a task that is known to be numerically challenging [103]. However, since the matrix exponential is low dimensional ($2r \times 2r$) and skew-symmetric, this was not found to be a problem in this work. A different approach is developed in [92].

There are four iteration steps associated with the geodesic descent algorithm which we describe in the following, note that subscript 0 denotes current iteration, and 1 denotes next iteration:

Direction:

Compute the Stiefel gradient $\tilde{\nabla}_{F_0} J_\theta(y)$ by using Equation (3.11). We can rewrite the Stiefel gradient $\tilde{\nabla}_{F_0} J_\theta(y)$ in the following way

$$\tilde{\nabla}_{F_0} J_\theta(y) \quad = \quad F_0 A + (I_M - F_0 F_0^T) C \tag{3.13}$$

where

$$\begin{aligned} A &= \frac{W_0^{-1} F_0^T S_y F_0 - F_0^T S_y F_0 W_0^{-1}}{\sigma_0^2} \\ C &= h D_\gamma F_0 - \frac{S_y F_0 W_0^{-1}}{\sigma_0^2}. \end{aligned}$$

showing that $\tilde{\nabla}_{F_0} J_\theta(y) \in \mathfrak{T}(F_0)$.

Geodesic:

Move along the geodesic on the Stiefel manifold emanating from F_0 in direction of $-\tilde{\nabla}_{F_0} J_\theta(y)$. The geodesic is given by [38]

$$\tilde{F}(t) = F_0 M(t) + Q N(t) \tag{3.14}$$

where

$$\begin{aligned} \begin{pmatrix} M(t) \\ N(t) \end{pmatrix} &= \exp t \begin{pmatrix} A & -R^T \\ R & 0_r \end{pmatrix} \begin{pmatrix} I_r \\ 0_r \end{pmatrix} \\ QR &= (I_M - F_0 F_0^T) \tilde{\nabla}_{F_0} J_\theta(y) \end{aligned}$$

where QR is computed using the compact QR decomposition. To compute the matrix exponential we use the Pade approximation with scaling and squaring [59].

Linesearch:

The new estimate is taken as $F_1 = \tilde{F}(t^*)$ where t^* is the first local minimum of $J_{(\tilde{F}(t), \Lambda_0, \sigma_0^2)}(y)$, obtained by line search.

Stop Condition:

If

$$\frac{|J_{\theta_0}(y) - J_{(F_1,\Lambda_0,\sigma_0^2)}(y)|}{|J_{\theta_0}(y)|} < \epsilon_2$$

then stop, else set $F_0 = F_1$ and return to the Direction step.

Under weak conditions [89] proved that the geodesic steepest descent algorithm is globally convergent to a point where the Stiefel gradient vanishes.

It is of interest to know the behavior of the geodesic and in particular to know if it is periodic. Observe that the argument $U = \begin{pmatrix} A & -R^T \\ R & 0_r \end{pmatrix}$ of the exponential matrix is a skew symmetric matrix. We can write

$$\exp(tU) = I_{r^2} + \sum_{j=1}^{r} (\sin(\theta_i t)U_i + (1 - \cos(\theta_i t))U_i^2)$$

where $\{\theta_1, ..., \theta_r\}$ is the set of the distinct positive square roots of the $2r$ positive eigenvalues of $-1/4(U - U^T)^2$ and $U_1, ..., , U_r$ are skew symmetric matrices that can be uniquely determined from U [57]. From this we see that the geodesic is a matrix polynomial weighted by sinusoids at frequencies $\{\theta_1, ..., \theta_r\}$. Thus, the geodesic is periodic if and only if θ_i/θ_j is a rational number for $1 \leq i, j \leq r$.

3.4 The Geodesic Newton Algorithm

In this section we derive a Newton algorithm on the Stiefel manifold. The difference between the Newton algorithm and the geodesic steepest descent from previous section is that the direction is not determined by the negative of the Stiefel gradient $\tilde{\nabla}_F J_\theta(y)$ but the Newton direction X which is a tangent vector that solves

$$\min_{X \in \mathfrak{T}} \frac{1}{2} \text{Hess}(X, X) + \langle \tilde{\nabla}_F J_\theta(y), X \rangle \tag{3.15}$$

where Hess is the Stiefel Hessian defined by

$$\mathrm{Hess}(X,X) = \frac{d^2}{dt^2} J_{(\tilde{F}(t),\Lambda_0,\sigma_0^2)}(y)\mid_{t=0} \tag{3.16}$$

where $\tilde{F}(t)$ is a geodesic with $\frac{d\tilde{F}(t)}{dt} = X$. Now we proceed to work out an explicit formula for the Newton direction X, and this requires few steps. The main difficulty we are facing is that X needs to lie in a tangent space.

First we rewrite the Stiefel Hessian in more convenient form [38]:

$$\begin{aligned}\mathrm{Hess}(X,X) &= J_{FF}(X,X) + \mathrm{tr}\left(\frac{\partial J_\theta(y)}{\partial F^T} X F^T X\right) \\ &- \mathrm{tr}\left(F^T \frac{\partial J_\theta(y)}{\partial F} X^T (I_M - FF^T) X\right) \end{aligned} \tag{3.17}$$

where $J_{FF}(X,X)$ is the second differential of the svnPCA cost function given by

$$\begin{aligned} J_{FF}(X,X) &= \mathrm{tr}(-\frac{1}{\sigma_0^2} W_0^{-1} X^T S_y X + h X^T D_\gamma X \\ &+ h F^T \tilde{D}_\gamma [F X^T \odot I_M] X) \end{aligned}$$

where

$$\tilde{D}_\gamma = \mathrm{diag}\left(-\frac{(\|f_1\|^2+\gamma^2)^{-3/2}}{M}, ..., -\frac{(\|f_M\|^2+\gamma^2)^{-3/2}}{M}\right)$$

and $\odot$ denotes the Hadamard product or a point-wise product of matrix elements. We need to ensure that the Newton direction X lies in the tangent space, otherwise, $\tilde{F}$ would step of the manifold. Therefore we write

$$X = F A_X + F_\perp B_X. \tag{3.18}$$

Substituting (3.18) into (3.17) yields

$$\mathrm{Hess}(X,X) = \tilde{\Psi}_{11}(A_x,A_x) + \tilde{\Psi}_{21}(A_x,B_x) + \tilde{\Psi}_{12}(A_x,B_x) + \tilde{\Psi}_{22}(B_x,B_x). \tag{3.19}$$

Now we work out an expression for each term in (3.19) starting with the first term

$$\begin{aligned}\tilde{\Psi}_{11}(A_x, A_x) &= \operatorname{tr}(-\frac{W_0^{-1}A_x^T F^T S_y F A_x}{\sigma_0^2} + hA_x^T F^T D_\gamma F A_x \\ &+ hA_x^T F^T \tilde{D}_\gamma [F A_x^T F^T \odot I_M] F A_x + \frac{\partial J_\theta(y)}{\partial F^T} F A_x F^T F A_x \\ &- \frac{\partial J_\theta(y)}{\partial F^T} F A_x^T F^T (I_M - FF^T) F A_x).\end{aligned}$$

Since $FA^T F$ has zero diagonal, term number 3 of this expression is zero, and therefore $FA^T F \otimes I_M = 0$. Furthermore, the last term is zero since $F^T(I_M - FF^T) = 0$. Now use the expression for $\frac{\partial J_\theta}{\partial F}$ given in Equation (3.12) and get

$$\begin{aligned}\tilde{\Psi}_{11}(A_x, A_x) &= \operatorname{tr}(-\frac{W_0^{-1}A_x^T F^T S_y F A_x}{\sigma_0^2} + hA_x^T F^T D_\gamma F A_x \\ &- \frac{W_0^{-1} F^T S_y F A_x A_x}{\sigma_0^2} + hF^T D_\gamma F A_x A_x) \\ &= \operatorname{tr}(-\frac{A_x W_0^{-1}(A_x^T F^T S_y F - F^T S_y F A_x^T)}{\sigma_0^2}). \qquad (3.20)\end{aligned}$$

To get the last expression the cyclic property of the trace ($\operatorname{tr}(AB) = \operatorname{tr}(BA)$) and $A_x^T = -A_x$ was used. Now we write the above expression in more standard quadratic form

$$\tilde{\Psi}(A_x, A_x) = a_x^T \tilde{\Psi}_{11} a_x \qquad (3.21)$$

where $a_x = \operatorname{vec}(A_x)$ is a $r^2 \times 1$ vector constructed by stacking the columns of A_x above each other. This is readily accomplished by applying the well known identity

$$\operatorname{tr}(ABCD) = \operatorname{vec}^T(D) A \otimes C^T \operatorname{vec}(B^T) \qquad (3.22)$$

on (3.20). This yields

$$\tilde{\Psi}_{11} = \frac{-F^T S_y F \otimes W_0^{-1}}{\sigma_0^2} + \frac{I_r \otimes W_0^{-1} F^T S_y F}{\sigma_0^2}.$$

The quadratic form (3.21) needs to be simplified more since the vector a_x is not freely adjustable. We follow a method presented in [40].

A skew-symmetric matrix can be written in the form $A = A_L - A_L^T$ where A_L is a strictly lower triangular matrix with $[A_L]_{ij} = A_{ij}$ for $i > j$. This fact yields

$$a_x = (I_{r^2} - K_r)a_{Lx}$$

where K_r is the commutation matrix [91] ($K_r\text{vec}(A) = \text{vec}(A^T)$). Since A_{Lx} is strictly lower triangular, a_{Lx} has at most $r(r-1)/2$ nonzero elements. Now define the $r^2 \times r(r-1)/2$ matrix

$$E = [e_2, e_3, \cdots, e_r, e_{r+3}, \cdots, e_{2r}, e_{2r+4}, \cdots].$$

Then the vector

$$\tilde{a}_x = E^T a_{Lx}$$

contains the nonzero elements of a_{Lx}. We now have

$$\tilde{\Psi}_{11}(A_x) = \tilde{a}_x^T \Psi_{11} \tilde{a}_x$$

where

$$\Psi_{11} = E^T(I_{r^2} - K_r)\tilde{\Psi}_{11}(I_{r^2} - K_r)E.$$

The elements of $\tilde{a}_x$ are unconstrained which is what we wanted.

Now we turn our attention to the second term of Equation (3.19). We have

$$\begin{aligned}
\tilde{\Psi}_{21}(A_x, B_x) &= \text{tr}(\frac{-W_0^{-1}A_x^T F^T S_y F_\perp B_x}{\sigma_0^2} + hA_x^T F^T D_\gamma F_\perp B_x \\
&+ hF^T\tilde{D}_\gamma[FA_x^T F \odot I_M]F_\perp B_x + \frac{\partial J_\theta}{\partial F^T} FA_x F^T F_\perp B_x \\
&- \frac{\partial J_\theta}{\partial F^T} FA_x^T F^T(I_M - FF^T)F_\perp B_x) \\
&= \text{tr}(\frac{-W_0^{-1}A_x^T F^T S_y F_\perp B_x}{\sigma_0^2} + hA_x^T F^T D_\gamma F_\perp B_x).
\end{aligned}$$

We see that terms 4 and 5 in the first expression are clearly equal to zero, term 4 is zero since $F^T F_\perp = 0$. This term can be rewritten as

$$\begin{aligned} \tilde{\Psi}_{21}(A_x, B_x) &= b_x^T \Psi_{21} \tilde{a}_x \\ &= b_x^T \tilde{\Psi}_{21} (I_r^2 - K_r) E \tilde{a}_x \end{aligned} \tag{3.23}$$

where $b_x = \text{vec}(B_x)$, and

$$\tilde{\Psi}_{21} = \frac{-W_0^{-1} \otimes F_\perp^T S_y F}{\sigma_0^2} + h I_r \otimes F_\perp^T D_\gamma F.$$

Similarly we get that $\Psi_{12} = \Psi_{21}^T$. Finally,

$$\begin{aligned} \tilde{\Psi}_{22}(B_x) &= \text{tr}(-\frac{1}{\sigma_0^2} W_0^{-1} B_x^T F_\perp^T S_y F_\perp B_x \\ &+ \nu B_x^T B_x + h B_x^T F_\perp^T D_\gamma F_\perp B_x \\ &+ h F^T \tilde{D}_\gamma [F B_x^T F_\perp^T \odot I_M] F_\perp B_x). \\ &= b_x^T \Psi_{22} b_x \\ \Psi_{22} &= \frac{-W_0^{-1} \otimes F_\perp^T S_y F_\perp}{\sigma_0^2} + \nu \otimes I_{M-r} \\ &+ h I_r \otimes F_\perp^T D_\gamma F_\perp + h(F^T \tilde{D}_\gamma \otimes F_\perp^T) \text{diag}(\text{vec}(I_M))(F \otimes F_\perp) \\ \nu &= \frac{1}{\sigma^2} F^T S_y F W_0^{-1} - h F^T D_\gamma F. \end{aligned}$$

To derive the last term of Ψ_{22} we use Equation (3.22), and the identities

$$\begin{aligned} \text{vec}(A \odot B) &= \text{diag}(\text{vec}(A))\text{vec}(B) \\ \text{vec}(ABC) &= C^T \otimes A\text{vec}(B). \end{aligned}$$

Now we are able to write Equation (3.19) in an useful form:

$$\begin{aligned} \text{Hess}(X, X) &= \begin{pmatrix} \tilde{a}_x^T & b_x^T \end{pmatrix} \Psi \begin{pmatrix} \tilde{a}_x \\ b_x \end{pmatrix} \\ &= \begin{pmatrix} \tilde{a}_x^T & b_x^T \end{pmatrix} \begin{pmatrix} \Psi_{11} & \Psi_{21}^T \\ \Psi_{21} & \Psi_{22} \end{pmatrix} \begin{pmatrix} \tilde{a}_x \\ b_x \end{pmatrix}. \end{aligned}$$

Now return to Equation (3.15) which defines the Newton step. Define

$$\begin{aligned}\tilde{\nabla}_F J_\theta(y) &= FA_h + F_\perp B_h \\ h &= \mathrm{vec}(\tilde{\nabla}_F J_\theta(y))\end{aligned}$$

and simplify the right side term of Equation (3.15) in the following way

$$\begin{aligned}\langle -\tilde{\nabla}_F J_\theta(y), X\rangle &= -\mathrm{tr}(H^T(I_M - FF^T)X) \\ &= -\frac{1}{2}\mathrm{tr}(A_h^T A_x) + \mathrm{tr}(B_h^T B_x) \\ &= -\begin{pmatrix} \frac{1}{2}a_h^T & b_h^T \end{pmatrix}\begin{pmatrix} a_x \\ b_x \end{pmatrix} \\ &= -\begin{pmatrix} \frac{1}{2}a_h^T(I_{r^2} - K_r)E & b_h^T \end{pmatrix}\begin{pmatrix} \tilde{a}_x \\ b_x \end{pmatrix}.\end{aligned}$$

We can rewrite Equation (3.15) in the following way

$$\min_{\tilde{a}_x, b_x} \frac{1}{2}\begin{pmatrix} \tilde{a}_x^T & b_x^T \end{pmatrix}\Psi\begin{pmatrix} \tilde{a}_x \\ b_x \end{pmatrix} + \begin{pmatrix} \frac{1}{2}a_h^T(I_{r^2} - K_r)E & b_h^T \end{pmatrix}\begin{pmatrix} \tilde{a}_x \\ b_x \end{pmatrix}.$$

This is a standard unconstrained quadratic problem which has solution

$$\begin{pmatrix} \tilde{a}_x \\ b_x \end{pmatrix} = \Psi^{-1}\begin{pmatrix} \frac{1}{2}E^T(I_{r^2} - K_r)a_h \\ b_h \end{pmatrix}.$$

What remains to be done is to get the Newton direction X from $\tilde{a}_x$ and b_x. We have

$$a_x = (I_r^2 - K_r)E\tilde{a}_x$$

and finally

$$X = FA_x + F_\perp B_x.$$

As we will see in simulations below the geodesic Newton method is very effective for small data sets. But it is impractical for large fMRI data sets since we need to invert a very large matrix Ψ at each iteration step; note that Edelman et al. [38] suggested to use the conjugate gradient algorithm to invert, but that is still computationally expensive, and one needs to worry about the positive definiteness of the matrix. For large data sets we can alternatively use a diagonal approximation to the geodesic Newton method, i.e., force the matrix Ψ to be diagonal. In that case the inversion is trivial. The diagonal elements of Ψ_{22} are given by

$$\begin{aligned}\Psi_{22}(k,k) &= -\frac{1}{\sigma_0^2} f_{\perp(i)}^T S_y f_{\perp(i)} [W_0]_{jj}^{-1} + \nu_{jj} + f_{\perp(i)}^T D_\gamma f_{\perp(i)} \\ &+ h f_{\perp(i)}^T \mathrm{dg}(I_M \odot f_{\perp(i)} f_{(j)}^T) \tilde{D}_\gamma f_{(j)} \\ k &= (M-r)(j-1)+i, \quad j=1,...,r, i=1,...,M-r.\end{aligned}$$

Since Ψ_{11} is a small matrix we can simply compute the whole matrix and pick out the diagonal elements.

3.5 F-step: Conditions for a Minimum

In this section, we work out conditions for the solution to be a minimum point. The geodesic descent algorithm is guaranteed to converge to a point where the Stiefel gradient is equal to zero, i.e.,

$$\frac{d}{dt} J_{(\tilde{G}(t),\Lambda_0,\sigma_0^2)}(y) \mid_{t=0} = 0.$$

We are interested to know whether this point represents local minimum. From elementary calculus, we have that sufficient condition for local minimum given by

$$\mathrm{Hess}(X,X) = \frac{d^2}{dt^2} J_{(\tilde{F}(t),\Lambda_0,\sigma_0^2)}(y) \mid_{t=0} \geq 0$$

i.e., the second differential of the Hessian for all perturbations in the tangent subspace is greater or equal to zero. This is equivalent to

$$\Psi \geq 0$$

which can be checked by verifying that all the eigenvalues of Ψ are greater or equal to zero.

3.5.1 Global Convergence of the svnPCA Algorithm

While conditions for the global convergence of cyclic descent algorithms have been provided by [173, 88, 14] they are not straightforward to check. Instead we modify (to handle constraints) the following result of [141], which builds on results from [100, 80]. Let $\alpha = F$, $\beta = (\Lambda, \sigma^2)$, $\nabla_\beta J_\theta(y) = (\frac{\partial J_\theta(y)}{\partial \Lambda}, \frac{\partial J_\theta(y)}{\partial \sigma^2})$, and introduce the assumptions

(A1): $J_\theta(y), \theta = (\alpha, \beta)$ is bounded for $\theta \in \Theta$ a subset of d-dimensional Euclidian space.

(A2): $\tilde{\nabla}_\alpha J_\theta(y)$ and $\nabla_\beta J_\theta(y)$ are continuous in θ for $\theta \in \Theta$.

(A3): $\{\theta_m\}$ is bounded uniformly, m is an iteration index.

(A4): $J_{(\alpha_m, \beta_m)}(y) < J_{(\alpha_{m-1}, \beta_{m-1})}(y)$ if $\theta_m = (\alpha_m, \beta_m)$ is not a stationary point.

Theorem III.3. *Consider the cyclic descent minimization of $J_\theta(y)$, $\theta = (\alpha, \beta)$ and denote the $m-th$ iterate by θ_m. Suppose A1-A4 hold. Then*

1. *All limit points of $\{\theta_m\}$ are stationary points, i.e., satisfy $\tilde{\nabla}_\alpha J_\theta(y) = 0$ and $\nabla_\beta J_\theta(y) = 0$.*

2. *The $\{\theta_m\}$ sequence converges to a compact connected subset of the set of stationary points. Note that the subsets could be isolated.*

3. *If the set of stationary points is discrete the $\{\theta_m\} \rightarrow \theta_\infty$ is an isolated stationary point.*

Proof. See Appendix H. □

For the svnPCA algorithm assumptions A1 and A2 hold trivially. A3 holds since the F iterates are constrained $F^T F = I_r$ while $\sigma_1^2 \leq \text{tr}(S_y)/(M-r)$ and $\Lambda_1 \leq \text{tr}(S_y)I_r$.

Now we verify condition A4. First assume that α_{m-1} is a stationary point but β_{m-1} is not, then

$$J_{(\alpha_m,\beta_m)}(y) < J_{(\alpha_m,\beta_{m-1})}(y) \leq J_{(\alpha_{m-1},\beta_{m-1})}(y).$$

This follows from the following; the Λ-step is a convex optimization problem (See Appendix F); the optimization problem in the F-step is globally convergent [89]. When β_{m-1} is a stationary point but α_{m-1} is not, we get

$$J_{(\alpha_m,\beta_m)}(y) < J_{(\alpha_{m-1},\beta_m)}(y) \leq J_{(\alpha_{m-1},\beta_{m-1})}(y).$$

When neither α_{m-1} or β_{m-1} are stationary we get

$$J_{(\alpha_m,\beta_m)}(y) < J_{(\alpha_{m-1},\beta_m)}(y) < J_{(\alpha_{m-1},\beta_{m-1})}(y).$$

Thus, we conclude that assumption A4 hold. Thus, the svnPCA algorithm is globally convergent. Finally, we note that since the parameters are identified up to the sign of the columns of F the set of stationary points is discrete. Thus, according to item 3 in Theorem III.3, the svnPCA algorithm converges to an isolated stationary point.

3.6 Model Selection for svnPCA

There are two tuning parameters that need to be selected; the number of sparse PCs r, and h which controls sparseness. Due to the fact we have a ML based procedure we propose to use the BIC criterion to select them. But it is not at all obvious how to do this since h is a continuous parameter while traditional BIC can only handle integer parameters. However, we can overcome this problem with a crucial observation. After solving the optimization problem for a given h we can count the number of variables $M = M_h$ left. This allows us to calculate the penalty term. This is unusual because traditional BIC would require analytic expression for M_h. Thus the criterion becomes

$$\mathrm{BIC}(r, h) = -2l_{\hat{\theta}}(y) + \frac{\dim(\theta)\log(T)}{T}$$

where $\dim(\theta)$ is the dimension of the parameter space given by $\dim(\theta) = M_h r - r(r-1)/2 + 1$. In principle, the h and r that minimize the BIC should be chosen, but we note that the BIC should be used as a guide, and all local minimums, and values close to minimums should be looked at [86]. Notice that to decide whether a variable number v is zeroed out we use the following rule

$$\max_u(|f_{v,u}|) \leq \epsilon_3.$$

3.7 Results for svnPCA

The algorithm was applied on a simulated and a real data sets.

3.7.1 Simulation 1

The purpose of this simulation is to visualize the geodesic $\tilde{F}(t)$ given in Equation (3.14). A loading matrix $F \in R^{3\times 2}$ was generated randomly and then orthogonalized. The corresponding noise parameters where set at $\lambda_1 = 200, \lambda_2 = 50$ and $\sigma^2 = 2$.

On Figure 3.1 we visualize the geodesic by plotting $\tilde{f}_{(1)}(t)$ and $\tilde{f}_{(2)}(t)$ on the unit sphere in $R^{3\times 1}$. We can see that the $\tilde{f}_{(1)}(t)$ and $\tilde{f}_{(2)}(t)$ trace small circles of the sphere. Of course, at a particular t, the vectors are that trace the small circles are orthonormal. Figure 3.2 shows the Sparse Variable Noisy PCA (svnPCA) cost function evaluated on the geodesic. We see that the cost function looks periodic, and it has multiple minima and maxima.

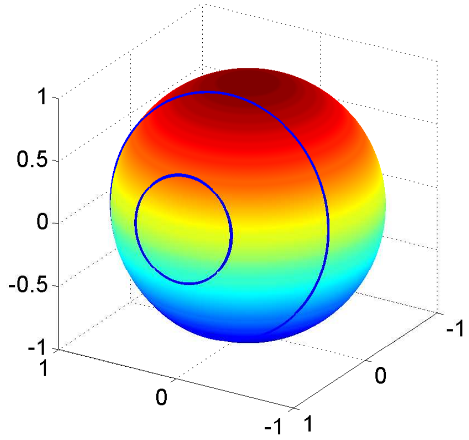


Figure 3.1: The geodesic in Simulation 1 visualized.

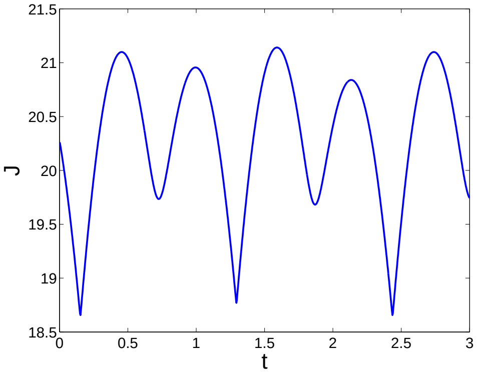


Figure 3.2: The svnPCA cost function evaluated on the geodesic in Simulation 1.

3.7.2 Simulation 2

The purpose of this simulation is to demonstrate the model selection method, and to show that the algorithm is able to zero out variables. For the simulation $T = 50$ samples from the model given in Equation (3.3) were generated with

$$\begin{aligned} F^T &= \begin{pmatrix} 0.5 & 0.5 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.71 & 0.71 \end{pmatrix} \\ \Lambda &= \begin{pmatrix} 300 & 0 \\ 0 & 50 \end{pmatrix} \\ \sigma^2 &= 2. \end{aligned}$$

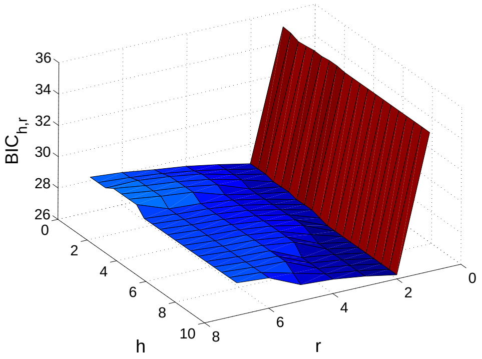


Figure 3.3: The BIC result for Simulation 3.7.2 with $\sigma^2 = 2$.

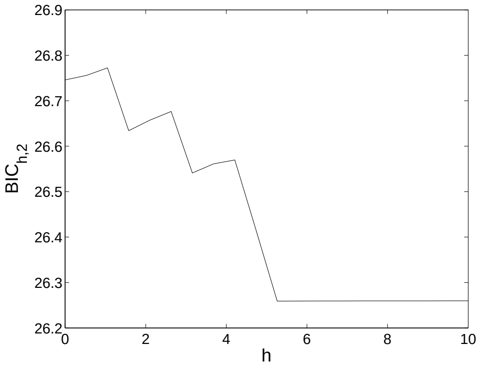


Figure 3.4: The BIC $r = 2$ profile for Simulation 3.7.2 with $\sigma^2 = 2$.

To run the simulation we set $\epsilon_1 = \epsilon_2 = \epsilon_3 = 1 \cdot 10^{-6}$, $\gamma = 1 \cdot 10^{-4}$, and BIC was computed on a grid for $r = 1, ..., 7$ and h for 20 points on the interval $[0, 10]$.

Fig. 3.3 shows the BIC result for the simulation, and Fig. 3.4 shows the BIC profile for $r = 2$. The BIC selects $h = 5.3$ and $r = 2$ which yields

$$\begin{aligned}
\hat{F} &= \begin{pmatrix} 0.5086 & 0.0675 \\ 0.4876 & 0.0777 \\ -0.0000 & 0.0000 \\ 0.0000 & -0.0000 \\ 0.4881 & 0.0594 \\ 0.5052 & -0.0075 \\ -0.0000 & -0.0000 \\ -0.0000 & -0.0000 \\ 0.0548 & -0.6949 \\ 0.0837 & -0.7091 \end{pmatrix} \\
\hat{\Lambda} &= \begin{pmatrix} 300.1116 & 0.0000 \\ 0.0000 & 51.8971 \end{pmatrix} \\
\hat{\sigma}^2 &= 2.0193.
\end{aligned}$$

For comparison, we display the nPCA solution $h = 0, r = 2$:

$$\begin{aligned}
\tilde{F} &= \begin{pmatrix} -0.5078 & 0.0672 \\ -0.4878 & 0.0779 \\ 0.0105 & 0.0069 \\ -0.0041 & 0.0069 \\ -0.4885 & 0.0592 \\ -0.5050 & -0.0094 \\ 0.0150 & -0.0294 \\ 0.0157 & -0.0323 \\ -0.0549 & -0.6947 \\ -0.0834 & -0.7080 \end{pmatrix} \\
\tilde{\Lambda} &= \begin{pmatrix} 300.3254 & 0.0000 \\ 0.0000 & 52.0380 \end{pmatrix} \\
\tilde{\sigma}^2 &= 1.9838.
\end{aligned}$$

Notice that the svnPCA solution accurately zeros out variables 2,3,7, and 8. To check whether this solution represents minimum point we compute the eigenvalues of the Hessian of the Lagrangian restricted to the tangent space, i.e., the matrix Φ

from Section 3.4. They are given by

$$\begin{pmatrix} 5.2638 \cdot 10^4 \\ 5.1963 \cdot 10^4 \\ 5.0688 \cdot 10^4 \\ 4.7655 \cdot 10^4 \\ 4.7002 \cdot 10^4 \\ 4.6449 \cdot 10^4 \\ 3.8582 \cdot 10^4 \\ 3.7017 \cdot 10^4 \\ 0.0145 \cdot 10^4 \\ 0.0137 \cdot 10^4 \\ 0.0137 \cdot 10^4 \\ 0.0137 \cdot 10^4 \\ 0.0028 \cdot 10^4 \\ 0.0027 \cdot 10^4 \\ 0.0027 \cdot 10^4 \\ 0.0017 \cdot 10^4 \\ 0.0004 \cdot 10^4 \end{pmatrix}$$

All eigenvalues are greater than zero. Thus, the restricted Hessian is positive definite and the solution point represents minimum.

We also performed the same simulation for $\sigma^2 = 35$. Figures 3.5 and 3.6 show the

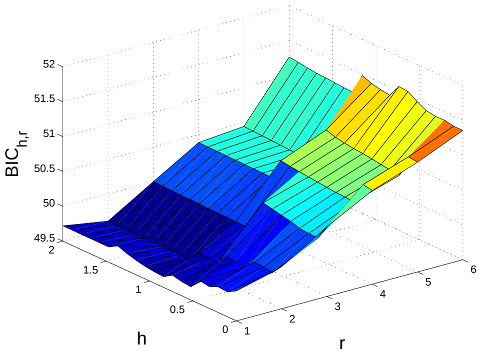


Figure 3.5: The BIC result for Simulation 3.7.2 with $\sigma^2 = 35$.

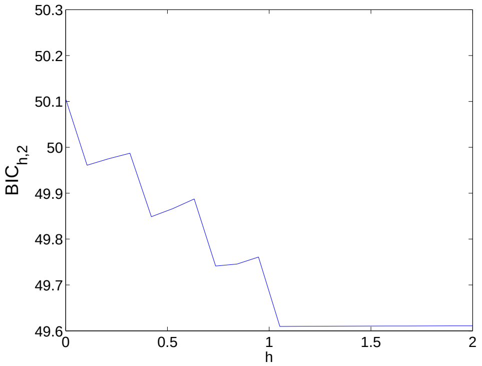


Figure 3.6: The BIC profile $r = 2$ for Simulation 3.7.2 with $\sigma^2 = 35$.

corresponding BIC plots. The BIC picked $r = 2, h = 1.05$ which corresponds to

$$\hat{F} = \begin{pmatrix} -0.5533 & 0.2222 \\ -0.4306 & 0.1867 \\ 0.0000 & 0.0000 \\ -0.0000 & -0.0000 \\ -0.4437 & 0.1033 \\ -0.5329 & -0.2181 \\ -0.0000 & -0.0000 \\ -0.0000 & -0.0000 \\ -0.0339 & -0.6077 \\ -0.1634 & -0.6987 \end{pmatrix}$$

and

$$\begin{aligned}\hat{\Lambda} &= \begin{pmatrix} 300.2472 & 0.0000 \\ 0.0000 & 52.7734 \end{pmatrix} \\ \hat{\sigma}^2 &= 34.7463\end{aligned}$$

For comparison, we display the nPCA solution $(h = 0, r = 2)$

$$\begin{aligned}\tilde{F} &= \begin{pmatrix} -0.5347 & 0.2089 \\ -0.4367 & 0.1913 \\ 0.0457 & -0.0427 \\ -0.0027 & -0.0674 \\ -0.4562 & 0.1024 \\ -0.5247 & -0.2371 \\ 0.0538 & -0.1186 \\ 0.0794 & -0.1068 \\ -0.0387 & -0.6081 \\ -0.1648 & -0.6719 \end{pmatrix} \\ \tilde{\Lambda} &= \begin{pmatrix} 300.1773 & 0.0000 \\ 0.0000 & 52.7868 \end{pmatrix} \\ \tilde{\sigma}^2 &= 34.7523.\end{aligned}$$

By looking at the nPCA loadings $\tilde{F}$ we see that it is hard to judge whether the noise variables are noise or not. But the svnPCA again correctly zeros out the noisy variables.

The svnPCA algorithm converged very quickly for this simulation, and in fact for every example we have tried it on. Convergence was usually attained after < 5 cyclic iteration. However, the F-step takes longer to converge. This is apparent, for this example, from looking at the eigenvalues of the restricted Hessian matrix for reasons we now describe. If we call A and a the largest and smallest eigenvalues of the restricted Hessian matrix, respectively. Then the references [89, 88] show that (close to the solution) the sequence of objective values converges to the solution linearly with ratio no greater than

$$\left(\frac{A-a}{A+a}\right)^2$$

which in this case is equal to 0.9997 which is rather slow. Below we look at some convergence plots for the case where $\sigma^2 = 2$.

Figure 3.7 shows a plot of the relative difference of the cost function between adjacent iteration steps for h and r that corresponded to the minimum of the BIC. We see that a relatively quick convergence is followed by a slow convergence phase from iteration step 100. Figure 3.8 shows a plot of the norm of the projected gradient. The oscillations seen on this plot are intriguing. However, the reason for them is unknown. Figure 3.9 shows a plot of the norm of the difference between F_1 and F_0.

A considerable increase in convergence speed can be obtained by using the Newton direction instead of the steepest gradient direction at the cost of increased computation load at each iteration. Figure 3.10 shows a convergence plot corresponding to Figure 3.10 for the Newton algorithm. Figure 3.11 shows the plot of the corresponding norm of the Stiefel gradient, and Figure 3.12 shows the norm between the difference between F_1 and F_0. These plots show significant increase in convergence speed, under 10 iteration compared to over 700 iteration for the geodesic steepest descent.

3.7.3 Real data

The algorithm was applied on the AFNI fMRI data introduced in Section 4.4.1.

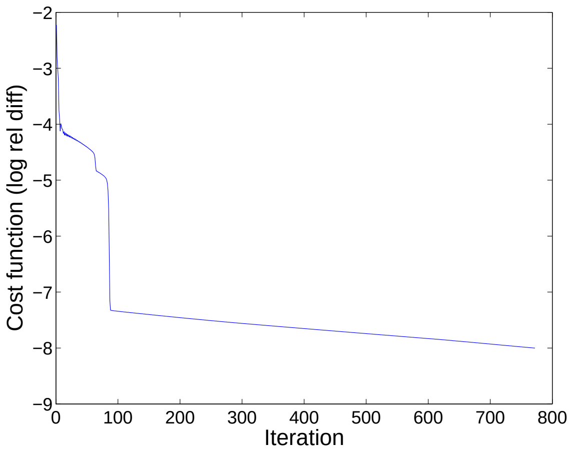


Figure 3.7: A plot of the (log10) relative difference of the cost for the geodesic steepest descent for Simulation 3.7.2.

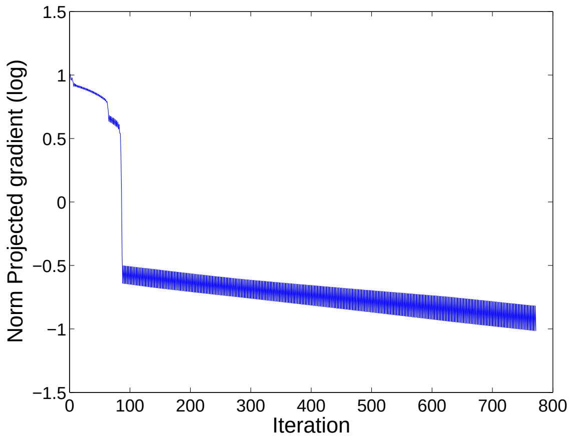


Figure 3.8: A plot of the norm of the Stiefel gradient for the geodesic steepest descent for Simulation 3.7.2.

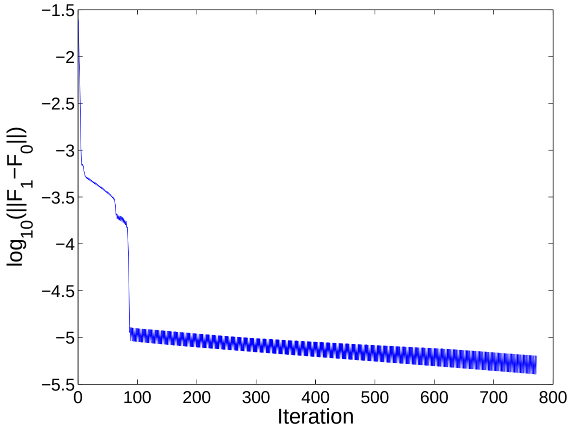


Figure 3.9: A plot of the norm of the log difference $F_1 - F_0$ for the geodesic steepest descent for Simulation 3.7.2.

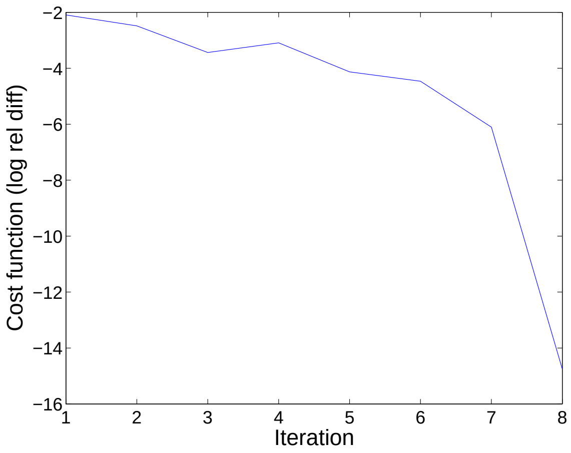


Figure 3.10: A plot of the (log10) relative difference of the cost for the geodesic Newton for Simulation 3.7.2.

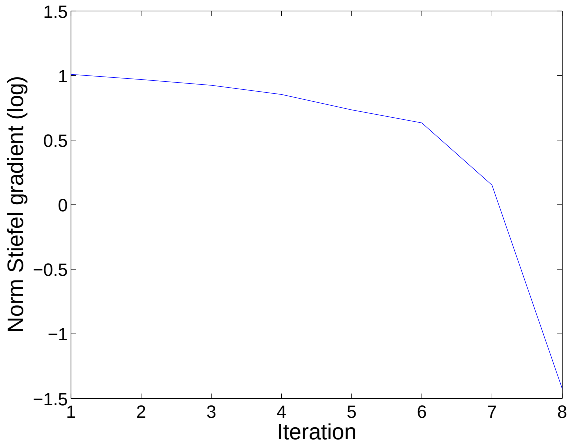


Figure 3.11: A plot of the norm of the Stiefel gradient for the geodesic Newton for Simulation 3.7.2.

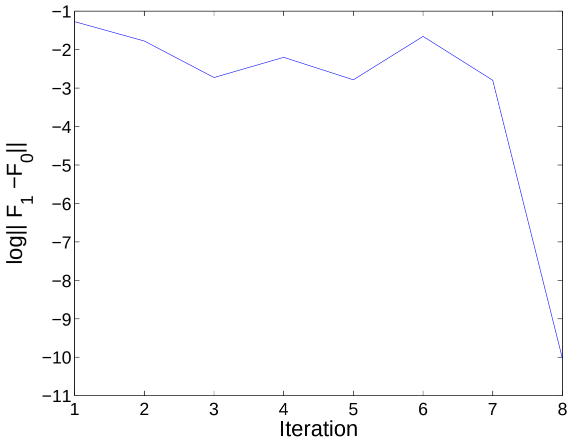


Figure 3.12: A plot of the norm of the log difference $F_1 - F_0$ for the geodesic Newton for Simulation 3.7.2.

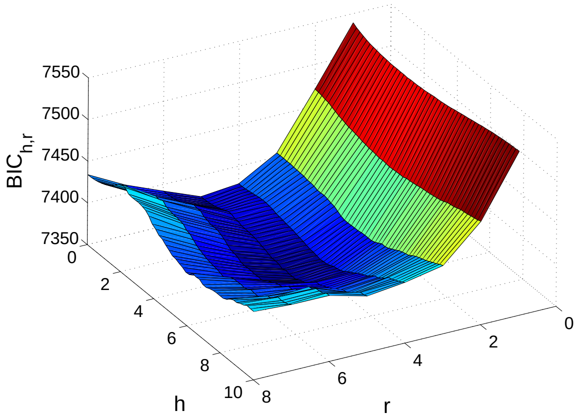


Figure 3.13: The svnPCA BIC result for the AFNI fMRI data

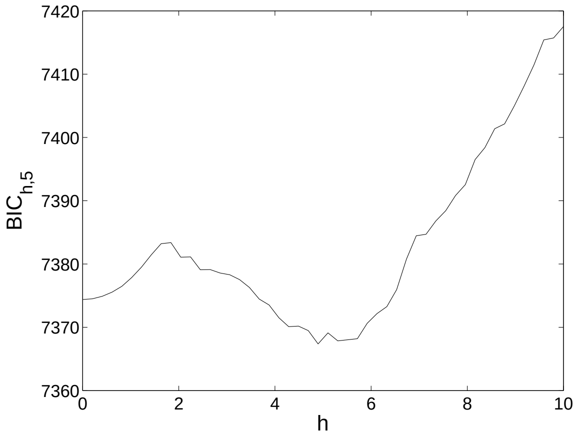
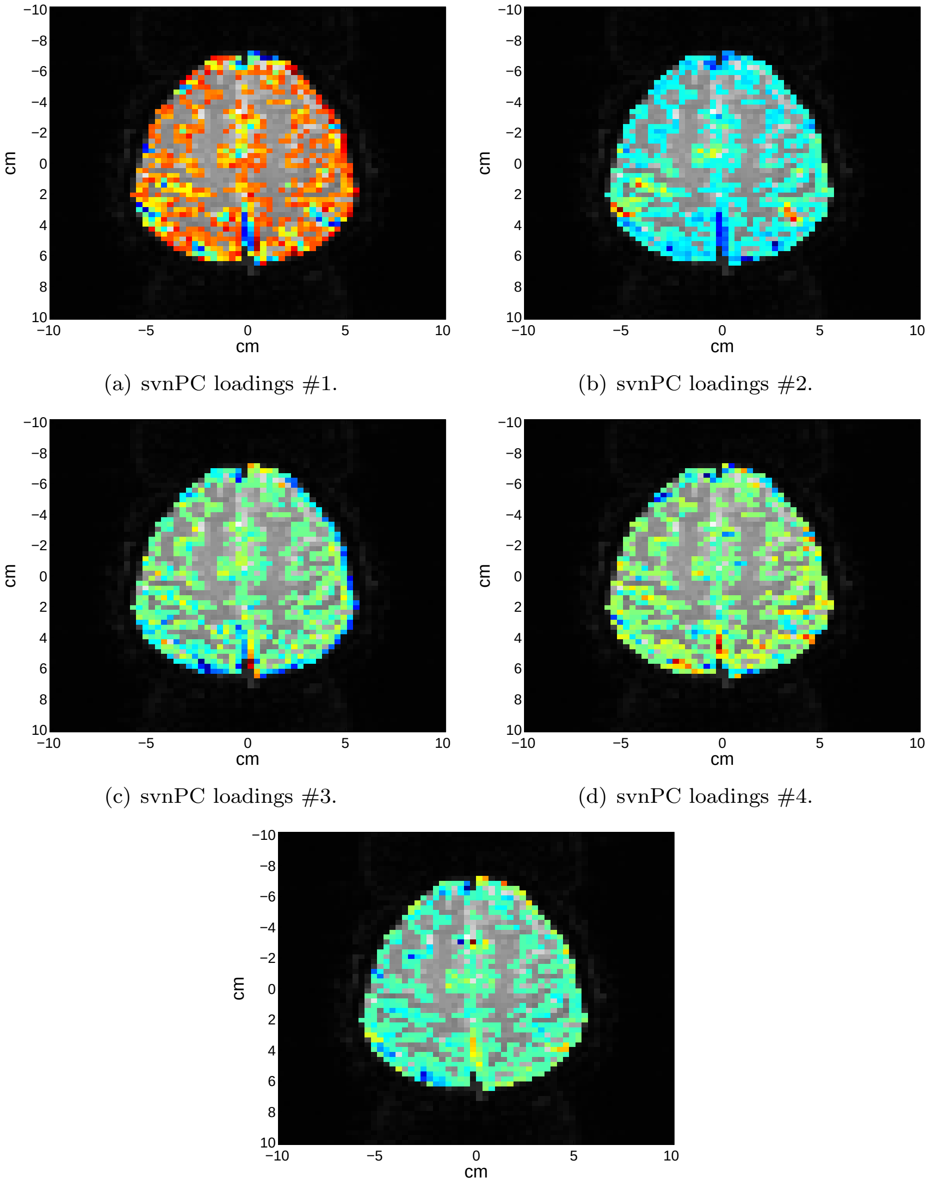


Figure 3.14: The svnPCA BIC $r = 5$ profile.

BIC was computed on a grid for $r = \{1, ..., 8\}$ and 50 points sampled uniformly on the interval $h = [0, 20]$. We set $k_1 = k_2 = 1 \cdot 10^{-5}$, $k_3 = 5 \cdot 10^{-4}$, and $\gamma = 1 \cdot 10^{-4}$.

Figure 3.13 shows the BIC result for the data. The minimum BIC occurred at $r = 5$ svnPCs profile which is shown on Figure 3.14. We picked $h = 5.92$ which was close to this minimum and corresponded to $M_h = 669$ voxels. Figure 3.15 depicts the loadings (columns of F) of the 5 svnPCs plotted spatially. Sparse variable nPC #2 has high loadings in the motor cortex clearly capturing motor activity. Figure 3.16 shows the corresponding 5 svnPCA timeseries ($u_{t,j}, t = 1, ..., T, j = 1, ..., 5$). Figures 3.17 to 3.19 show convergence plots for the geodesic steepest descent method.



(a) svnPC loadings #1.

(b) svnPC loadings #2.

(c) svnPC loadings #3.

(d) svnPC loadings #4.

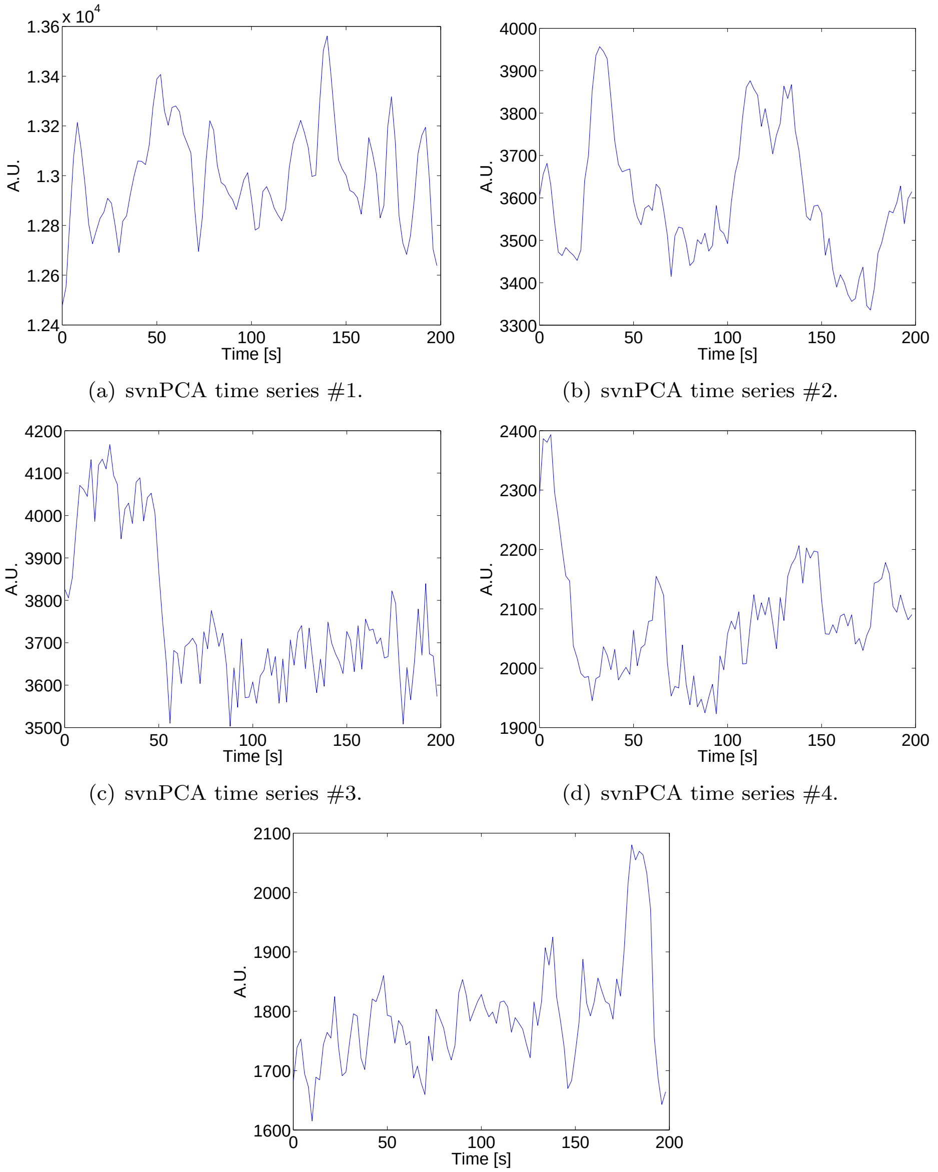
(e) svnPC loadings #5.

Figure 3.15: The svnPCA sparse loadings (columns of F) plotted spatially.

Figure 3.20 to 3.22 show convergence plots for the diagonal geodesic Newton method, illustrating quicker convergence.

3.8 Other svnPCA Approaches

In this section, we investigate other approach to the svnPCA problem. We optimize a different cost function that is based on the l_0 penalty. By using the EM



(a) svnPCA time series #1.

(b) svnPCA time series #2.

(c) svnPCA time series #3.

(d) svnPCA time series #4.

(e) svnPCA time series #5.

Figure 3.16: The svnPCA time series ($u_{t,j}$).

algorithm, which we present below, we show that we do not need to make smooth approximation to the penalty. Other notable difference is that we allow the penalty to depend on σ^2. But, as we point out later, we do not have a global convergence theory for this EM algorithm so the following results should be considered preliminary.

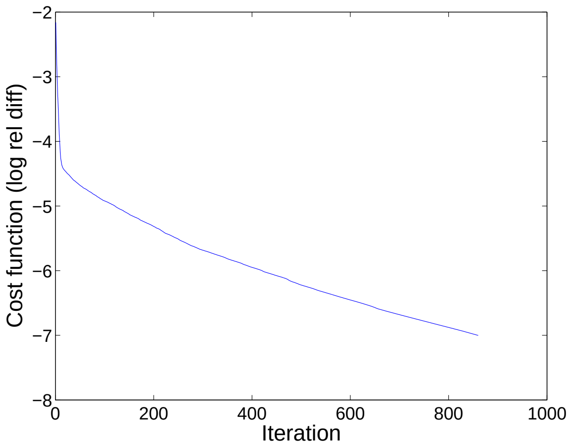


Figure 3.17: A plot of the (log10) relative difference of the cost for the geodesic steepest descent for the real fMRI data.

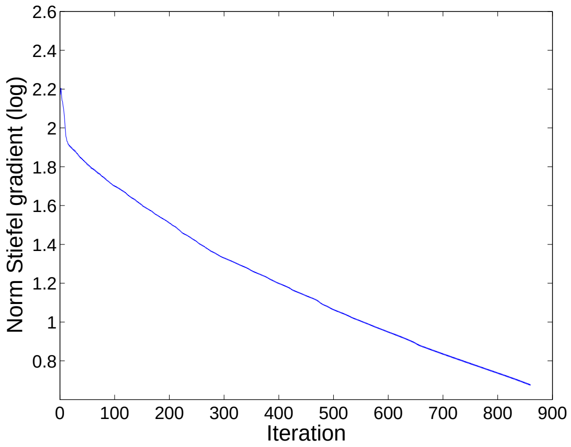


Figure 3.18: A plot of the norm of the Stiefel gradient for the geodesic steepest descent for the real fMRI data.

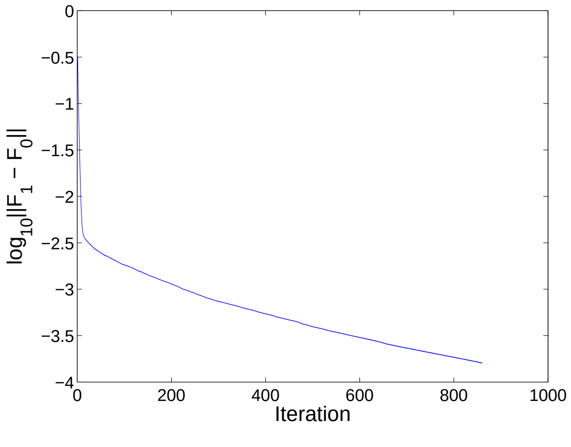


Figure 3.19: A plot of the norm of the log difference $F_1 - F_0$ for the geodesic steepest descent for the real fMRI data.

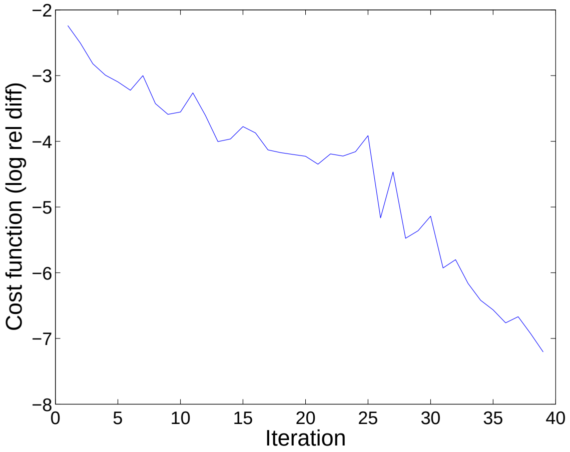


Figure 3.20: A plot of the (log10) relative difference of the cost for the diagonal geodesic Newton for real fMRI data.

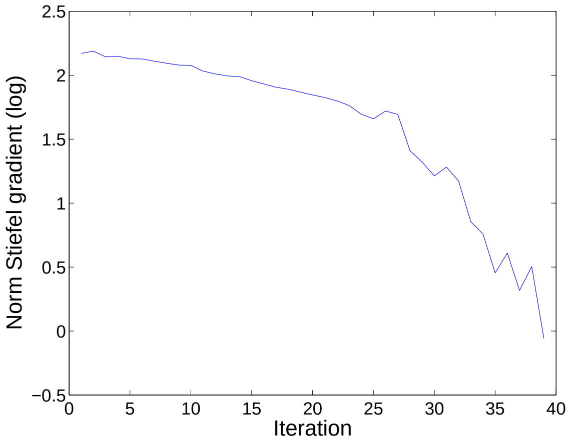


Figure 3.21: A plot of the norm of the Stiefel gradient for the diagonal geodesic Newton for the real fMRI data.

3.9 The EM Algorithm

The Expectation-Maximization (EM) algorithm [32] is an iterative algorithm to maximize a likelihood. It has proven to be extremely useful in dealing with signal processing problems like demonstrated with a wealth of examples in [104, 97].

The EM algorithm has many nice properties, e.g., unlike the more conventional steepest descent or Newton algorithm there is no need to select step size. In addition, the implementation is usually very simple. On the negative side, the EM algorithm is known, for some applications, to have a slow convergence rate. But there are ways

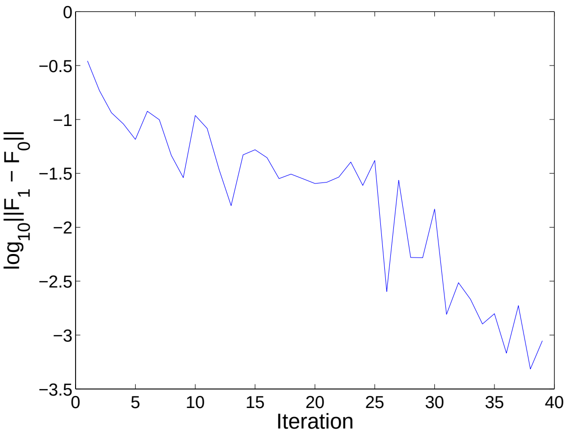


Figure 3.22: A plot of the norm of the log difference $F_1 - F_0$ for the diagonal geodesic Newton for the real fMRI data.

to accelerate it, e.g., [98].

Neal et al. [105] showed that the EM algorithm can be viewed as a cyclic ascent algorithm, albeit an unusual one since one of the quantities being optimized is infinite dimensional. A typical scenario is that we wish to estimate a parameter vector θ by maximizing a likelihood $p_\theta(y)$ which is a complex nonlinear function of θ. Many practical statistical models include a latent variable u, that if observed, the parameter θ could be more easily estimated by maximizing the so-called complete likelihood $p_\theta(y, u)$.

Instead of maximizing the log-likelihood directly, the EM algorithm works with a properly constructed surrogate function and maximizes that instead. Below we will present this surrogate function, and illustrate that maximizing it leads to the ML solution. We will derive the EM algorithm for the case where the log-likelihood is penalized by some penalty $\rho(\theta)$.

Given a marginal density $\pi(u)$ of u the EM algorithm maximizes the functional

$$J(\theta, \pi) = \int \log\left(\frac{p_\theta(y, u)\exp(-\rho(\theta))}{\pi(u)}\right)\pi(u)du \tag{3.24}$$

by using the following cyclic ascent algorithm

1. Given θ_0 get $\pi_0 = \text{argmax}_\pi J(\theta_0, \pi)$.

2. Get $\theta_1 = \text{argmax}_\theta J(\theta, \pi_0)$.

In the first step of the cyclic ascent, we fix θ_0, and optimize $J(\theta_0, \pi)$. To facilitate optimization we rewrite the function in the following form

$$\begin{aligned} J(\theta_0, \pi) &= \int \log\left(\frac{p_{\theta_0}(y,u)\exp(\rho(-\theta_0))}{\pi(u)}\right)\pi(u)du \\ &= \int \log\left(\frac{p_{\theta_0}(u|y)p_{\theta_0}(y)\exp(\rho(-\theta_0))}{\pi(u)}\right)\pi(u)du \\ &= \log p_{\theta_0}(y) - \rho(\theta_0) + \int \pi(u)\log\left(\frac{p_{\theta_0}(u|y)}{\pi(u)}\right)du \\ &= \log p_{\theta_0}(y) - \rho(\theta_0) + \text{KL}(p_{\theta_0}(\cdot|y), \pi(\cdot)) \qquad (3.25) \end{aligned}$$

where $\text{KL}(f, g) = \int g(x)\ln\frac{f(x)}{g(x)}dx$ is the Kullback-Leibler distance. Now we introduce the information inequality [28], which states that $\text{KL}(f, g) \leq 0$ with equality if and only if $f(x) = g(x)$ almost everywhere. By observing that the first two terms of (3.25) do not depend on π and using the information inequality we get

$$J(\theta_0, \pi) \leq J(\theta_0, \pi_0) = J(\theta_0, p_{\theta_0}(\cdot|y)) = \log p_{\theta_0}(y) - \rho(\theta_0). \qquad (3.26)$$

That is $\pi_0(u) = p_{\theta_0}(u|y)$ maximizes (3.25).

In the second step of the cyclic ascent algorithm, we fix $\pi_0(u)$, and optimize with respect to θ. The function to optimize is given by

$$\begin{aligned} J(\theta, \pi_0) &= \int \log\left(\frac{p_\theta(y,u)\exp(-\rho(\theta))}{\pi_0(u)}\right)\pi_0(u)du \\ &= \int \log\left(\frac{p_\theta(y,u)\exp(-\rho(\theta))}{p_{\theta_0}(u|y)}\right)p_{\theta_0}(u|y)du \\ &= \int \log\left(p_\theta(y,u)\exp(-\rho(\theta))\right)p_{\theta_0}(u|y)du - \int \log\left(p_{\theta_0}(u|y)\right)p_{\theta_0}(u|y)du \\ &= \int \log\left(p_\theta(y,u)\right)p_{\theta_0}(u|y)du - \int \rho(\theta)p_{\theta_0}(u|y)du - \int \log\left(p_{\theta_0}(u|y)\right)p_{\theta_0}(u|y)du \\ &= E_{\theta_0}[\log p_\theta(y,u)|y] - \rho(\theta) - E_{\theta_0}[\log p_{\theta_0}(y,u)|y]. \end{aligned}$$

Only the first two terms which we call the penalized EM functional depend on θ. So to carry out the cyclic ascent maximization of (3.24) we need to 1) Construct the penalized EM functional 2) Maximize the penalized EM functional. In the usual EM terminology 1) is called the E-step and 2) is called the M-step. The usual form of the EM algorithm is given by

1. The E-step involves taking a conditional expectation to construct the EM functional

$$\mathrm{EM}(\theta_0, \theta) = E_{\theta_0}[\log p_{\theta_0}(y, u)|y]$$

2. The M-step involves maximizing the penalized EM functional

$$\theta_1 = \mathrm{argmax} \mathrm{EM}(\theta_0, \theta) - \rho(\theta)$$

The cyclic ascent algorithm produces a sequence

$$J(\theta_0, \pi_0) \leq J(\theta_1, \pi_0) \leq J(\theta_1, \pi_1).$$

From this and (3.26) we get

$$\log p_{\theta_0}(y) - \rho(\theta_0) \leq \log p_{\theta_1}(y) - \rho(\theta_1)$$

so the penalized likelihood does not decrease between EM iterations. As we have seen the EM algorithm is simply a cyclic descent algorithm. So the convergence theory in Section 3.5.1 applies. Convergence issues relating to the EM algorithm are also discussed by Wu [171] and Lange [80].

3.10 svnPCA with l_0 Penalty

With the l_0 penalty the penalized negative log-likelihood takes the form

$$J_\theta(y) = -l_\theta(y) + \frac{hT}{2\sigma^2} \sum_{v=1}^{M} I(g_v \neq 0).$$

where

$$\begin{aligned} l_\theta(y) &= -\frac{T}{2\sigma^2}\mathrm{tr}S_y + \frac{T}{2\sigma^2}\mathrm{tr}(W^{-1}G^TS_yG) \\ &- \frac{T(M-r)}{2}\log\sigma^2 - \frac{T}{2}\log|W| \end{aligned} \tag{3.27}$$

$W = G^TG + \sigma^2 I_r$, and g_v is the v-th row of G. The l_0 penalty is sometimes called complexity penalty and was analyzed in relation to image denoising in [85]. This penalty is often used in wavelet analysis where it leads to hard thresholding [34]. The penalty works in the following way. If $g_v \neq 0$ then a penalty of $\frac{hT}{2\sigma^2}$ is added to the negative log-likelihood but if $g_v = 0$ no penalty is added. Another interesting feature of this penalty is that it depends on the noise variance σ^2. This simplifies the EM algorithm as we see below.

3.10.1 Estimation

We maximize the criterion with the EM algorithm. Notice, that in the estimation we do not impose the orthogonality constraint $G^TG = I_r$. First we construct, the penalized complete likelihood

$$J_\theta(y,u) = l_\theta(y,u) - \frac{hT}{2\sigma^2}\sum_{v=1}^{M} I(g_v \neq 0)$$

where the complete log-likelihood is given by

$$\begin{aligned} l_\theta(y,u) &= \sum_{t=1}^{T}\Bigg(-\frac{M}{2}\log\sigma^2 - \frac{\|y_t - m - Gu_t\|^2}{2\sigma^2} \\ &- \frac{1}{2}u_t^Tu_t\Bigg) - \frac{hT}{2\sigma^2}\sum_{v=1}^{M} I(g_v \neq 0). \end{aligned}$$

In the E-step we compute the penalized EM functional

$$\mathrm{EM}_p(\theta_0,\theta) = \mathrm{EM}(\theta_0,\theta) - \frac{hT}{2\sigma^2}\sum_{v=1}^{M} I(g_v \neq 0)$$

where the EM functional is given by

$$\begin{aligned}\mathrm{EM}(\theta_0,\theta) &= E_{\theta_0}(l_\theta(y_t,u_t)|y_t). \\ &= -\frac{1}{2\sigma^2}\mathrm{tr}(S_y)+\frac{1}{\sigma^2}\mathrm{tr}(GB_0^T)-\frac{1}{2\sigma^2}\mathrm{tr}(GA_0G^T) \\ &- \frac{M}{2}\log\sigma^2-\frac{h}{2\sigma^2}\sum_{v=1}^{M}I(g_v\neq 0)\end{aligned}$$

where

$$\begin{aligned}A_0 &= \frac{1}{T}\sum_{t=1}^{T}E_{\theta_0}[u_tu_t^T|y_t] \\ &= \sigma_0^2W_0^{-1}+W_0^{-1}G_0^TS_yG_0W_0^{-1}\end{aligned}$$

,

$$\begin{aligned}B_0 &= \frac{1}{T}\sum_{t=1}^{T}y_tE_{\theta_0}^T[u_t|y_t] \\ &= S_yG_0W_0^{-1}\end{aligned}$$

and $W_0 = G_0^TG_0+\sigma_0^2I_r$. A_0 and B_0 are derived in Appendix E.

We maximize the penalized EM functional in the M-step. First we maximize with respect to G, this is equivalent to minimizing

$$\begin{aligned}J_1(G) &= \sum_{v=1}^{M}J(g_v) \\ &= \sum_{v=1}^{M}\left(\frac{1}{2}g_v^TA_0g_v-g_v^Tb_{v,0}+\frac{h}{2}I(g_v\neq 0)\right).\end{aligned}$$

Importantly, J_1 is a separable function since it is a sum of functions of the g_v's. This means that we can optimize J_1 by optimizing J. The function J is not differentiable at zero, so to optimize J we need to compare two solution; the zero solution $g_v = 0$, and the nonzero solution g_v that minimizes J. Assume that $g_v \neq 0$ and differentiate J and set to zero. We get

$$g_v = A_0^{-1}b_{v,0}, \quad v = 1, ..., M.$$

Now compare this solution to the $g_v = 0$ solution

$$J(g_v) - J(0) = -\frac{1}{2} b_{v,0}^T A_0^{-1} b_{v,0} + \frac{h}{2} \geq 0$$

so we pick the $g_v = 0$ solution if

$$h \geq b_{v,0}^T A_0^{-1} b_{v,0}.$$

Therefore the minimizer is given by

$$g_{v,1} = A_0^{-1} b_{v,0} I(h < b_{v,0}^T A_0^{-1} b_{v,0}), \quad v = 1, ..., M.$$

Optimization of the penalized EM functional with respect to σ^2 gives

$$\sigma_1^2 \;=\; \frac{1}{M} \left[\mathrm{tr}(A_0 G_1^T G_1) - 2\mathrm{tr}(B_0^T G_1) + \mathrm{tr}(S_y) \right] + \frac{h}{M} \sum_{v=1}^{M} I(g_{v,1} \neq 0).$$

We can simplify this using the Euler equations and get

$$\begin{aligned} \sigma_1^2 &= \frac{\mathrm{tr}(S_y)}{M} - \frac{1}{M} \sum_{v=1}^{M} (b_{v,0}^T A_0^{-1} b_{v,0} - h) I(b_{v,0}^T A_0^{-1} b_{v,0} > h) \\ &= \frac{\mathrm{tr}(S_y)}{M} - \frac{1}{M} \sum_{v=1}^{M} b_{v,0} A_0^{-1} b_{v,0} I(b_{v,0}^T A_0^{-1} b_{v,0} > h) + h \frac{M_h}{M}. \end{aligned}$$

3.10.2 The EM Algorithm for svnPCA l_0

To sum up, the EM algorithm is based on performing the following steps until convergence:

Algorithm III.4 (EM algorithm for svnPCA l_0)**.**

$$\begin{aligned} W_0 &= G_0^T G_0 + \sigma_0^2 I_r \\ A_0 &= \sigma_0^2 W_0^{-1} + W_0^{-1} G_0^T S_y G_0 W_0^{-1} \\ B_0 &= S_y G_0 W_0^{-1} \\ g_{v,1} &= A_0^{-1} b_{v,0} I(b_{v,0}^T A_0^{-1} b_{v,0} > h), \quad v = 1, ..., M \\ \sigma_1^2 &= \frac{\mathrm{tr}(S_y)}{M} - \frac{1}{M} \sum_{v=1}^{M} b_{v,0} A_0^{-1} b_{v,0} I(b_{v,0}^T A_0^{-1} b_{v,0} > h) + h \frac{M_h}{M} \end{aligned}$$

3.10.3 Analysis of Stationary Points for svnPCA with l_0 Penalty

In this section, we analyze the solution obtained by the EM algorithm if it converges to a stationary point. At a stationary point ($\frac{\partial J_1(G)}{\partial G} = 0$) we have

$$S_y \hat{\Omega}^{-1} \hat{G} = \hat{G}$$

now $\hat{G}$ has only M_h non-zero rows so we cannot proceed as we did for the nPCA in Appendix A. But we can if we consider the $M_h \times r$ matrix $\tilde{G}$ which consists of the non-zero data of $\hat{G}$. Using the SVD $\tilde{G} = \tilde{P}\tilde{\Lambda}^{1/2}\tilde{R}^T$ and denote $\tilde{S}_y$ as the reduced data covariance matrix we arrive to

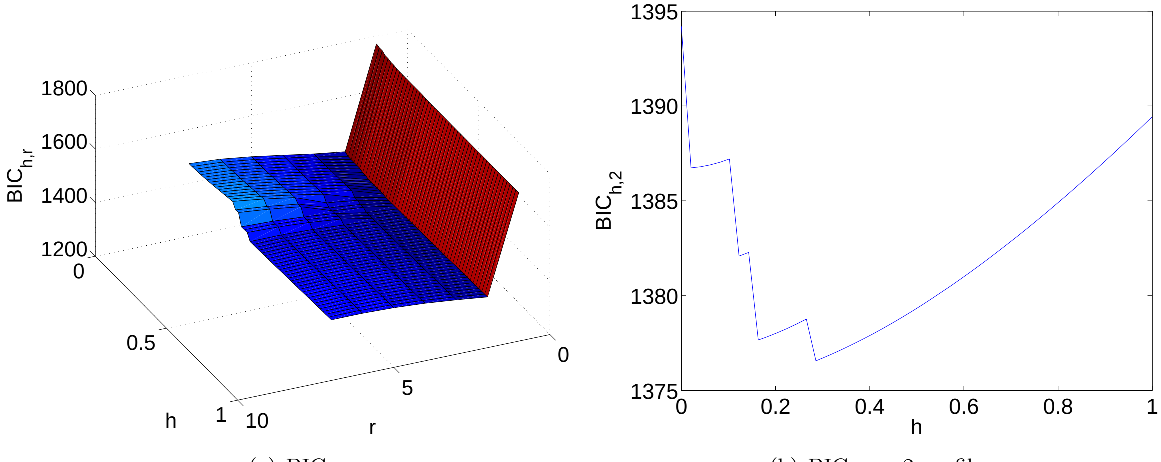
$$\tilde{S}_y \tilde{P} = \tilde{P}(\hat{\sigma}^2 I_r + \tilde{\Lambda})$$

By using the same arguments as in Appendix A we get that the unique ML solution is given by

$$\begin{aligned} \tilde{\Lambda} &= \tilde{L}_r - \hat{\sigma}^2 I_r \\ \tilde{G} &= \tilde{P}_r(\tilde{L}_r - \hat{\sigma}^2 I_r)^{1/2}\tilde{R}^T \end{aligned}$$

where $\tilde{P}_r$ and $\tilde{L}_r$ are the r largest eigenvectors/eigenvalues of the reduced data covariance matrix $\tilde{S}_y$ and $\tilde{R}$ arbitrary rotation matrix. Notice that the EM algorithm will convergence to an arbitrary value of the rotation matrix $\tilde{R}$. So at convergence we do a SVD on the resulting loading matrix, i.e., $G_1 = \hat{G}\hat{\Lambda}^{1/2}\tilde{R}^T$ and pick out $\hat{G}$ and $\hat{\Lambda}$ as our final solution.

Remark III.5 (Global Convergence for svnPCA l_0)*.* The l_0 penalty is not continuously differentiable so assumption A2 in Section 3.5.1 does not hold. Although the algorithm produces useful results a global convergence theory should be developed.



(a) BIC

(b) BIC, $r = 2$ profile

Figure 3.23: The svnPCA l_0 BIC result for the simulation in Section 3.11.1.

3.11 Results for svnPCA based on the l_0 Penalty

3.11.1 Simulation

To illustrate the svnPCA based on L0 penalty method we use the simulation setup presented in Section 3.7.2 for the $\sigma^2 = 2$ case. To compute the BIC we sampled 50 points uniformly on the interval $[0, 1]$. Figure 3.23 shows the BIC result for the simulation. If we look at Figure III.23(b) we see interesting structure. The sequence of local minimums we see for low values of h correspond to the values where the variables are zeroed out. So when a noise variable is zeroed out there is a drop in BIC and then it increases until another noise variable is zeroed out. The BIC selects

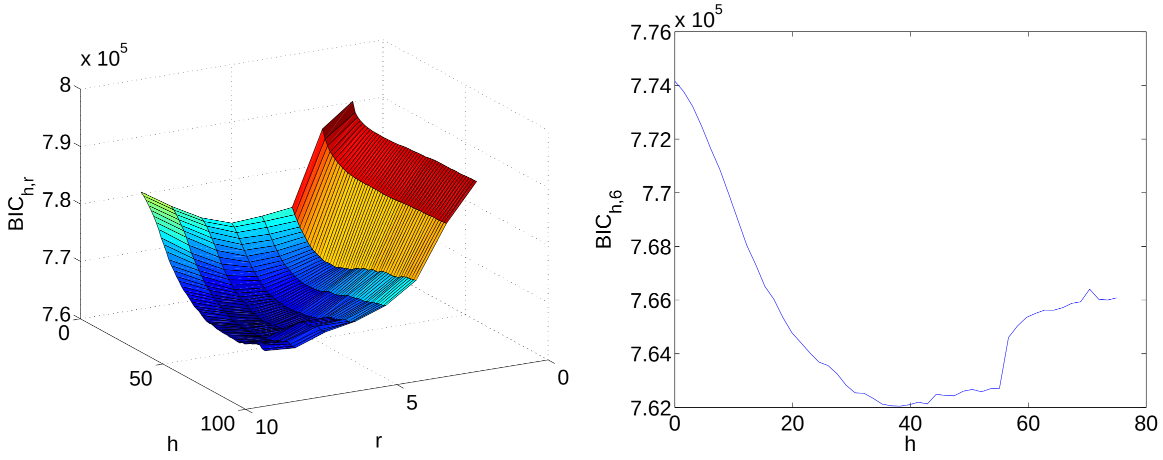
$r = 2$ and $h = 0.28$ which gives

$$\begin{aligned}
\hat{G} &= \begin{pmatrix} -0.5080 & -0.0678 \\ -0.4879 & -0.0786 \\ 0 & 0 \\ 0 & 0 \\ -0.4886 & -0.0597 \\ -0.5051 & 0.0089 \\ 0 & 0 \\ 0 & 0 \\ -0.0551 & -0.6957 \\ -0.0835 & -0.7087 \end{pmatrix} \\
\hat{\Lambda} &= \begin{pmatrix} 300.0526 & 0.1135 \\ 0.1135 & 51.7901 \end{pmatrix} \\
\hat{\sigma}^2 &= 2.1267
\end{aligned}$$

Note that the loadings for the noisy variables 3,4,7 and 8 are nulled out as expected. Of course this agrees with the simulation result presented in Section 3.7.2.

3.11.2 Real fMRI Data

Again we use the AFNI fMRI data introduced in Section 4.4.1 to illustrate our algorithm. In this case the BIC was computed for 50 values of h uniformly sampled on the interval $[0, 75]$ and for $r = 1, .., 8$. Figure 3.24 shows the BIC result for the data. The minimum BIC occurred at $r = 6$ svnPCs profile for $h = 42.9$ which corresponds to $M_h = 578$ voxels. Figure 3.25 depicts the 6 loadings (columns of G) of the svnPCs plotted spatially. Sparse variable nPC #2 has high loadings in the motor cortex clearly capturing motor activity. The other svnPCs are not so easily

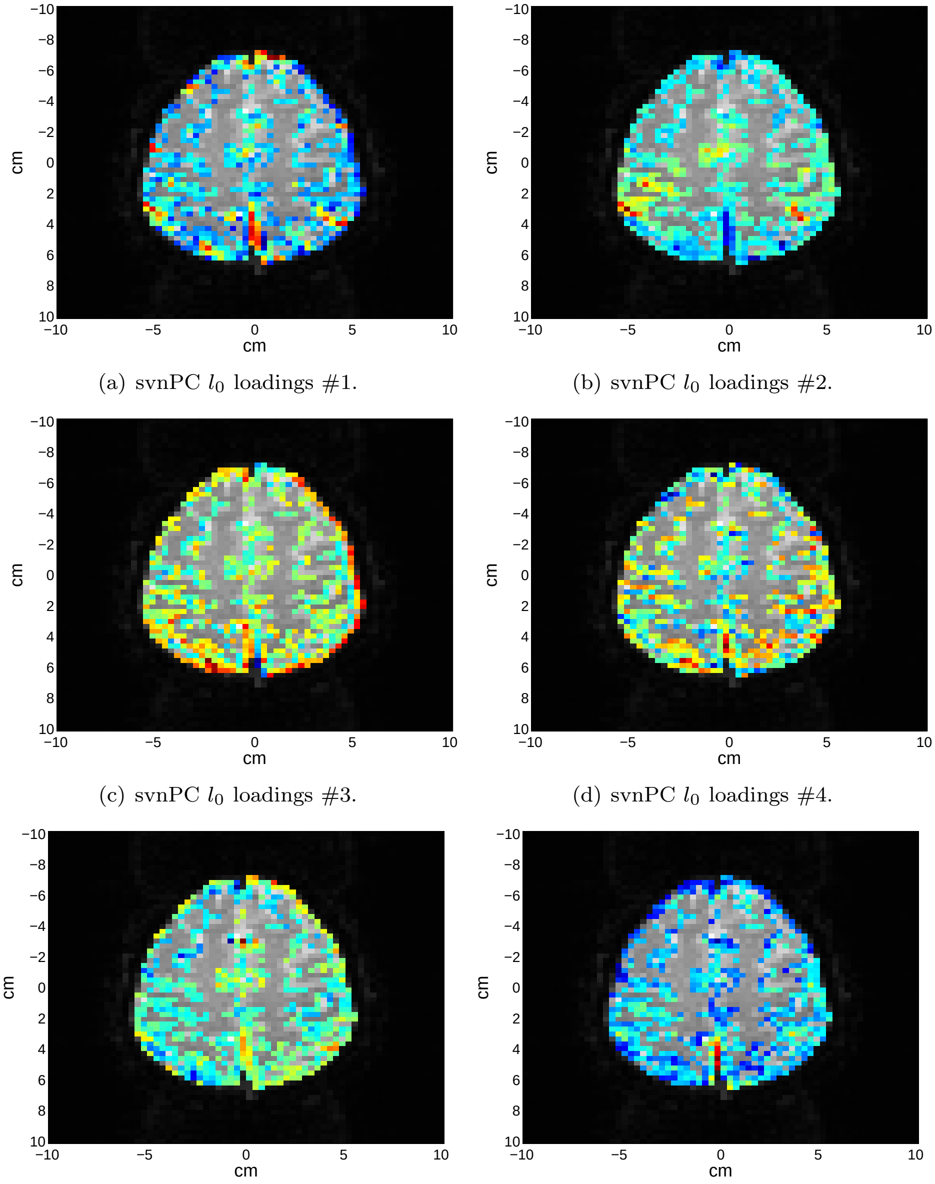


(a) BIC

(b) BIC, $r = 6$ profile

Figure 3.24: The svnPCA l_0 BIC result for the fMRI data.

identified. Figure 3.26 shows the 6 svnPCA timeseries $(u_{t,j}, t = 1, ..., T, j = 1, ..., 6)$.



(a) svnPC l_0 loadings #1.

(b) svnPC l_0 loadings #2.

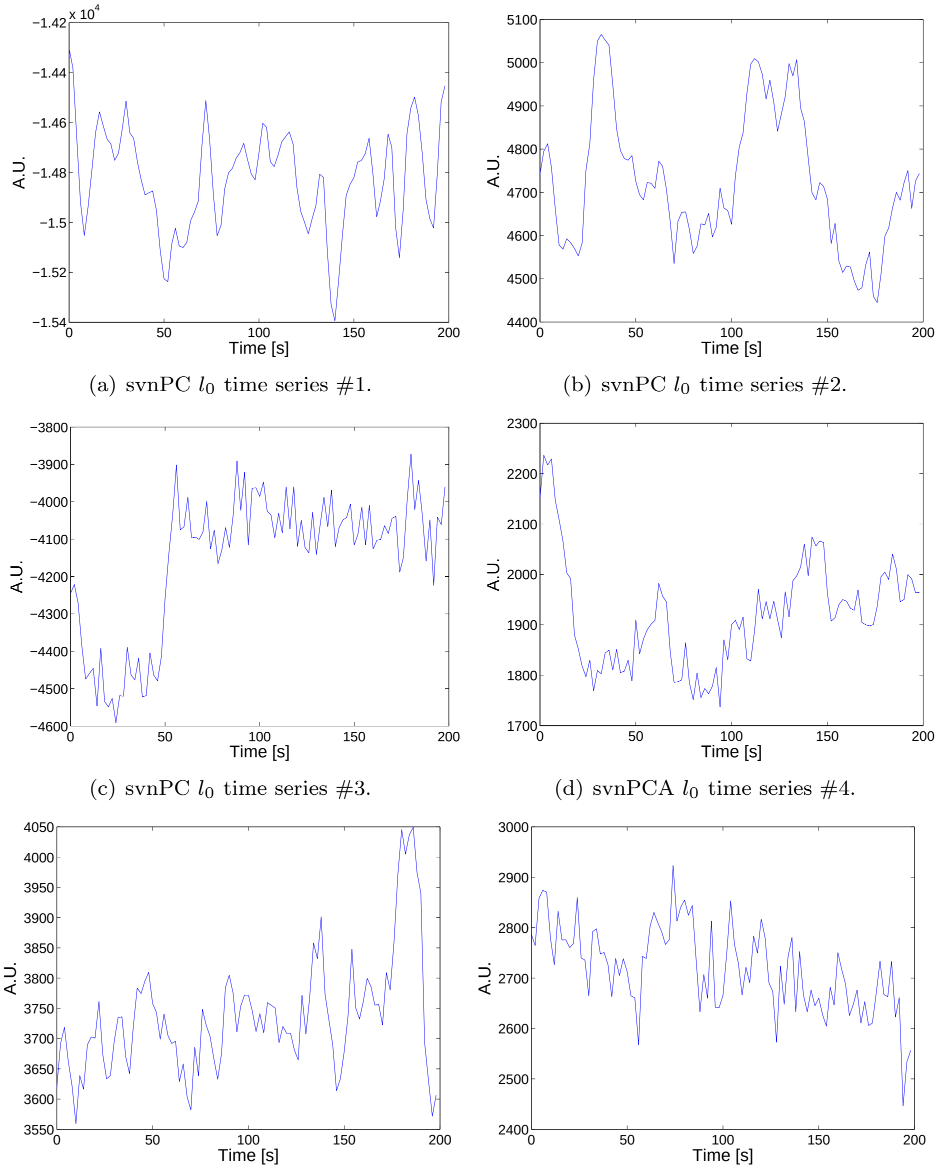
(c) svnPC l_0 loadings #3.

(d) svnPC l_0 loadings #4.

(e) svnPC l_0 loadings #5.

(f) svnPC l_0 loadings #6.

Figure 3.25: The svnPCA l_0 sparse loadings (columns of G) plotted spatially.



(a) svnPC l_0 time series #1.

(b) svnPC l_0 time series #2.

(c) svnPC l_0 time series #3.

(d) svnPCA l_0 time series #4.

(e) svnPC l_0 time series #5.

(f) svnPC l_0 time series #6.

Figure 3.26: The svnPCA l_0 timeseries $(u_{t,j})$.

CHAPTER IV

SURE AND RANDOM MATRIX THEORY FOR nPCA MODEL SELECTION

In this section, we revisit the problem of choosing the number of nPCs or equivalently the rank of the nPC loading matrix. The problem with most method reviewed in Section 1.2.2 on PCA was the need for a subjective decision from the user. The ML based methods such as AIC and BIC that are discussed in Section 1.4.2 are more objective. However, they are based on a asymptotic argument that do not hold for modern data set which are often very high dimensional with T and M of comparable sizes, i.e., low $\gamma = T/M$ ratio. Examples of such data sets can be found in Meteorology and Oceanography [115]; functional data analysis [118], financial data analysis and of course fMRI.

Very few examples exist about selection methods specifically designed to deal with such data sets for nPCA. Minka [101] develops a Bayesian model selection method, which we call the Laplace method, based on the Laplace approximation, and shows that it performs very well in cases where T and M are of comparable sizes. Furthermore, in simulations, it compares well against cross-validation, and other similar Bayesian methods [117, 16, 42]. Beckmann et al. [12] employs the Laplace criterion for fMRI data, and in addition he makes use of RMT to modify it. It is worth noting that Ferre [44] develops a method for comparable T and M for the deterministic PCA model (see Section 1.4.4), and compares it with other methods in [45]. Another interesting paper is by Hoff [65]. However, his method is developed for a model different from nPCA. In addition, this method is very computationally intensive.

In this chapter, we propose to use SURE [142] to choose the nPCs. SURE was originally not designed for model selection, but following [35, 67] Solo realized [134] that it could be used as a general purpose tool for tuning parameter selection in non-linear ill-conditioned inverse problems. Applications of SURE for model selection include: [35] for choosing the threshold in wavelet estimation; [136] to choose when to stop the iteration in anisotropic diffusion signal reconstruction; [106] for choosing the neighborhood size in optical flow estimation; [135] for selection of regularization parameters for total variation de-noising; [131] for selection of smoothing parameter in optical flow estimation; and [137] for selection of tuning parameter for support vector machines.

The advantages of our SURE based selection method are: 1) it is computationally simple, i.e., does not require much more computation than that needed to obtain the PCs, 2) it has an unbiasedness property even for non-linear problems 3) it is exact, i.e., no approximations are needed. To implement SURE in practice it is necessary to estimate a noise variance, and we develop a novel method based on RMT to do that. Part of the material in this chapter has been submitted for publication [159].

4.1 Model Selection for nPCA Revisited

In this section, we first review the Laplace method, and then introduce our SURE method. We chose the Laplace method since it is the only method we know of that shows good performance in low sample framework. The BIC is chosen as a surrogate for all methods that assume a large sample framework. We will work with the temporal nPCA model given by Equation (1.9) to derive our methods.

4.1.1 The Laplace Method

The Laplace method is derived from a Bayesian framework and is based on maximizing the evidence that the nPCA model consists of r PCs. Minka [101] used approximated the evidence using the Laplace method yielding (using the notation from Section 1.4.1, except now $\hat{\sigma}^2 = \hat{\sigma}_r^2$)

$$\begin{aligned} -\log p(y|r) &= -l_{\hat{\theta}}(y) - \log p(P) - \frac{\dim(\theta) - M - 1}{2} \log 2\pi \\ &+ \frac{1}{2} \log |A_z| + \frac{r}{2} \log T \end{aligned}$$

where

$$\begin{aligned} \dim(\theta) &= Mr - r(r-1)/2 + 1 + M \\ p(P) &= 2^{-r} \prod_{i=1}^{r} \Gamma((M-i+1)/2) \pi^{-(M-i+1)/2} \\ |A_z| &= T \prod_{i=1}^{r} \prod_{j=i+1}^{M} (\tilde{l}_j^{-1} - \tilde{l}_i^{-1})(l_i - l_j) \end{aligned}$$

where $p(P)$ is a noninformative prior distribution for P, $\tilde{l}_j$ is equal to l_j when $j \leq r$ and equal to $\hat{\sigma}_r^2$ when $j > r$, and

$$-l_{\hat{\theta}}(y) = \frac{T}{2} \sum_{j=1}^{r} \log l_j + \frac{T}{2}(M-r) \log \hat{\sigma}_r^2.$$

The r that minimizes $-\log p(y|r)$ is picked as the number of nPCs.

4.1.2 The SURE Method

SURE is based on the following considerations. Ideally, we would like to choose the value of r that minimizes the risk, (see Equation (1.13) for definition of $\hat{\mu}$)

$$R_r = E\|\mu - \hat{\mu}\|^2.$$

We generally do not know the true signal, so we cannot compute the risk. But the idea is to try to find a computable unbiased estimator of it and minimize that instead. Indeed, remarkably Stein [142] showed how to construct such an estimator under Gaussian assumptions. For completeness, we have provided an "engineering" derivation of Stein's result in Appendix C (Stein's original derivation is not intuitive). SURE is given by

$$\hat{R}_r = \frac{1}{T}\sum_{t=1}^{T} \|n_t\|^2 + 2\sigma^2 \frac{1}{T}\sum_{t=1}^{T} \mathrm{tr}(\frac{\partial \hat{\mu}_t}{\partial y_t^T}) - M\sigma^2 \tag{4.1}$$

where our estimator of μ is given by (assuming $\bar{y} = 0$)

$$\hat{\mu}_t = \hat{G}\hat{u}_t = \sum_{j=1}^{r} p_{(j)} \frac{l_j - \hat{\sigma}_r^2}{l_j} p_{(j)}^T y_t \tag{4.2}$$

and $n_t = y_t - \hat{\mu}_t$. The idea behind SURE as a tuning parameter selector is that [134] since it is an unbiased estimator of the risk then on average one can hope that its minimizer is an unbiased estimator of the minimizer of the risk.

In order to compute SURE, the main task is to compute the derivative of the signal estimate with respect to the data

$$\begin{aligned} \frac{\partial \hat{\mu}_t}{\partial y_t^T} &= \sum_{j=1}^{r} \Bigg(\frac{\partial p_{(j)}}{\partial y_t^T} \frac{l_j - \hat{\sigma}_r^2}{l_j} p_{(j)}^T y_t \\ &+ p_{(j)} \frac{\partial}{\partial y_t^T} (\frac{l_j - \hat{\sigma}_r^2}{l_j}) p_{(j)}^T y_t \\ &+ p_{(j)} \frac{l_j - \hat{\sigma}_r^2}{l_j} p_{(j)}^T \\ &+ p_{(j)} \frac{l_j - \hat{\sigma}_r^2}{l_j} y_t^T \frac{\partial p_{(j)}}{\partial y_t^T} \Bigg). \end{aligned} \tag{4.3}$$

To compute these derivatives we make use of the following theorem from [91].

Theorem IV.1. *Let S_y be a real symmetric $M \times M$ matrix. Let $p_{(j)}$ be a normalized eigenvector associated with a simple eigenvalue l_j of S_y. Then the differentials of $p_{(j)}$*

and l_j at S_y are given by

$$\begin{aligned} dp_{(j)} &= (\lambda_j I_M - S_y)^+ (dS_y) p_{(j)} && (4.4) \\ dl_j &= p_{(j)}^T (dS_y) p_{(j)} && (4.5) \end{aligned}$$

respectively.

For subsequent calculations we rewrite (4.4) as:

$$\begin{aligned} dp_{(j)} &= (l_j I_M - S_y)^+ dS_y p_{(j)} \\ &= (P(l_j I_M - L)P^T)^+ dS_y p_{(j)} \\ &= \sum_{i \neq j} p_i (l_j - l_i)^{-1} p_i^T dS_y p_j. && (4.6) \end{aligned}$$

To proceed, we need to express perturbation of S_y induced by perturbation of $y_{t,v}$. Let $\tilde{d}$ denote perturbation induced by perturbation in $y_{t,v}$:

$$\begin{aligned} \tilde{d}S_y &= \frac{1}{T} \tilde{d} Y^T Y \\ &= \frac{1}{T} (\tilde{d}(Y^T) Y + Y^T \tilde{d} Y) \\ &= \frac{1}{T} (e_v y_t^T + y_t e_v^T) dy_{t,v}. && (4.7) \end{aligned}$$

We see that perturbation in $y_{t,v}$ induces perturbation in row v and column v of S_y. Equations (4.6) and (4.7) yield:

$$\frac{\partial p_{(j)}}{\partial y_t^T} = \frac{1}{T} \sum_{i \neq j} p_{(i)} (l_j - l_i)^{-1} (y_t^T p_{(j)} p_{(i)}^T + y_t^T p_{(i)} p_{(j)}^T). \qquad (4.8)$$

We also need $\frac{\partial l_j}{\partial y_t^T}$. Starting from (4.5):

$$\begin{aligned} \tilde{d} l_j &= p_{(j)}^T \tilde{d} S_y p_{(j)} \\ &= \frac{1}{T} p_{(j)}^T (\tilde{d}(Y^T) Y + Y^T \tilde{d} Y) p_{(j)} \\ &= \frac{2}{T} p_{(j)}^T y_t p_{v,j} dy_{t,v}. \end{aligned}$$

From this we get

$$\frac{\partial l_j}{\partial y_t^T} = \frac{2}{T}(p_{(j)}^T y_t) p_{(j)}^T. \tag{4.9}$$

Now we use (4.8) and (4.9) and compute the trace of the terms in (4.3) one by one. The trace of the first term, where $d_j = \frac{l_j - \hat{\sigma}_r^2}{l_j}$, is given by

$$\begin{aligned} \operatorname{tr}(\sum_{j=1}^{r} \frac{\partial p_{(j)}}{\partial y_t^T} d_j p_{(j)}^T y_t) &= \frac{1}{T}\sum_{j=1}^{r}\sum_{i \neq j} d_j (l_j - l_i)^{-1} (p_{(j)}^T y_t)^2 \\ &= \sum_{j=1}^{r}\sum_{i \neq j} (l_j - \sigma_r^2)(l_j - l_i)^{-1} q_{t,j}^2 \end{aligned}$$

where we have used the singular value decomposition $Y = \frac{1}{\sqrt{T}} Q L^{-1/2} P^T$. The trace of the second term is given by

$$\begin{aligned} \operatorname{tr}\left(\sum_{j=1}^{r} p_{(j)} \frac{\partial}{\partial y_t^T} d_j p_{(j)}^T y_t\right) &= \frac{2}{T}\sum_{j=1}^{r} \frac{1}{\lambda_j}(1 - d_j)(p_{(j)}^T y_t)^2 \\ &= 2\sum_{j=1}^{r} \frac{\hat{\sigma}_r^2}{l_j} q_{t,j}^2 \end{aligned}$$

The trace of the third term is given by

$$\operatorname{tr}(\sum_{j=1}^{r} p_{(j)} \frac{l_j - \hat{\sigma}_r^2}{l_j} p_{(j)}^T) = \sum_{j=1}^{r} \frac{l_j - \hat{\sigma}_r^2}{l_j}.$$

The trace of the fourth term is given by

$$\begin{aligned} \operatorname{tr}\left(\sum_{j=1}^{r} p_{(j)} d_j y_t^T \frac{\partial p_{(j)}}{\partial y_t^T}\right) &= \frac{1}{T}\sum_{j=1}^{r} d_j (l_j - l_i)^{-1} (y_t^T p_{(i)})^2 \\ &= \sum_{j=1}^{r}\sum_{i \neq j} d_j (l_j - l_i)^{-1} l_i q_{t,i}^2. \end{aligned}$$

Finally we note that using (4.2) we can write

$$\frac{1}{T}\sum_{t=1}^{T} \|n_t\|^2 = (M - r)\hat{\sigma}_r^2 + \sum_{j=1}^{r} \frac{\hat{\sigma}_r^4}{l_j} \tag{4.10}$$

Putting all these expressions into the SURE formula (4.1) and dropping a constant term that does not depend on r yields

$$\begin{aligned}\hat{R}_r &= (M-r)\hat{\sigma}_r^2 + \hat{\sigma}_r^4 \sum_{j=1}^{r} \frac{1}{l_j} + 2\sigma^2 r \\ &- 2\sigma^2\hat{\sigma}_r^2 \sum_{j=1}^{r} \frac{1}{l_j} + \frac{4\sigma^2\hat{\sigma}_r^2}{T} \sum_{j=1}^{r} \frac{1}{l_j} + C \qquad (4.11) \\ C &= \frac{2\sigma^2}{T} \sum_{j=1}^{r} (1 - \frac{\hat{\sigma}_r^2}{l_j}) \sum_{i \neq j} \frac{l_j + l_i}{l_j - l_i}.\end{aligned}$$

We can simplify further the last term of the expression, which we call the interaction term (see Appendix D).

$$\begin{aligned}C &= \frac{4\sigma^2}{T} \sum_{j=1}^{r} \sum_{i=r+1}^{M} \frac{l_j - \hat{\sigma}_r^2}{l_j - l_i} + \frac{2\sigma^2}{T} r(r-1) \\ &- \frac{2\sigma^2}{T}(M-1) \sum_{j=1}^{r} (1 - \frac{\hat{\sigma}_r^2}{l_j}).\end{aligned}$$

It would be nice to obtain distributional properties of SURE but that would require limiting results on bivariate functions of sample eigenvalues. So far only results for univariate functions are available [8].

The noise variance σ^2 is assumed known in the SURE formula. A natural choice for it is $\hat{\sigma}_r^2$ but that does not work well in practice. And finding a reliable estimator turns out to be a non-trivial issue which we now pursue.

4.2 Estimation of σ^2 via Random matrix Theory

We seek an estimator of σ^2 that does not require a good estimate of r. Our idea is to use the tail end of the eigenvalue spectrum to estimate σ^2. We do this by 'flattening' the Empirical Cumulative Distribution (ECD) of the sample eigenvalues by using RMT.

4.2.1 Random Matrix Theory

RMT is defined by a scenario in which $T \to \infty$, $M \to \infty$ while $T/M = \gamma \neq 0, \gamma < \infty$. This differs from the classical PCA asymptotic where M is fixed and $T \to \infty$ [5]. In this RMT case, the eigenvalues of the sample covariance matrix do not converge in probability to the true values. Rather the empirical distribution converges to a limit called the Marchenko-Pastur (MP) distribution and is described in a seminal paper [93].

Theorem IV.2. *Given a $T \times M$ data matrix Y, with independent zero mean and unit variance entries, and $T, M \to \infty$, such that $T/M \to \gamma \geq 1$. Then the ECD function is given by*

$$\hat{F}_\gamma(x) = \frac{1}{M} \sum_{i=1}^{M} I(l_{M-i+1} \leq x) \tag{4.12}$$

of the eigenvalues associated with the covariance matrix S_y converges almost surely to the MP distribution

$$\hat{F}_\gamma(x) \to F_\gamma(x)$$

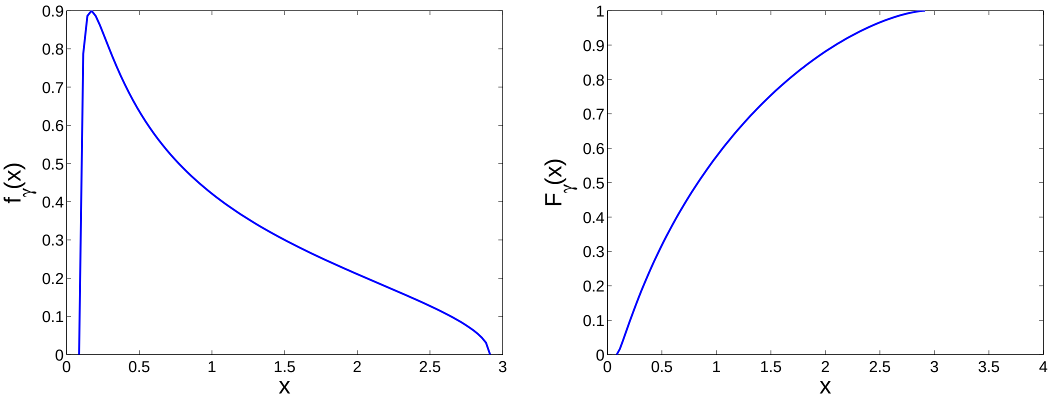
where [115],p205 (with a correction of a typographical error)

$$\begin{aligned} F_\gamma(x) &= \frac{1}{2} + \frac{\gamma}{2\pi}\left[\sqrt{(x-a)(b-x)}\right. \\ &+ \sqrt{ab}\,\text{arcsin}\left(\frac{(a+b)x - 2ab}{x(a-b)}\right) \\ &- \left.\frac{1}{2}(a+b)\,\text{arcsin}\left(\frac{a+b-2x}{b-a}\right)\right], a \leq x \leq b \end{aligned}$$

and the associated MP density is given by [73, 93]

$$f_\gamma(x) = F'_\gamma(x) = \frac{\gamma}{2\pi x}\sqrt{(b-x)(x-a)}, \quad a \leq x \leq b$$

where $a = (1 - \gamma^{-1/2})^2$ and $b = (1 + \gamma^{-1/2})^2$.



(a) MP density

(b) MP distribution

Figure 4.1: The MP density and distribution for $\gamma = 2$.

Figure 4.1 show the MP density and distribution for $\gamma = 2$.

If $\gamma < 1$ the above formulas have to be modified. In that case $\gamma' = \gamma^{-1}$ takes place of γ in the above formulation and there will be some probability mass at zero. An explicit formulation for that case is given in [132].

Although this result is stated for the case where T and M go to infinity in fixed ratio there is empirical evidence [73, 42] and [115],p.237-252 that it is a good approximation for very low values of T and M, e.g., $T = M = 10$. For the case of non-unit noise variance σ^2, the distribution simply scales up and is given by $F_\gamma(\sigma^2 x)$ [133].

The case of nPCA where there are few leading signal eigenvalues and many equal valued noise eigenvalues is called the spiked model [73], i.e., noise model spiked with few significant eigenvalues. It has long been known that the MP result still holds [93] for this case, except that there may be some eigenvalues outside the MP support. Two recent papers [9, 112] give theoretical discussion of the asymptotic behavior of those eigenvalues. We present a simplified version of a theorem proved in [9]:

Theorem IV.3. *Given a $T \times M$ data matrix Y, with independent zero mean and unit variance entries, and $T, M \to \infty$ with $\frac{T}{M} \to \gamma > 1$. Further assume that the* *population eigenvalues of the associated covariance matrix are given by* $\alpha_1 > \alpha_2 > \alpha_r > 1$*; and* $\alpha_i = 1$*,* $r < i \leq M$*. Let* r_0 *be the number of* j*'s such that* $\alpha_j > 1+\gamma^{-1/2}$*. Then for each* $1 \geq j \geq r_0$

$$\begin{aligned} l_j &\to \alpha_j + \frac{\gamma^{-1}\alpha_j}{\alpha_j - 1}, \quad a.s. \\ l_{M_0+1} &\to (1+\gamma^{-1/2})^2, \quad a.s. \end{aligned}$$

Theorem 2 applies for the $r_0 - r$ *eigenvalues.*

Similar version of this theorems for the case where $\gamma < 1$ and $\gamma = 1$ are given in [9]. From this theorem we see that signal eigenvalues behave differently when close to the noise eigenvalues than when far away. This fact is called the phase transition phenomenon.

4.2.2 RMT Noise Variance σ^2 Estimation Method

The spiked model and the scaling property of the MP distribution lead to the following idea for estimating the noise variance.

Algorithm IV.4 (RMT noise variance estimation)**.** Given a sample covariance matrix S_y.

1. Compute the eigenvalues $l_1 > l_2 >, ..., > l_M$ of S_y.

2. Compute the corrected eigenvalues

$$\tilde{l}_j^{(1)} = \frac{l_j}{F_\gamma^{-1}(\hat{F}_\gamma(l_j))} = \frac{l_j}{F_\gamma^{-1}(\frac{M-j+1}{M})}, \quad j = 1, ..., M$$

   where F_γ^{-1} is the quantile function associated with F_γ, and $\hat{F}_\gamma$ is the ECD function (4.12). We expect that $\hat{F}_\gamma(l_j) \approx F_\gamma(\frac{l_j}{\sigma^2})$ so $\tilde{l}_j^{(1)} \approx \sigma^2$.

3. A rough estimate of σ^2 is then given by

$$\tilde{\sigma}^2_{RMT} = \text{ 25th percentile of } \tilde{l}^{(1)}.$$

4. Normalize eigenvalues

$$\tilde{l}_j = \frac{l_j}{\tilde{\sigma}^2_{RMT}}, \quad j = 1, ..., M.$$

5. Get a crude estimate of the number of signal eigenvalues using the upper support limit of the MP density

$$r = \text{argmin}(\tilde{l}^{(1)}_j - b), \ \tilde{l}^{(1)}_j - b > 0.$$

6. Construct the ECD function for the noise eigenvalues such that

$$\tilde{F}_\gamma(x) = \frac{1}{M-r} \sum_{i=M-r}^{M} I(l_{M-i+1} \leq x).$$

7. Recompute the corrected eigenvalue

$$\tilde{l}^{(2)}_j = \frac{l_j}{F^{-1}_\gamma(\tilde{F}_\gamma(l_j))} = \frac{l_j}{F^{-1}_\gamma(\frac{M-j+1}{M-r})}, j = M-r, ..., M.$$

8. The final estimate of the noise variance is given by

$$\hat{\sigma}^2_{RMT} = \text{ 25th percentile of } \tilde{l}^{(2)}.$$

   Note that we are implicitly assuming that the 25th percentile of $\tilde{l}$ is a noise eigenvalue. An inspection of the Scree plot is recommended.

Now we make few remarks about the algorithm:

Remark IV.5. Notice that this noise variance estimation method also includes a crude estimate for r. This method is far from competitive with SURE but helps provide a good estimate for σ^2.

Remark IV.6. The essence of the algorithm is captured in steps 1-3, steps 4-8 refine the noise variance estimate obtained in step 3.

Remark IV.7. In steps 3 and 8 we choose the 25th percentile of the corrected eigenvalues as an estimate for the noise variance. There is nothing special about the 25th percentile, we could have chosen the 30th percentile or the median instead, like Figure 4.3 below illustrates.

4.3 Simulation Result

In this section, we present a simulation study, where we compare the SURE with the Laplace method, BIC. The BIC and Laplace methods were implemented using formulas from Sections 4.1.1 and 1.4.2. We simulated the data according to Equation (1.9), with the following parameters:

- $M = 64$
- $T = [64, 96, 128, 160]$
- $\lambda = l - \sigma^2 = [(r+1)^2, r^2, ..., 3^2, \lambda_r]$
- $\lambda_r = [1.5, 2]$
- $r = [5, 10, 15, 30]$
- $\sigma^2 = 1.$

So for each method this gives rise to $5 \times 2 \times 4 = 40$ different simulation settings.

The loading matrix was simulated by generating $M \times r$ matrix of unit variance Gaussian random variables. It was then orthonormalized. All simulations were repeated $N_{rep} = 1500$ times, and it was recorded how many times each method choose the correct dimensionality.

First we compare the performance of the RMT noise estimation method to the ML estimator (1.15) for the noise variance. Table 4.1 shows bias, variance and Mean

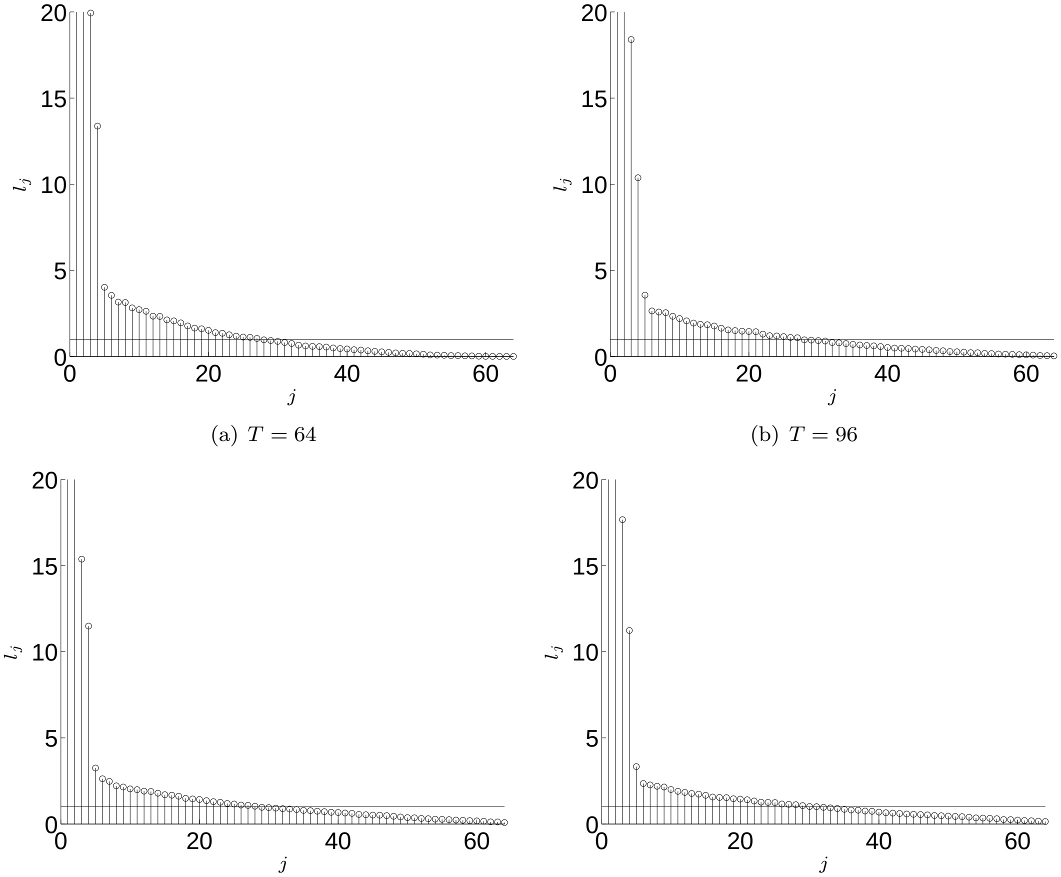
Table 4.1: Comparison of noise variance estimators for the $\lambda_r = 2$ case

$T = 64$	RMT method			ML method		
r	bias	variance	MSE	bias	variance	MSE
5	0.0278	0.0024	0.0032	-0.1062	0.0005	0.0118
10	-0.0357	0.0019	0.0032	-0.1850	0.0005	0.0347
15	-0.1067	0.0018	0.0132	-0.2648	0.0004	0.0705
30	-0.2478	0.0027	0.0640	-0.4988	0.0004	0.2493
$T = 96$	bias	variance	MSE	bias	variance	MSE
5	-0.0023	0.0013	0.0013	-0.0708	0.0003	0.0053
10	-0.0149	0.0010	0.0012	-0.1227	0.0003	0.0154
15	-0.0394	0.0008	0.0023	-0.1762	0.0004	0.0314
30	-0.1216	0.0011	0.0156	-0.3331	0.0004	0.1114
$T = 128$	bias	variance	MSE	bias	variance	MSE
5	0.0054	0.0007	0.0007	-0.0528	0.0002	0.0030
10	0.0005	0.0007	0.0007	-0.0921	0.0003	0.0087
15	-0.0081	0.0006	0.0007	-0.1312	0.0003	0.0175
30	-0.0585	0.0006	0.0041	-0.2489	0.0003	0.0622
$T = 160$	bias	variance	MSE	bias	variance	MSE
5	0.01041	0.0005	0.0006	-0.0418	0.0002	0.0020
10	0.0040	0.0004	0.0005	-0.0743	0.0002	0.0057
15	0.0000	0.0005	0.0005	-0.1054	0.0002	0.0114
30	-0.0281	0.0005	0.0013	-0.2000	0.0003	0.0402

Square Error (MSE) for both methods for the case of $\lambda_r = 2$. The case of $\lambda_r = 1.5$ is not presented here since the results are very similar. In terms of MSE, the RMT estimator shows superior performance over the ML estimator. The RMT estimator has higher variance, but much lower bias in all cases.

Figure 4.2 shows sample Scree plots from one of the replicates with $r = 5$ and $\lambda_r = 2$, the noise variance level σ^2 is indicated by a horizontal line. It can be seen that for this simulation setting it is sometimes rather easy for the eye to determine the correct dimensionality, e.g., Figure 4.2(b,d). But sometimes rather hard Figure 4.2(a,c).

Figure 4.3 shows the corrected Scree plots. The noise eigenvalues fall nicely on the horizontal noise variance line, so almost any of them could be used as an estimate for σ^2.



(a) $T = 64$

(b) $T = 96$

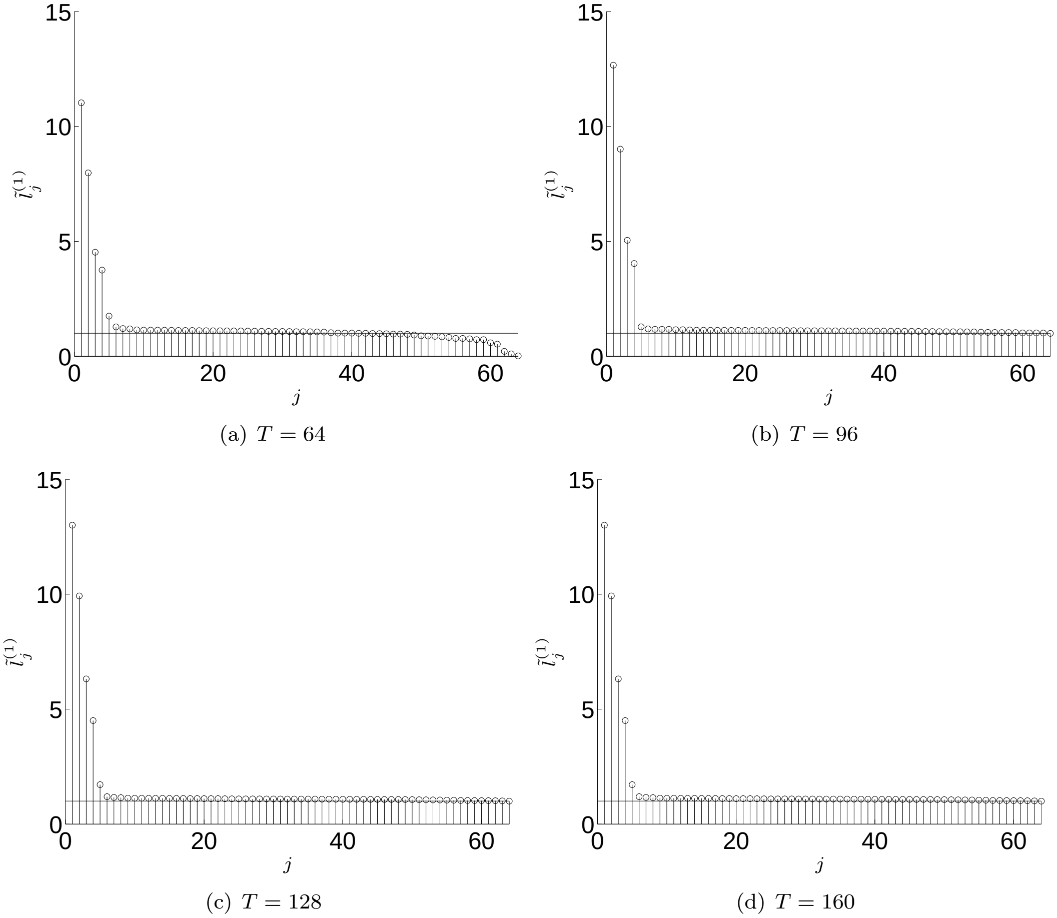
(c) $T = 128$

(d) $T = 160$

Figure 4.2: Simulation: Sample Scree plots with $r = 5$ and $\lambda_r = 2$.

Table 4.2 shows the percentage of correct selection of PCs for SURE, Laplace, and BIC methods for the case of $\lambda_r = 1.5$. The bold face entries indicate which method performs best according to the 95% significant level. It can be seen that SURE performs best in almost all of the cases, sometimes by a wide margin. Even in the cases where SURE does not perform best it is a close contender. BIC does not perform well in any of the cases presented, which is not surprising since BIC is based on an asymptotic argument which does not hold. Especially interesting is how well SURE performs in low T and M cases.

Table 4.3 shows the percentage of correct selection of PCs for SURE, Laplace, and BIC methods for the case of $\lambda_r = 2$. Basically, the same conclusions can be



(a) $T = 64$

(b) $T = 96$

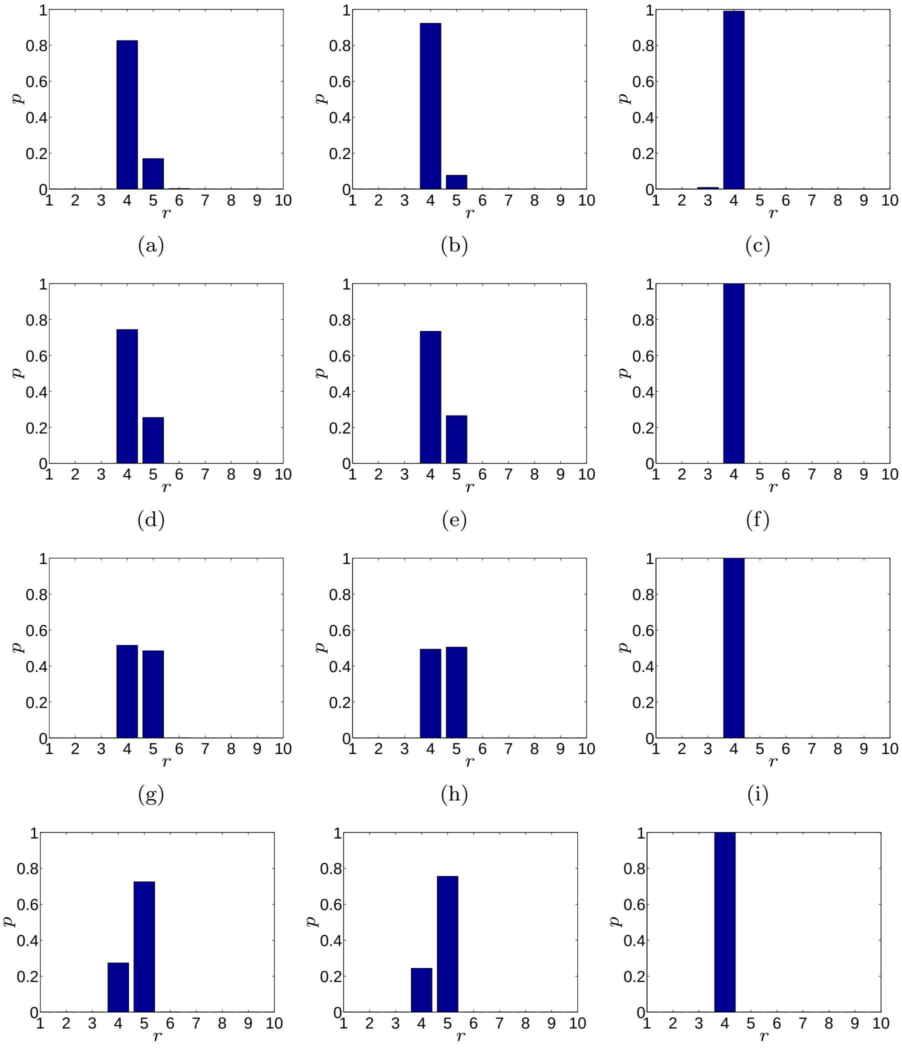
(c) $T = 128$

(d) $T = 160$

Figure 4.3: Simulation. Corrected Scree Plot (Corresponding to Figure 4.2)

drawn as for the case of $\lambda_r = 1.5$ except the results are a little better for all the methods due to the higher SNR. Note that in both Tables 4.2 and 4.3, include the selection result for the true risk. As expected it generally works well, but somewhat surprisingly SURE outperforms it sometimes. The reason for this is not known, but a possible explanation is that SURE has access to the RMT noise variance estimator while risk has not.

Finally, Figure 4.9-4.11 shows a histogram of the number of PC chosen by the considered methods for $r = 5, 10, 15, 30$, $\lambda_r = 2$, and $T = 64, 96, 128, 160$. And 4.5-4.7 shows histograms for the $\lambda_r = 1.5$ case.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

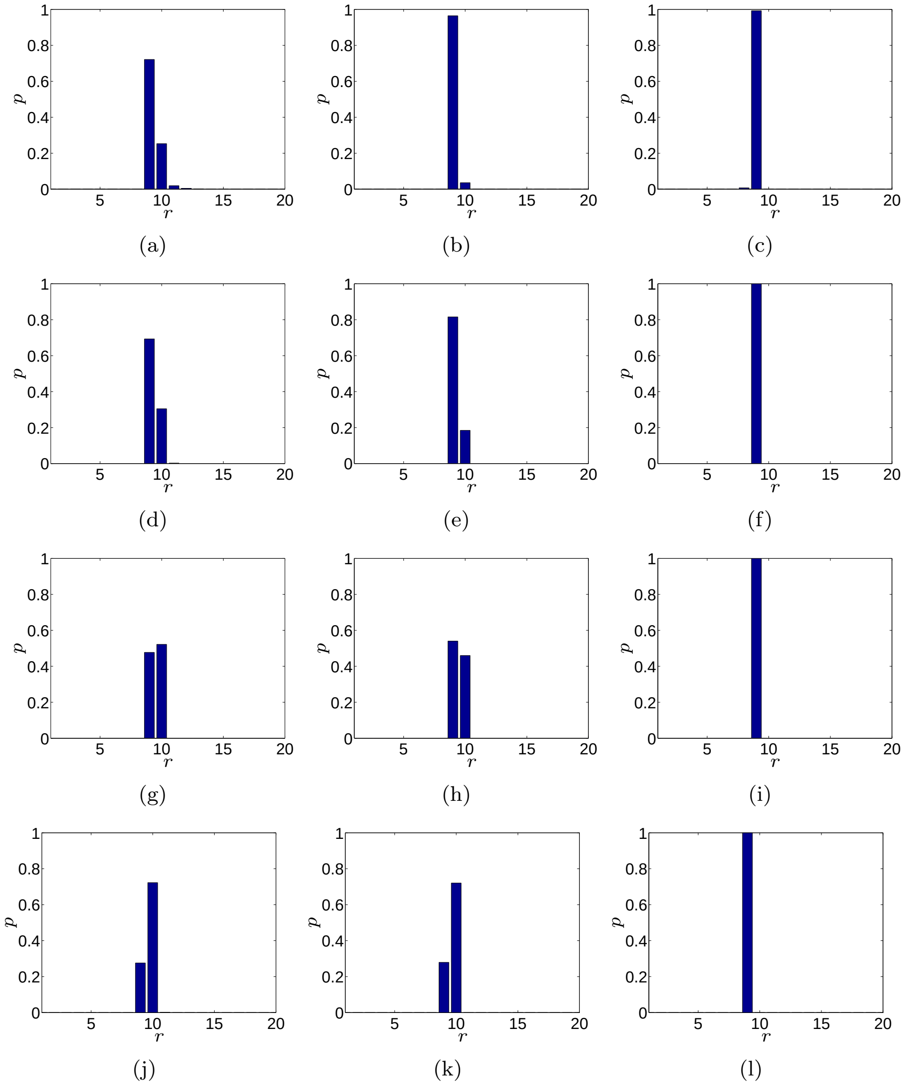
Figure 4.4: Simulation: Histogram of number of nPCs, where $r = 5$ and $\lambda_r = 1.5$. First column is SURE, second is Laplace, and third is BIC. The rows represent $T = 64, 96, 128, 160$.

4.4 Real Data Results

In this section, we apply the SURE criterion on a high dimensional fMRI data set.

4.4.1 The fMRI Data Set

Selection of the number of PCs is very important in fMRI analysis, both for dimensionality reduction [63], and as a preprocessing step for further analysis.The



(a) (b) (c)

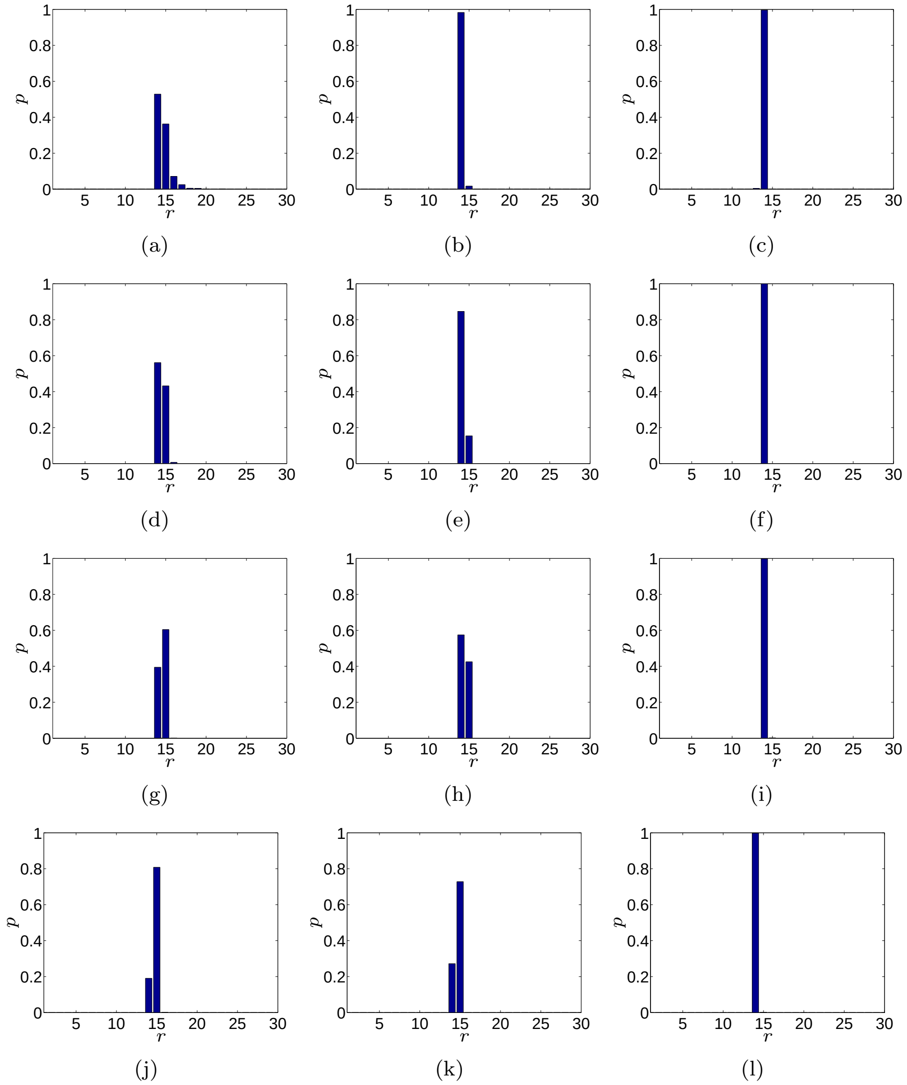
(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.5: Simulation: Histogram of number of nPCs, where $r = 10$ and $\lambda_r = 1.5$. First column is SURE, second is Laplace, and third is BIC. The rows represent $T = 64, 96, 128, 160$.

data set, which we will refer to as the AFNI data set is freely available on the Internet [29], comes from a combined visual-motor experiment. A human subject performed finger-thumb opposition, and looked at visual patterns, while being scanned. One hundred $T = 100$ observations on 21 brain slices were recorded with TR=2s sampling interval at 3 Tesla using Echo Planar Imaging (EPI). The stimuli in this experiment are, right hand finger thumb opposition, annular 8Hz checkerboard, and anti-annular 8Hz checkerboard. Figure 4.12 depicts the stimulus signals.



(a) (b) (c)

(d) (e) (f)

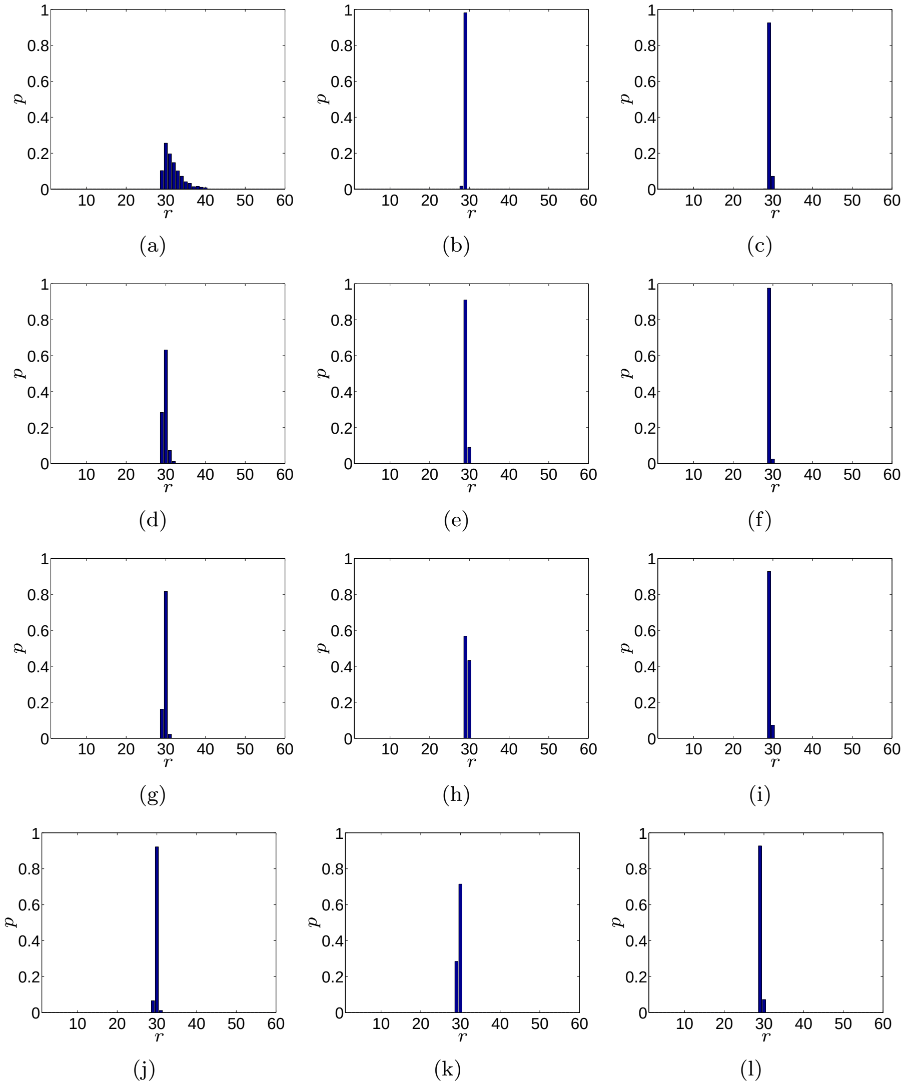
(g) (h) (i)

(j) (k) (l)

Figure 4.6: Simulation: Histogram of number of nPCs, where $r = 15$ and $\lambda_r = 2$. First column is SURE, second is Laplace, and third is BIC. The rows represent $T = 64, 96, 128, 160$.

Since the fMRI data set is a time series of brain images it is not obvious whether to let the brain images or the pixels play the role of independent observations. We call the former approach temporal nPCA and the second approach spatial nPCA. In this section we choose the temporal nPCA approach.

Figure 4.13 shows both the Scree plot and the corrected Scree plot for the fMRI brain slice number 5. The correction clearly flattens out the noise tail validating our noise variance estimation scheme. The two last eigenvalues are zero, the reason for



(a) (b) (c)

(d) (e) (f)

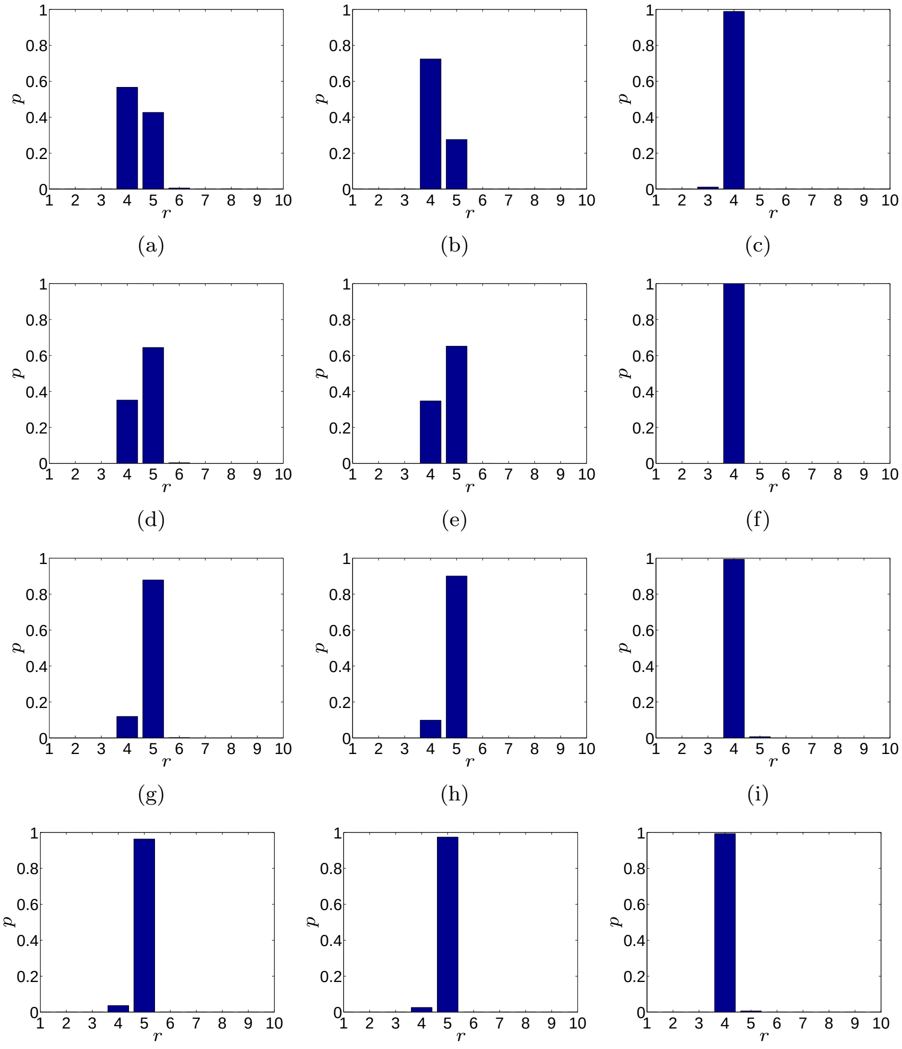
(g) (h) (i)

(j) (k) (l)

Figure 4.7: Simulation: Histogram of number of nPCs, where $r = 30$ and $\lambda_r = 1.5$. First column is SURE, second is Laplace, and third is BIC. The rows represent $T = 64, 96, 128, 160$.

that is that the baseline and the drift were regressed out at each pixel. This is a common practice in fMRI analysis.

Figure 4.14 displays SURE plot for brain slice number 5 along with its components. It selects $r = 9$ PCs, it is interesting that in this case there is an intriguing jump in the interaction term at $r = 31$. The reason is that the interaction term has a dividing term of $l_j - l_i$, and it turns that the smallest difference between the eigenvalues is 0.0208 between eigenvalue 31 and 32 therefore causing the jump. For comparison,



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.8: Simulation: Histogram of number of nPCs, where $r = 5$ and $\lambda_r = 2$. First column is SURE, second is Laplace, and third is BIC. The rows represent $T = 64, 96, 128, 160$.

Figure 4.15 shows the Laplace and the BIC plots. The Laplace criterion picks $r = 11$ components. But the BIC is more complex. We believe the fall-off it at high values of r is due to the RMT scenario and that the appropriate choice of r should be based on the first minimum of the BIC which is $r = 5$. This differs a lot from the choice of SURE and Laplace. The BIC example brings up an important point advocated by [84, 86] that in cases of multiple local minimums of the model selection criterion one should not select the global minimum blindly but look at all them before making

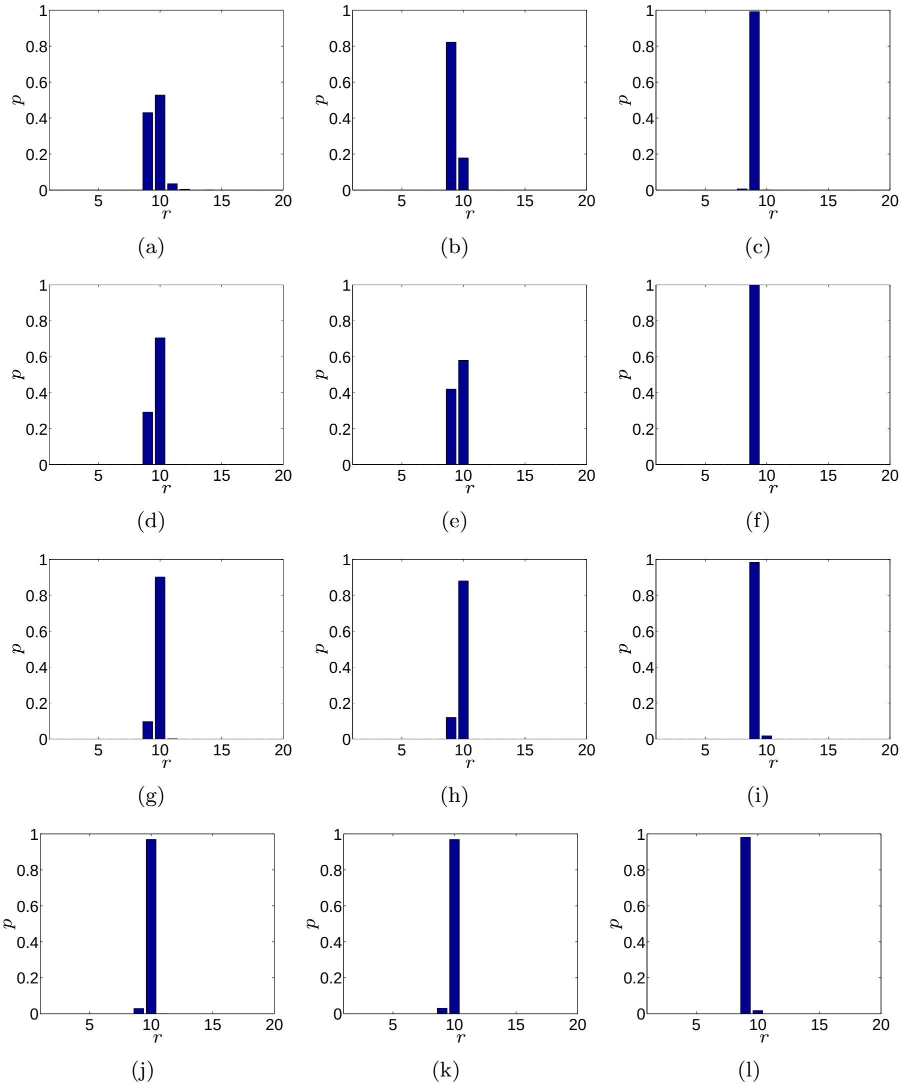
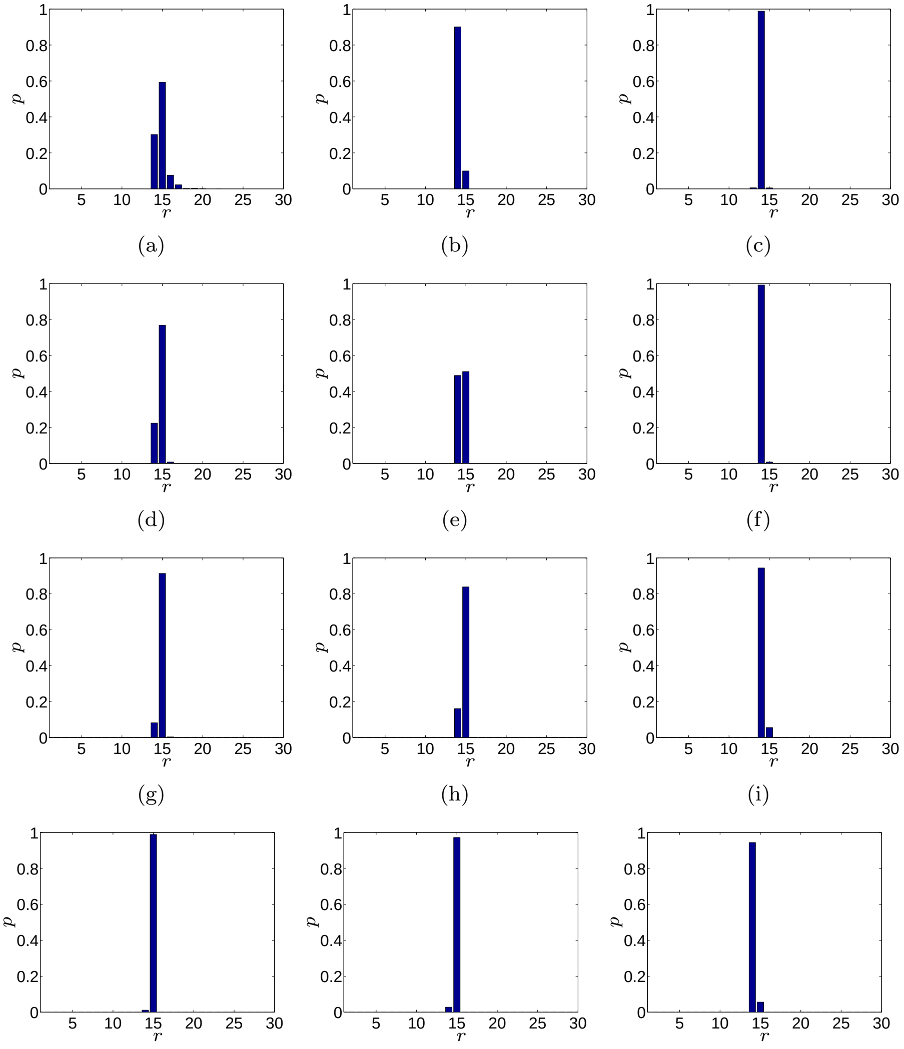


Figure 4.9: Simulation: Histogram of number of nPCs, where $r = 10$ and $\lambda_r = 2$. First column is SURE, second is Laplace, and third is BIC. The rows represent $T = 64, 96, 128, 160$.

final decision.

Table 4.4 shows selection result for all the brain slices. The difference in selection between the Laplace method and SURE is 0-3 components. But the BIC result is very different it generally selects much fewer components than the other methods.

Notice that all of the methods we looked at assume that the observations, in this case brain slices, are independent. This is generally not true for fMRI data. An obvious improvement to SURE would be to take this fact into account.



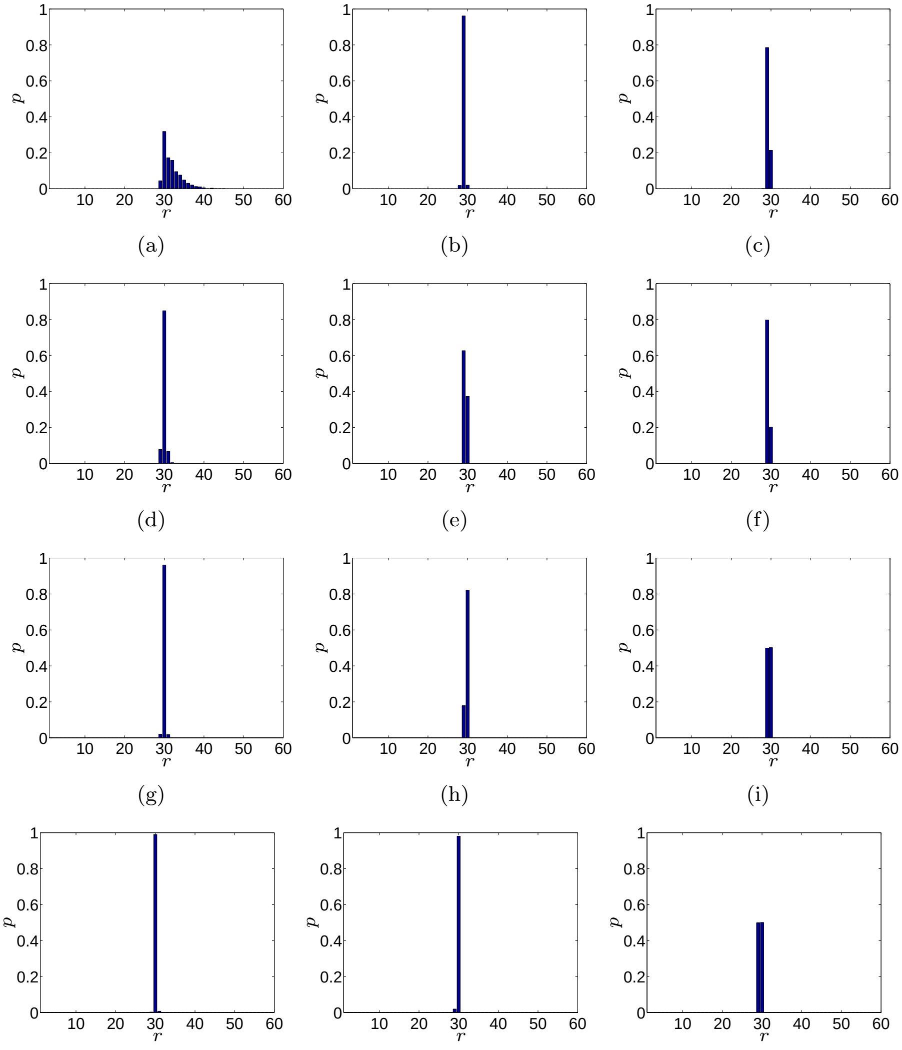
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.10: Simulation: Histogram of number of nPCs, where $r = 15$ and $\lambda_r = 2$. First column is SURE, second is Laplace, and third is BIC. The rows represent $T = 64, 96, 128, 160$.



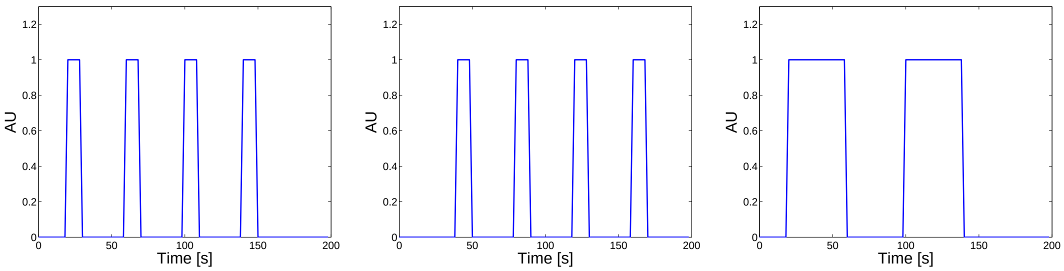
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.11: Simulation: Histogram of number of nPCs, where $r = 30$ and $\lambda_r = 2$. First column is SURE, second is Laplace, and third is BIC. The rows represent $T = 64, 96, 128, 160$.



(a) Annular checkerboard (b) Anti-annular checkerboard (c) Finger-Thumb opposition

Figure 4.12: The AFNI stimulus signals.

Table 4.2: Percentage of correct selection for $\lambda_r = 1.5$. Bold face entries represent the best performing method

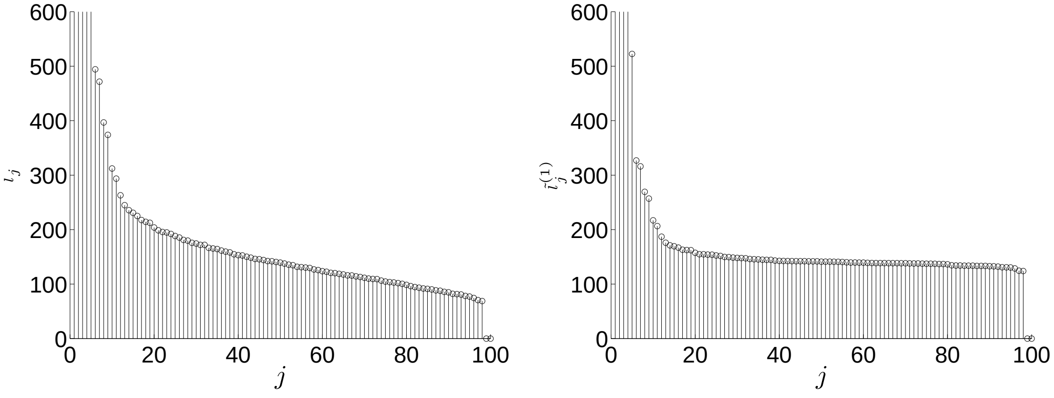
$T = 64$				
r	SURE	Laplace	BIC	RISK
5	**0.169**	0.074	0	0.022
10	**0.279**	0.031	0	0.025
15	**0.373**	0.014	0	0.032
30	**0.205**	0.003	0.043	0.068
$T = 96$	SURE	Laplace	BIC	RISK
5	**0.268**	**0.263**	0	0.187
10	**0.333**	0.198	0	0.219
15	**0.422**	0.142	0	0.277
30	**0.671**	0.100	0.0353	0.442
$T = 128$	SURE	Laplace	BIC	RISK
5	0.521	**0.552**	0	0.504
10	**0.538**	0.469	0	0.583
15	**0.636**	0.451	0	0.643
30	**0.830**	0.423	0.039	0.823
$T = 160$	SURE	Laplace	BIC	RISK
5	0.711	**0.742**	0	0.815
10	**0.749**	0.725	0	0.865
15	**0.802**	0.700	0	0.897
30	**0.923**	0.729	0.062	0.967

Table 4.3: Percentage of correct selection for $\lambda_r = 2$. Bold face entries represent the best performing method.

$T = 64$				
r	SURE	Laplace	BIC	RISK
5	**0.425**	0.285	0	0.226
10	**0.536**	0.175	0	0.257
15	**0.577**	0.092	0.005	0.244
30	**0.242**	0.015	0.117	0.323
$T = 96$	SURE	Laplace	BIC	RISK
5	**0.671**	**0.661**	0	0.726
10	**0.718**	0.571	0	0.735
15	**0.775**	0.498	0.010	0.777
30	**0.825**	0.353	0.185	0.846
$T = 128$	SURE	Laplace	BIC	RISK
5	**0.886**	**0.899**	0	0.957
10	**0.901**	0.883	0.005	0.955
15	**0.930**	0.840	0.022	0.958
30	**0.956**	0.833	0.299	0.991
$T = 160$	SURE	Laplace	BIC	RISK
5	**0.965**	**0.970**	0.009	0.991
10	**0.977**	**0.975**	0.015	0.993
15	**0.981**	0.965	0.062	0.998
30	**0.983**	0.973	0.473	1.000

Table 4.4: Number of nPCs for the AFNI fMRI data set

Slice nr.	M	SURE	Laplace	BIC
1	368	14	14	7
2	622	9	10	6
3	843	9	9	5
4	1024	9	11	5
5	1185	9	11	5
6	1329	9	10	5
7	1462	9	10	5
8	1574	9	10	5
9	1656	10	11	4
10	1719	10	11	4
11	1761	10	11	4
12	1794	13	13	5
13	1809	14	14	5
14	1767	16	16	7
15	1691	18	17	6
16	1542	16	16	6
17	1312	19	19	9
18	1257	24	21	11
19	1122	20	19	11
20	1000	19	19	8
21	798	17	17	10



(a) Scree Plot

(b) Corrected Scree Plot

Figure 4.13: The Scree plot and the corrected Scree plot for the fMRI data.

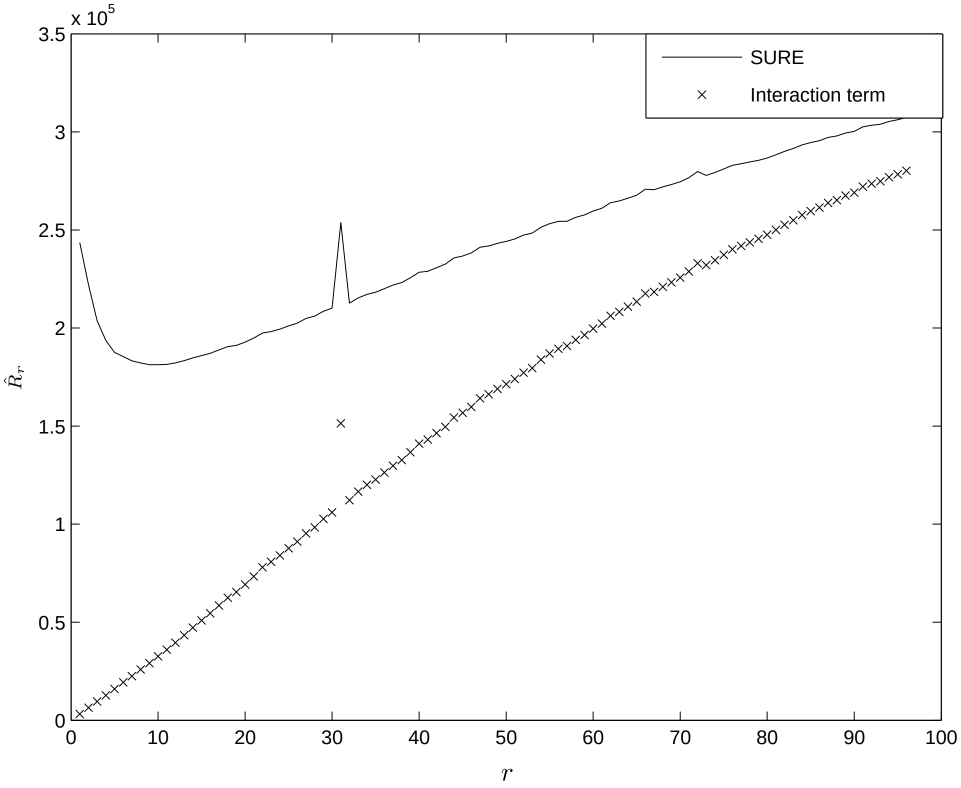
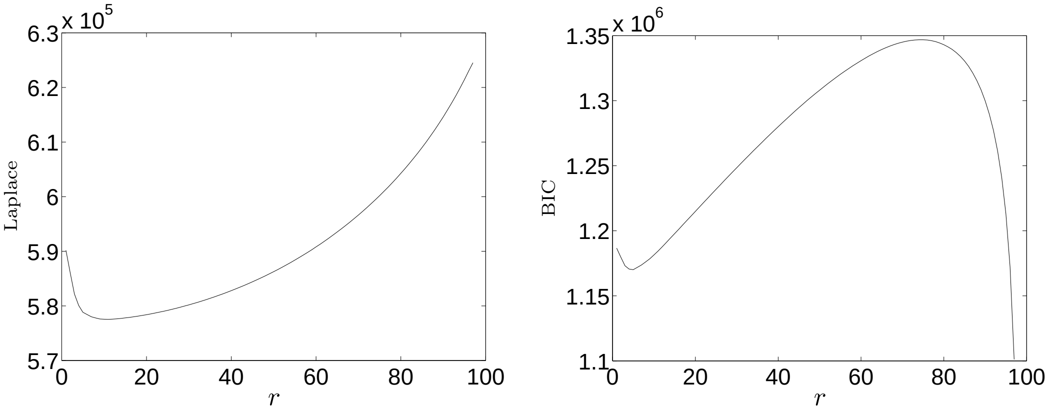


Figure 4.14: The SURE for the fMRI data



(a) Laplace

(b) BIC

Figure 4.15: The BIC and the Laplace method for the fMRI data.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

Traditionally, there are two main paths to analyze fMRI data; firstly univoxel methods, and secondly multivoxel methods. Univoxel methods have more solid statistical grounding than multivoxel methods which are usually exploratory, i.e., not based on a statistical model. In this thesis, we have demonstrated that the model based nPCA is a very useful tool for analysis of fMRI data.

In Chapter 2, we introduced two nPCA based models which extends the nPCA model so it recognizes temporal smoothness, and also demonstrated how to make it spatially local. The BIC criterion was used in a novel way to select three parameter simultaneously, i.e., the number of nPCA, the degree of smoothness and spatial localization. We showed that by using temporally smoothed nPCA to model the noise we are able to handle non-stationary noise. Which is something that univoxel methods are incapable of doing. In addition, we showed that it is possible to construct a likelihood ratio test statistic for the nPCA based models, and very importantly decompose it spatially.

In Chapter 3, we introduced a new model based sparse variable PCA. The main method we presented is based on optimizing a penalized log-likelihood where the penalty is specially designed to allow for sparseness. We developed a practical algorithms based on geodesic descent methods. In addition, we also presented alternative approach based on the EM algorithm. These methods are a valuable addition to a class of sparse PCA methods that have been developed recently in statistics since those methods are not based on a statistical model so model selection and inference is problematic. Moreover, we introduced a new terminology that distinguished between whether the methods zero out just few loadings of a variable or the whole variable. Finally, we exhibited how to select the number of svnPCs and the sparseness tuning parameter by using BIC.

In Chapter 4, we presented a new method to select the number of noisy principal component based on the nonlinear SURE technique with some help from Random matrix theory. We have the scenario where the number of time points and voxels are in the realms of RMT. This scenario causes significant problem for most other selection methods. For practical use, it is necessary to estimate the noise variance and we have developed a reliable estimator based on RMT. In simulations we have shown that BIC fails badly and our new method outperforms the Laplace method, especially in cases where T and M are low.

5.1 Future Work

The thesis provides quite a few interesting future research directions:

- Develop SURE to select the number of nPCA and the sparseness tuning parameter for svnPCA.

- Further development of the spatial decomposition of the Likelihood ratio test in relation to both the temporally smooth and spatially local models in Chapter 2 and in terms of svnPCA. The specification of the null hypothesis and the alternative to get good detection of the activation is far from trivial. Moreover, it would be interesting to analyze the distributional properties of the LRT statistic.

- Development of convergence theory for the EM algorithm that can deal with the l_0 penalty is an interesting topic.

- The simplicity of the l_0 penalty EM algorithm for svnPCA is very appealing. The zeroing out of variables was done by a simple hard thresholding operation. Development of an EM algorithm for the l_2 penalty is an interesting topic.

- In the svnPCA framework we assume that svnPCs $u_t, t = 1, ..., T$ are independent. It is of interest to investigate how to deal with temporal correlation. A good starting point would be the paper by Pham [114] which uses discrete Fourier transform to decorrelate the signals, and [66] that uses autocorrelation models. These papers deal with ICA, but the ideas are applicable for nPCA.

- Independent component analysis and canonical correlation analysis have been successfully applied in fMRI research. Up to this point the use has been exploratory. An interesting research direction would be to develop a statistical framework for those method in a similar manner as was done for PCA in this thesis.

- Although the methods in this thesis were designed with application to fMRI in mind they are more generally applicable. We will seek collaboration with researcher in other fields.

APPENDICES

A Derivation of the nPCA ML Estimates

The log-likelihood is given by

$$l_\theta(y) = -\frac{T}{2}\mathrm{tr}(S_y\Omega^{-1}) - \frac{T}{2}\log|\Omega|.$$

By using the identities

$$\begin{aligned} d\Omega^{-1} &= -\Omega^{-1}d\Omega\Omega^{-1} \\ d\log|\Omega| &= \mathrm{tr}(\Omega^{-1}d\Omega) \end{aligned}$$

we get that the first differential of the log-likelihood is given by

$$dl_\theta = -\frac{T}{2}\mathrm{tr}[(\Omega^{-1}S_y\Omega^{-1} - \Omega^{-1})d\Omega].$$

The differential of Ω w.r.t. G is given by

$$d\Omega = (dG)G^T + G(dG^T).$$

We set the first differential of dl_θ to zero and obtain the Euler equations for stationary points

$$S_y\Omega^{-1}G = G.$$

Now use the matrix inversion lemma

$$\begin{aligned} S_y(I_M - G(G^TG + \sigma^2 I_r)^{-1}G^T)G &= \sigma^2 G \\ S_yG(I_r - (I_r + \sigma^2(G^TG)^{-1})^{-1}) &= \sigma^2 G. \end{aligned}$$

Now write $G = K_rD_r^{1/2}R^T$ in terms of its SVD

$$\begin{aligned} S_yK_rD_r^{1/2}[I_r - (I_r + \sigma^2 D_r^{-1})^{-1}] &= \sigma^2 K_rD_r^{1/2} \\ S_yK_rD_r^{1/2}(D_r + \sigma^2 I_r)^{-1} &= K_rD_r^{1/2} \\ S_yK_r &= K_r(D_r + \sigma^2 I_r). \qquad \text{(A.1)} \end{aligned}$$

From Equation (A.1) we see that K_r contains r eigenvectors of S_y (not necessarily the the largest ones) and $D_r + \sigma^2 I_r$ contains the corresponding r eigenvalues. Therefore we get

$$\hat{G} = K_r(D_r - \hat{\sigma}^2 I_r)^{1/2} R^T. \tag{A.2}$$

Now optimize in terms of σ^2. Computing $\frac{\partial l_\theta}{\partial \sigma^2}$ and solving the Euler equation yields (very similar computation can be found in Appendix F)

$$\hat{\sigma}^2 = \frac{\mathrm{tr}(S_y) - \mathrm{tr}(D_r)}{M - r}. \tag{A.3}$$

Solution (A.3) and (A.2) represent the stationary points of the log-likelihood. To determine whether those point represent maximum, minimum or saddle points on the likelihood surface we have to look at the second differential and use the following fact. If

$$d^2 l_\theta \leq 0$$

for all perturbations at the stationary point, then the point is a local maximum (for a more careful statement of this fact and a proof see [91]). By considering perturbation to a column vector $\hat{g}_i$ of the form p_j, where p_j is a eigenvector of S_y, Tipping [152] demonstrated that all the stationary points represent saddle points except the one where K_r is equal to the r largest eigenvectors of S_y which represents global maximum. Hence, the MLE is given by

$$\begin{aligned}\hat{\Lambda} &= L_r - \hat{\sigma}^2 I_r \\ \hat{G} &= P_r(L_r - \hat{\sigma}^2 I_r)^{1/2} R^T.\end{aligned}$$

B The nPCA Maximum Likelihood

In this section, we obtain an expression for the ML. Application of the matrix inversion lemma easily gives

$$\mathrm{tr}(\hat{\Omega}^{-1}S_y) = M \tag{B.1}$$

and

$$\log|\Omega| = \sum_{j=1}^{r} \log(l_j) + (M-r)\log\hat{\sigma}^2.$$

Using this we obtain an expression for the maximum log-likelihood

$$\begin{aligned} l_{\hat{\theta}}(y) &= -\frac{T}{2}\mathrm{tr}(\hat{\Omega}^{-1}S_y) - \frac{T}{2}\log|\hat{\Omega}| \\ &= -\frac{MT}{2} - \frac{T}{2}\sum_{j=1}^{r}\log(l_j) - \frac{T(M-r)}{2}\log\hat{\sigma}^2. \end{aligned} \tag{B.2}$$

C The SURE Criterion

In this section, we derive the SURE criterion in a more straightforward manner than [142]. Let $\hat{\mu} = \hat{\mu}_r(y)$ be a differentiable function of the data y. We would like to choose r to minimize the risk

$$R_r = E\|\mu - \hat{\mu}\|^2.$$

By adding and subtracting the true signal we can get an expression in terms of the error signal n

$$\begin{aligned} R_r &= E\|y - \hat{\mu} - (y-\mu)\|^2 \\ &= E\|n - \epsilon\|^2 \\ &= E\|n\|^2 - 2E(n^T\epsilon) + E\|\epsilon\|^2 \\ &= E\|n\|^2 - 2E(n^T\epsilon) + M\sigma^2. \end{aligned}$$

To make progress we need an expression for the cross-covariance between the error and the noise:

$$\begin{aligned} E(n^T\epsilon) &= \int n^T \epsilon p(\epsilon) d\epsilon \\ &= \int n^T (y-\mu) p(y-\mu) dy. \end{aligned} \tag{C.1}$$

The data y follows Gaussian distribution

$$p(y-\mu) = \frac{1}{(2\pi\sigma^2)^{M/2}} n^{-\frac{\|y-\mu\|^2}{2\sigma^2}}$$

and this leads to the key observation $\frac{\partial p}{\partial y} = \frac{-1}{\sigma^2} p(y-\mu)$. Which we now use in (C.1)

$$\begin{aligned} E(n^T\epsilon) &= -\sigma^2 \int n^T \frac{\partial p}{\partial y} dy \\ &= -\sigma^2 \int (\sum_{v=1}^{M} n_v \frac{\partial p}{\partial y_v}) dy. \end{aligned}$$

Integrating by parts gives

$$\begin{aligned} E(n^T\epsilon) &= \sigma^2 \int (\sum_{v=1}^{M} \frac{\partial n_v}{\partial y_v}) p dy \\ &= \sigma^2 E \mathrm{tr}(\frac{\partial n}{\partial y^T}) \\ &= M\sigma^2 - \sigma^2 E \mathrm{tr} \frac{\partial \hat{\mu}}{\partial y^T}. \end{aligned}$$

We finally get

$$R_r = E\|n\|^2 + \sigma^2 E \mathrm{tr} \frac{\partial \hat{\mu}}{\partial y^T} - M\sigma^2.$$

We exchange the expectation operator E for the sample average and get the SURE criterion which is thus unbiased.

D The Interaction Term in the SURE Formula

The interaction term is given by

$$C = \frac{2\sigma^2}{T} \sum_{j=1}^{r} (1 - \frac{\hat{\sigma}_r^2}{l_j}) \sum_{i \neq j} \frac{l_j + l_i}{l_j - l_i}. \tag{D.1}$$

We can write

$$\begin{aligned}\sum_{i\neq j}\frac{l_j+l_i}{l_j-l_i} &= \sum_{i\neq j}\frac{2l_j+l_i-l_j}{l_j-l_i}\\ &= 2l_j\sum_{i\neq j}\frac{1}{l_j-l_i}-(M-1).\end{aligned}\tag{D.2}$$

Equations (D.1) and (D.2) together give

$$C=\frac{4\sigma^2}{T}\sum_{i=1}^{r}\sum_{i\neq j}\frac{l_j-\hat{\sigma}_r^2}{l_j-l_i}-\frac{\sigma^2}{T}\sum_{j=1}^{r}(1-\frac{\hat{\sigma}_r^2}{l_j})(M-1)\tag{D.3}$$

but

$$\begin{aligned}\sum_{j=1}^{r}\sum_{i\neq j}\frac{l_j-\hat{\sigma}_r^2}{l_j-l_i} &= \sum_{j=1}^{r}(\sum_{\substack{j=1\\ i\neq j}}^{r}\frac{l_j-\hat{\sigma}_r^2}{l_j-l_i}+\sum_{i=r+1}^{M}\frac{l_j-\hat{\sigma}_r^2}{l_j-l_i})\\ &= \frac{r(r-1)}{2}+\sum_{j=1}^{r}\sum_{i=r+1}^{M}\frac{l_j-\hat{\sigma}_r^2}{l_j-l_i})\end{aligned}$$

where we used that

$$\begin{aligned}\sum_{\substack{j=1\\ i\neq j}}^{r}\frac{l_j-\hat{\sigma}_r^2}{l_j-l_i} &= \frac{1}{2}\sum_{\substack{j=1\\ i\neq j}}^{r}\left(\frac{l_j-\hat{\sigma}_r^2}{l_j-l_i}+\frac{l_j-\hat{\sigma}_r^2}{l_j-l_i}\right)\\ &= \frac{1}{2}\sum_{\substack{j=1\\ i\neq j}}^{r}\frac{l_j-l_i}{l_j-l_i}\\ &= \frac{r(r-1)}{2}.\end{aligned}$$

Using this in Equation (D.3) gives the final expression for the interaction term

$$\begin{aligned}C &= \frac{4\sigma^2}{T}\sum_{j=1}^{r}\sum_{i=r+1}^{M}\frac{l_j-\hat{\sigma}_r^2}{l_j-l_i}+\frac{2\sigma^2}{T}r(r-1)\\ &- \frac{2\sigma^2}{T}(M-1)\sum_{j=1}^{r}(1-\frac{\hat{\sigma}_r^2}{l_j}).\end{aligned}$$

E Conditional Expectation for the M-step of the EM Algorithm in Section 3.10.2

In this section, we compute the conditional expectation needed to do the M-step of the EM algorithm in Chapter III. According to Model (1.9) we have

$$\begin{aligned}\begin{pmatrix} y \\ u \end{pmatrix} &\sim \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{pmatrix}\right] \\ &= N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} GG^T + \sigma^2 I_M & G \\ G^T & I_r \end{pmatrix}\right].\end{aligned}$$

From normal distribution theory [94] [p.63,Thm. 3.2.4] we have

$$u|y \sim N(\Omega_{21}\Omega_{11}^{-1}y, \Omega_{22} - \Omega_{21}\Omega_{11}^{-1}\Omega_{12}).$$

We use this expression and compute

$$\begin{aligned}\mathrm{E}(u|y) &= G^T(G\Lambda G^T + \sigma^2 I_M)^{-1}y \\ &= \frac{1}{\sigma^2}(I_r - (I_r + \sigma^2(G^TG)^{-1})^{-1})G^Ty \\ &= (G^TG + \sigma^2 I_r)^{-1}G^Ty \qquad \text{(E.1)} \\ &= W^{-1}G^Ty.\end{aligned}$$

To get to Equation (E.1) we used the matrix inversion lemma backwards. Now we compute B_0

$$\begin{aligned}B_0 &= \frac{1}{T}\sum_{t=1}^{T} y_t E_{\theta_0}^T[u_t|y_t] \\ &= \frac{1}{T}\sum_{t=1}^{T} y_t y_t^T G_0 W_0^{-1} \\ &= S_y G_0 W_0^{-1}.\end{aligned}$$

Now compute the variance of u_t given y_t

$$\begin{aligned}
\text{var}(u|y) &= I_r - G^T(GG^T + \sigma^2 I_M)^{-1}G \\
&= I_r - (G^TG + \sigma^2 I_r)^{-1}G^TG \\
&= I_r - (I_r + \sigma^2(G^TG)^{-1})^{-1} \\
&= \sigma^2 W^{-1}.
\end{aligned}$$

Finally we compute A_0

$$\begin{aligned}
A_0 &= \frac{1}{T}\sum_{t=1}^{T} E_{\theta_0}[u_t u_t^T|y_t] \\
&= \text{var}(u_t|y_t) + \frac{1}{T}\sum_{t=1}^{T} E_{\theta_0}[u_t|y_t]E_{\theta_0}[u_t|y_t]^T \\
&= \sigma_0^2 W_0^{-1} + W_0^{-1}G_0^T S_y G_0 W_0^{-1}.
\end{aligned}$$

F svnPCA Λ-step: Estimation of Λ and σ^2.

Given F_1 and the fact that $F_1^T F_1 = I_r$ we find that $J_\theta(y)$ diagonalizes to become (with $\text{tr}(S_y) = \tau_y$; $V_i = (F_1^T S_y F_1)_{ii}$)

$$\begin{aligned}
J_\theta(y) &= -\frac{\tau_y}{2\sigma^2} + \frac{1}{2\sigma^2}\sum_{i=1}^{r}\frac{V_i}{1+\sigma^2/\lambda_i} \\
&- \frac{M-r}{2}\log(\sigma^2) - \frac{1}{2}\sum_{i=1}^{r}\log(\lambda_i + \sigma^2) \\
&= J_a(\sigma^2) + J_b(d) \qquad \text{(F.2)}
\end{aligned}$$

where

$$\begin{aligned}
d_i &= \lambda_i + \sigma^2 \\
J_a(\sigma^2) &= -\frac{1}{2\sigma^2}(\tau_y - \sum_{i=1}^{r} V_i) - \frac{M-r}{2}\log(\sigma^2) \\
J_b(d) &= -\frac{1}{2}\sum_{i=1}^{r}\frac{V_i}{d_i} - \frac{1}{2}\sum_{i=1}^{r}\log(d_i).
\end{aligned}$$

Thus $J_\theta(y)$ is convex in σ^2, d and so has a unique minimum which by elementary calculus is

$$\begin{aligned}\hat{\sigma}^2 &= \frac{1}{M-r}(\tau_y - \sum_{i=1}^{r} V_i)\\ \lambda_i &= V_i - \hat{\sigma}^2.\end{aligned}$$

G A Derivation of the Stiefel Gradient

The Stiefel gradient is a tangent vector at F that satisfies

$$\mathrm{tr}(\frac{\partial J}{\partial F^T}\Delta) = \langle \tilde{\nabla}_F J_\theta(y), \Delta \rangle$$

for all tangent vectors Δ. This is equivalent to

$$\mathrm{tr}([\frac{\partial J}{\partial F^T} - \tilde{\nabla}_F J_\theta(y)^T + \frac{1}{2}\tilde{\nabla}_F J_\theta(y)^T F F^T]\Delta) = 0.$$

Since Δ is an arbitrary tangent vector, the quantity inside the bracket needs to be a normal vector, which can be represented as

$$\frac{\partial J}{\partial F} - \tilde{\nabla}_F J_\theta(y) + \frac{1}{2} F F^T \tilde{\nabla}_F J_\theta(y) = FS \tag{G.3}$$

where S is a $r \times r$ symmetric matrix. Transposing (G.3) and multiplying by F from the left yields

$$S = \frac{\partial J}{\partial F^T} F - \frac{1}{2}\tilde{\nabla}_F J_\theta(y)^T F. \tag{G.4}$$

Now Equations (G.3), (G.4) and $\tilde{\nabla}_F J_\theta(y)^T F = -F^T \tilde{\nabla}_F J_\theta(y)$ yield

$$\tilde{\nabla}_F J_\theta(y) = \frac{\partial J_\theta(y)}{\partial F} - F\frac{\partial J_\theta(y)}{\partial F^T} F. \tag{G.5}$$

H The Proof of the Cyclic Descent Convergence Theorem

In this section, we present the proof of the cyclic descent convergence theorem [1]. Below we use (A4) to show that $\|\theta_{m+1} - \theta_m\| \to 0$. Since $\{\theta_m\}$ is bounded we can

[1] The proof has been slightly modified from [141] to handle the constraints.

find a subsequence $\{\theta_{m_k}\}$ converging to a limit point θ_*. From the cyclic descent algorithm we have

$$\nabla_\beta J_{(\alpha_{m-1},\beta_m)}(y) = 0, \quad \tilde{\nabla}_\alpha J_{(\alpha_m,\beta_m)}(y) = 0 \tag{H.6}$$

and from continuity we find $\tilde{\nabla}_\alpha J_{\theta_*}(y) = 0$. But also then

$$(\alpha_{m_k}, \beta_{m_k+1}) = (\alpha_{m_k}, \beta_{m_k}) + (0, \beta_{m_k+1} - \beta_{m_k}) \to \theta_*.$$

So from (H.6) via continuity $\nabla_\beta J_{\theta_*}(y) = 0$. So θ_* is a stationary point.

Next we appeal to Ostrowski's theorem [111], i.e., $\|\theta_m\|$ uniformly bounded, $\|\theta_{m+1} - \theta_m\| \to 0$ implies that the set of limit points is a compact connected set. And the results are established.

Suppose now $\|\theta_{m+1} - \theta_m\| \nrightarrow 0$. Then by boundedness there is a subsequence $\{\theta_{m_k}\}$ with $\{\theta_{m_k}\} \to \theta'$, $\{\theta_{m_k+1}\} \to \theta''$ and $\|\theta' - \theta''\| > 0$. If θ_{m_k} is a stationary point there is nothing to prove. On the other hand by (A4), $J_{\theta_{m_k+1}}(y) > J_{\theta_{m_k}}(y)$ and continuity (A2) gives $J_{\theta'}(y) > J_{\theta''}(y)$. Continuing, on the other hand $J_m = J_{\theta_m}(y)$ is a bounded nondecreasing sequence and so has a limit J_* and so $J_m - J_{m-1} \to 0 \Rightarrow J_{\theta_{m_k+1(y)}} - J_{\theta_{m_k}}(y) \to 0$, which, via continuity, gives $J_{\theta'}(y) - J_{\theta''}(y) = 0$, which is a contradiction. So, indeed $\|\theta_{m+1} - \theta_m\| \to 0$.

BIBLIOGRAPHY

[1] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Automat. Contr.*, 19(6), 1974.

[2] S. Alliney and S.A. Ruzinsky. An algorithm for the minimization of mixed l_1 and l_2 norms with application to Bayesian estimation. *IEEE Trans. Signal Proc.*, 42(3):618–627, 1994.

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

[4] T.W. Anderson. Asymptotic theory for principal component analysis. *Ann. J. Math. Stat.*, 34:122–148, 1963.

[5] T.W. Anderson. Estimating linear statistical relationships. *Ann. Statist.*, 12(1):1–45, 1984.

[6] T.W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, New York, NY, third edition, 2003.

[7] W. Backfrieder, R. Baumgartner, M. Samal, E. Mosert, and H. Bergman. Quantification of intensity variation in functional MR images using rotated principal components. *Phys. Med. Biol.*, 41:1425–1438, 1996.

[8] Z.D. Bai and J.W. Silverstein. CLT for linear spectral statistics of large-dimensional sample covariance matrices. *Ann Prob*, 32(1A):553–605, 2004.

[9] J. Baik and J.W. Silverstein. Eigenvalues of large sample covariance matrices of spiked population models. *J. Multivariate Anal.*, 97:1382–1408, 2006.

[10] M.S. Bartlett. A note on the multiplying factors for various χ^2 approximations. *J. Roy Stat. Soc., Series B.*, 16:296–298, 1954.

[11] N. Bazargani, A. Nosratina, K. Gopinath, and R. Briggs. fMRI baseline drift estimation method by MDL principle. In *Proc. IEEE International Symposium on Biomedical Imaging (ISBI'07)*, Washington D.C., 2007.

[12] C.F. Beckmann and S.M. Smith. Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE Trans. Med. Imag*, 23(2), 2004.

[13] Y. Behzadi, K. Restom, J. Liau, and T. Liu. A component based noise correction method (CompCor) for BOLD and perfusion based fMRI. *Neuromimage*, 2007.

[14] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, second edition, 1999.

[15] P.J. Bickel and K.A. Doksum. *Mathematical Statistics*, volume 1. Prentice Hall, Upper Saddle River, NJ, second edition, 2007. Updated Printing.

[16] C. Bishop. Bayesian PCA. In *Advances in Neural Information Processing Systems (NIPS'98)*, volume 11, pages 382–388, 1998.

[17] B. Biswal, F. Yetkin, V. Haughton, and J. Hyde. Functional connectivity in the motor cortex of resting state human brain using echo-planar MRI. *Magn. Reson. Med.*, 34:537–541, 1995.

[18] G.M. Boynton, S.A. Engel, G.G. Glover, and D.J. Heeger. Linear Systems Analysis of Functional Magnetic Resonance Imaging in Human V1. *J. Neurosci.*, 16(13):4207–4221, 1996.

[19] R.L. Buckner. The hemodynamic inverse problem: Making inferences about neural activity from measured MRI signals. *Proc. Nat. Ac. Sci.*, 100(5):2177–2179, 2003.

[20] E. Bullmore, S.C. Brammer, S. Rabe-Hasketh, N. Janot, D.A. Mellers, R.J. Howard, and P. Sham. Statistical methods of estimation and inference for functional MR image analysis. *Magn. Reson. Med.*, 35(2):261–277, 1996.

[21] E.T. Bullmore, S. Rabe-Hesketh, R.G. Morris, S.C. Williams, L. Gregory, J.A. Gray, and M.J. Brammer. Functional magnetic resonance image analysis of a large-scale neurocognitive network. *Neuroimage*, 4(1):16–33, Aug 1996.

[22] M.A. Burock and A.M Dale. Estimation and detection of event-related fMRI signals with temporally correlated noise: A statistically efficient and unbiased approach. *Human Brain Mapping*, 11:249–260, 2000.

[23] R.B. Buxton. *Introduction to Functional Magnetic Resonance Imaging: Principles and Techniques*. Cambridge University Press, Cambridge, England, 2002.

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

[25] V.D. Calhoun, T. Adali, J. Hansen, J. Larsen, and J.J. Pekar. ICA of functional MRI data: an overview. *4th International Symposium on Independent Component Analysis and Blind Signal Separation*, pages 281–288, April 2003.

[26] A. Cichocki and P. Georgiev. Blind source separation algorithms with matrix constraints. *IEICE Trans. Fundamentals*, E86-A(1):1–9, 2003.

[27] M.S. Cohen. Parametric analysis of fMRI data using linear system methods. *NeuroImage*, 6:93–103, 1997.

[28] T.M. Cover and J.A. Thomas. *The Elements of Information Theory.* Prentice Hall Inc., Englewood Cliffs N.J., 1991.

[29] Robert Cox. AFNI.

[30] P. Craven and G. Wahba. Smoothing noisy data with spline functions. *Numer. Math.*, 31:377–402, 1979.

[31] A. D'Aspremont, L. E. Ghaoui, M.I. Jordan, and G.R. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. To be published in SIAM review.

[32] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc., Series B.*, 39(1):1–38, 1977.

[33] J.A. Detre and J. Wang. Technical aspects and utility of fMRI using BOLD and ASL. *Clinical Neurophysiology*, 113:621–631, 2002.

[34] D.L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inform. Theory*, 41(3):613–627, 1995.

[35] D.L. Donoho and I. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[36] S.C. Douglas, S. Amari, and S.Y. Kung. On gradient adaptation with unit norm constraints. *IEEE Trans. Signal Proc.*, 48(6):1843–1847, 2000.

[37] H.T. Eastment and W.J. Krzanowski. Cross-validatory choice of the number of components from a principal component analysis. *Techometrics*, 24(1), 1982.

[38] A. Edelman, T.A. Aries, and S.T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, 1998.

[39] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Stat.*, 32(2):407–499, 2004.

[40] L. Elden and H. Park. A Procrustes problem on the Stiefel manifold. *Numerische Mathematik*, 82(4):599–619, 1999.

[41] A.C. Evans, S. Marret, P. Neelin, L. Collins, K. Worsley, W. Dai, S. Milot, E. Meyer, and D. Bub. Anatomical mapping of functional activation in stereotactic coordinate space. *Neuroimage*, 1(1):43–53, 1992.

[42] R. Everson and S. Roberts. Inferring the eigenvalues of covariance matrices from limited, noisy data. *IEEE Trans. Signal Proc.*, 48(7):2083–2091, 2000.

[43] J. Fadili and E. Bullmore. Penalized partially linear models using sparse representation with an application to fMRI time series. *IEEE Trans. Signal Proc.*, 53(9), 2005.

[44] L. Ferre. Improvement of some multidimensional estimates by reduction of dimensionality. *J. Multivariate Anal.*, 54:147–162, 1995.

[45] L. Ferre. Selection of components in principal component analysis: a comparison of methods. *Comput. Stat. and Data Anal.*, 19:669–682, 1995.

[46] S. Fiori. A theory for learning by weight flow on stiefel-grassman manifold. *Neural Comput.*, 13:1625–1637, 2001.

[47] R.S.J. Frackowiak, J.T. Ashburner, W.D. Penny, and S.Zeki. *Human Brain Function.* Academic Press, San Diego, CA, USA, 2004.

[48] J. Friedman and J. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Computers*, 23:881–889, 1974.

[49] O. Friman, M. Borga, P. Lundberg, and Knutsson. Exploratory fMRI analysis by autocorrelation maximization. *NeuroImage*, 16:454–464, 2002.

[50] O. Friman, J. Cedefamn, P. Lundberg, Magnus Borga, and Hans Knutsson. Detection of neural activity in functional MRI using canonical correlation analysis. *Magn. Res. Med.*, 45:323–330, 2001.

[51] K. Friston, O. Joseph, E. Zarahn, A. Holmes, S. Roquette, and J. Poline. To smooth or not to smooth? bias and efficiency in fMRI time-series analysis. *Neuroimage*, 12:196–208, 2000.

[52] K. Friston, J. Phillips, D. Chawla, and C. Buchel. Nonlinear PCA: characterizing interaction between modes of brain activity. *Phil Trans R Soc Lond B*, 355:135–146, 2000.

[53] K. J. Friston, C. D. Frith, P. F. Liddle, and R. S. Frackowiak. Functional Connectivity: The principal component analysis of large data sets. *J. Cereb. Blood Flow Metab.*, 13:5–14, 1993.

[54] K.J Friston. Eigenimages and multivariate analyses. *Human Brain Mapping*, 1994.

[55] K.J. Friston, C.D. Frith, S.J. Frackowiak, and R. Turner. Characterizing dynamic brain responses with fMRI: A multivariate approach. *Neuroimage*, 2:166–172, 1995.

[56] K.J. Friston, P. Jezzard, and R. Turner. Analysis of functional MRI series. *Human Brain Mapping*, pages 153–171, 1994.

[57] J. Gallier and D. Xu. Computing exponentials of skew symmetric matrices and logarithms of orthogonal matrices. *Int. J. Robotic and Automation*, 18(1), 2003.

[58] G. Glover, T. Li, and D. Ress. Image-based method for retrospective correction of physiological effect in fMRI: RETROICOR. *Magn. Res. Med.*, 44(1):162–167, 2000.

[59] G.H. Golub and C.F. Van Loan. *Matrix Computation*. Johns Hopkins University Press, third edition, 1996.

[60] M. Greicius, B. Krasnow, A. Reiss, and V. Menon. Functional connectivity in the resting brain: A network analysis of the default mode hypothesis. *Proc. Natl. Acad. Sci.*, 100(1), 2003.

[61] H. Gudbjartsson and S. Patz. The Rician distribution of noisy MRI data. *Magn. Reson. Imag.*, 41(2), 1995.

[62] L.K. Hansen and J. Larsen. Unsupervised learning and generalization. In *Proc. IEEE International Conference on Neural Networks (ICNN'96)*, volume 1, pages 25–30, Washington DC, USA, 1996.

[63] L.K. Hansen, J. Larsen, F.A. Nielsen, S.C. Strother, E. Rostrup, R. Savoy, N. Lange, J. Sidtis, C. Svarer, and O.B. Paulson. Generalizable patterns in neuroimaging: How many principal components? *Neuroimage*, 9:534–544, 1999.

[64] A.E. Hoerl and R. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[65] P.D. Hoff. Model averaging and dimension selection for the singular value decomposition. *J. Amer. Stat. Assoc.*, 102(478):674–685, 2007.

[66] S. Hosseini, C. Jutten, and D.T. Pham. Markovian source separation. *IEEE Trans. Signal Proc.*, 51(12), 2003.

[67] M. Hudson. Maximum likelihood restoration and choice of smoothing parameter in deconvolution of image data subject to Poisson noise. *Comput Stat Data Anal*, 26(4):393–410, 1998.

[68] A. Hyvarien, J. Karhunen, and E Oja. *Independent Component Analysis*. John Wiley and Sons, New York, 2001.

[69] A Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. on Neural Networks*, 10(3), 1999.

[70] J.E. Jackson. *A Users Guide to Principal Components*. Wiley, New York, NY, 1991.

[71] Cardoso J.F. Blind signal separation: Statistical principles. *Proceedings of the IEEE*, 9(10):2009–2025, Oct 1998.

[72] I Johnstone and A.Y. Lu. Sparse principal component analysis. Technical report, Statistics department, Stanford University, 2004.

[73] I.M. Johnstone. On the distribution of the largest eigenvalue in principal component analysis. *Annals Stat.*, 29(2):295–327, 2001.

[74] I.T. Jolliffe. *Principal Component Analysis*. Springer, New York, NY, second edition, 2002.

[75] I.T. Jolliffe and M. Uddin. A modified principal component technique based on the lasso. *J. Comput. Graphical Stat.*, 12(3):531–547, 2003.

[76] H.F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23:187–200, 1958.

[77] R.L. Kashyap. Inconsistency of the AIC rule for estimating the order of AR models. *IEEE Trans. Automat. Contr.*, 25(5):996–998, 1980.

[78] H. Krim and M. Viberg. Two decades of array signal processing research: The parametric approach. *IEEE Signal Proc. Mag.*, 13(4):67–94, 1996.

[79] A.M. Kshirsagar. *Multivariate Analysis.* Marcel Dekker Inc, New York, NY, 1972.

[80] K. Lange. A gradient algorithm locally equivalent to the EM algorithm. *J. Royal Stat. Soc. Series B*, 57(2):425–437, 1995.

[81] D.N. Lawley. A modified method of estimation in factor analysis and some large sample results. In *Uppsala symposium on psychological factor analysis*, pages 35–42, Uppsala, Sweden, 1953.

[82] D.N. Lawley. Tests of significance of the latent roots of the covariance and correlation matrices. *Biometrica*, 43:128–136, 1956.

[83] D.N. Lawley and A.E. Maxwell. *Factor Analysis as a Statistical Method.* Butterworth and Co., London, UK, 1971.

[84] H. Linhart and W. Zucchini. *Model Selection.* Wiley, New York, NY, 1986.

[85] J Liu and Moulin P. Complexity-regularized image denoising. *IEEE Tran. Image Proc.*, 10(6):841–851, 2001.

[86] C. Loader. *Local Regression and Likelihood.* Springer, New York, NY, 1999.

[87] C.J. Long, E.N. Brown, C. Triantafyllou, I. Aharon, L.L. Wald, and V. Solo. Nonstationary noise estimation in functional MRI. *Neuroimage*, 28:890–903, 2005.

[88] D. G. Luenberger. *Introduction to Linear and Nonlinear Programming.* Addison-Wesley, New York, NY, 1973.

[89] D.K. Luenberger. The gradient projection method along geodesic. *Management Sci., Theory Series*, 18(11):620–631, 1972.

[90] T.E. Lund, K. Madsen, K. Sidaros, W. Luo, and T. Nichols. Non-white noise in fMRI: Does modelling have an impact? *Neuroimage*, 29:54–66, 2006.

[91] J.R. Magnus and H. Neudecker. *Matrix Differential Calculus.* Wiley, New York, NY, 1999.

[92] J.H. Manton. Optimization algorithms exploiting unitary constraints. *IEEE Trans. Signal Proc.*, 50(3):635–650, 2002.

[93] V.A. Marcenko and L.A. Pastur. Distribution of eigenvalues of some sets of random matrices. *Math. USSR-Sb.*, 1:507–536, 1967.

[94] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis.* Academic Press, London, UK, 1979.

[95] A.R. McIntosh, F.L. Bookstein, J.V. Haxby, and C.L. Grady. Spatial pattern analysis of functional brain images using partial least squares. *Neuroimage*, 3:143–157, 1996.

[96] M.J. McKeown, S. Makeig, Brown G.G., T.P. Jung, Bell A.J. Kinderman, S.S., and T.J. Sejnowski. Analysis of fMRI data by blind separation into independent spatial component. *Human Brain Mapping*, 6:160–188, 1998.

[97] G.J. McLachlan and T. Krishnan. *The EM algorithm and extensions.* Wiley, New York, NY, 1997.

[98] X. Meng and D. Dyk. The EM algorithm - an old folk song sung to a fast new tune. *J.R. Statist. Soc. B.*, 59(3):511–567, 1997.

[99] F. Meyer. Wavelet-based estimation of a semiparametric generalized linear model of fMRI time-series. *IEEE Trans. Med. Imag*, 22(3), 2003.

[100] R. Meyer. Sufficient conditions for the convergence of monotonic mathematical programming algorithms. *J. Comput. System Sci.*, 12:108–121, 1976.

[101] T.P. Minka. Automatic choice of dimensionality for PCA. In *Advances in Neural Information Processing Systems (NIPS'00)*, pages 598–604, 2000.

[102] P.P. Mitra and B Pesaran. Analysis of dynamic brain imaging data. *Biophysical Journal*, 76:691–708, 1999.

[103] C. Moler and C.V. Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 20(4):801–831, 1978.

[104] T.K. Moon. The Expectation-Maximization Algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996.

[105] R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. Jordan, editor, *Learning in Graphical Models*, pages 355–368, Boston, MA, 1998. Kluwer Academic Publishers.

[106] L. Ng and V. Solo. A data-driven method for choosing smoothing parameters in optical flow problems. In *Proc. IEEE International Conference on Image Processing (ICIP'97)*, volume 3, pages 360–363, Washington DC, USA, October 1997.

[107] Y. Nishimori. Learning algorithm for ICA by geodesic flows and orthogonal groups. In *Proc. International joint conference on Neural Networks (IJCNN'99)*, volume 2, pages 933–938, Washington D.C., 1999.

[108] D. Noll. A primer on MRI and functional MRI. 2001.

[109] H. Obrig, M. Neufang, R. Wenzel, M. Kohl, J. Steinbrink, K. Einhaupl, and A. Villringer. Spontaneous low frequency oscillations of cerebral hemodynamic and metabolism in human adults. *Neuroimage*, 12:623–639, 2000.

[110] S. Ogawa, D.W. Tank, R. Menon, J.M. Ellerman, S.G. Kim, and H. Merkle. Intrinsic signal changes accompanying sensory stimulation: Functional brain mapping with magnetic resonance imaging. *Proc. Natl. Acad. Sci. USA*, 89:5951–5955, 1992.

[111] A.M. Ostrowski. *Solutions of Equations in Euclidian and Banach Spaces.* Academic Press, New York, NY, 1973.

[112] D. Paul. Asymptotics of the leading sample eigenvalues for a spiked covariance model. Technical report, Department of Statistic, Stanford University, 2004.

[113] S.J. Peltier, T.A. Polk, and D.C. Noll. Detecting low-frequency functional connectivity in fMRI using self-organizing map (SOM) algorithm. *Human Brain Mapping*, 20(4):220–226, 2003.

[114] D.T. Pham and P. Garat. Blind separation of mixture of independent sources through a quasi-maximum likelihood approach. *IEEE Trans. Signal Proc.*, 45(7), 1997.

[115] R.W. Preisendorfer. *Principal Component Analysis in Meteorlogy and Oceangraphy.* Elsevier, Amsterdam, Holland, 1988.

[116] P. Purdon, V. Solo, R. Weisskopf, and E. Brown. Locally regularized spatiotemporal modelling and model comparison for functional MRI. *Neuroimage*, 14:912–923, 2000.

[117] J.J. Rajan and P.J.W Rayner. Model order selection for the singular value decomposition and the discrete Karhunen-Loeve transform using Bayesian approach. *IEE Vision, Image, and Signal Proc.*, 144:116–123, 1997.

[118] J.O. Ramsay and B.W. Silverman. *Functional Data Analysis*. Springer, New York, NY, first edition, 1997.

[119] J.O. Ramsay and B.W. Silverman. *Functional Data Analysis*. Springer, New York, NY, second edition, 2005.

[120] C.R. Rao and S.K Mitra. *Generalized Inverse of Matrices and its Application*. John Wiley and Sons, New York, 1971.

[121] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[122] G.K. Robinson. That BLUP is a good thing: The estimation of random effects. *Statistical Science*, 6(1):15–32, 1991.

[123] B.R. Rosen, R.L. Buckner, and A.M. Dale. Event-related functional MRI: Past, present and future. *Proceedings of the National Academy of Sciences*, 95(31):773–780, Feb 1998.

[124] T. Ross, P. Myllymaki, and J. Rissanen. MDL denoising principle revisited. Submitted to IEEE Trans. Information Theory.

[125] D.B. Rowe. Modeling both the magnitude and phase of complex-valued fMRI data. *Neuroimage*, 25:1310–1324, 2005.

[126] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[127] B. Scholkopf, A.J. Smola, and K.R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural comput.*, 10:1299–1319, 1998.

[128] G. Schwartz. Estimating the dimension of a model. *Ann. Stat.*, 6(2):461–464, 1978.

[129] G.A.F. Seber. *Multivariate Observations*. Wiley, New York, NY, 1984.

[130] A. Seghouane and A. Cichocki. Bayesian estimation of the number of principal components. *Signal Processing*, 87:562–568, 2007.

[131] M. Shi and V. Solo. Empirical choice of smoothing parameters in robust optical flow estimation. In *Proc. IEEE International Conference on Signal Processing (ICASSP'04)*, volume 3, pages 349–352, Montreal, Canada, 2004.

[132] J.W. Silverstein. Eigenvalues and eigenvectors of large dimensional sample covariance matrices. *Contemp. Math.*, 50:153–159, 1986.

[133] J.W. Silverstein and P.L. Combettes. Signal detection via spectral theory of large dimensional random matrices. *IEEE Trans. Signal Proc.*, 40(8), 1992.

[134] V. Solo. A sure-fired way to choose smoothing parameters in ill-conditioned inverse problems. In *Proc. IEEE International Conference on Image Processing (ICIP'96)*, volume 3, pages 89–92, Lausanne, Switzerland, 1996.

[135] V. Solo. Selection of regularization parameters for total variation denoising. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'99)*, volume 3, pages 1653–1655, Phoenix, Arizona, USA, 1999.

[136] V. Solo. Automatic stopping criterion for anisotropic diffusion. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'01)*, volume 6, pages 3441–3444, Salt Lake City, Utah, USA, May 2001.

[137] V. Solo. Selection of tuning parameter for support vector machines. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, volume 5, pages 235–240, Philadelphia, PA, 2005.

[138] V. Solo, C. Long, E.N. Brown, E. Aminoff, M. Bar, and S. Saha. FMRI signal modeling using Laguerre polynomials. In *Proc. ICIP*, pages 2431–2434, 2004.

[139] V. Solo and J. Noh. An EM algorithm for Rician fMRI activation detection. In *Proc. IEEE International Symposium on Biomedical Imaging (ISBI'07)*, Washington D.C., 2007.

[140] V. Solo, P. Purdon, and E. Brown. Spatio-temporal signal processing for multi-subject functional MRI studies. In *Proc. IEEE International Conference on Aucoustics, Speech and Signal Processing (ICASSP'01)*, Salt Lake City, Utah, USA, 2001.

[141] V. Solo and S. Ratcliffe. Rotated functional principal components analysis with regression. Technical report, University of New South Wales, 2002.

[142] C Stein. Estimation of the mean of a multivariate normal distribution. *Ann. Stat.*, 9(6):1135–1151, 1981.

[143] R.L. Stevenson, B.E. Schmitz, and E.J. Delp. Discontinuity preserving regularization of inverse visual problems. *IEEE Trans. Syst. Man Cybern*, 24(3):455–469, 1994.

[144] G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesley - Cambridge Press, Wellesley MA, 1997.

[145] S.Z.Li. On discontinuity-adaptive smoothness priors in computer vision. *IEEE Trans. Pat. Anal and Mach. Intel.*, 17(6), 1995.

[146] J. Talairach and P. Tournoux. *Co-planar Steriotaxic Atlas of the Human Brain: 3-Dimensional Proportional System - an approach to Cerebral Imaging.* Thiene Medical Publisher, New York, NY, 1988.

[147] C.M. Theobald. An inequality with application to multivariate analysis. *Biometrika*, 62(2):461–466, 1975.

[148] B. Thirion and O Faugeras. Dynamical components analysis of fMRI data through kernel PCA. *NeuroImage*, 20:34–49, 2003.

[149] C.G. Thomas, R. A. Harshman, and R.S. Menon. Noise reduction in BOLD-based fMRI using component analysis. *Neuroimage*, 17:1521–1537, 2002.

[150] R Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Stat. Soc., Series B*, 58(1):267–288, 1996.

[151] R. Tibshirani and T. Hastie. Local likelihood estimation. *J. Amer. Statist. Assn*, 82:559–567, 1987.

[152] M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *J. Royal Stat. Soc., Series B*, 61(3):611–622, 1999.

[153] N.T. Trendafilov and I.T. Jolliffe. Projected gradient approach to the numerical solution of the SCoTLASS. *Comp. Stat. Data Anal.*, 50:242–253, 2006.

[154] B. Turlach, W. Venables, and S. Wright. Simultaneous variable selection. *Technometrics*, 47(3):349–363, 2005.

[155] K. Ugurbil, L. Toth, and D-S Kim. How accurate is magnetic resonance imaging of brain function. *TRENDS in Neurosciences*, 26(2), 2003.

[156] M.O. Ulfarsson and V. Solo. Smooth principal component analysis with application to functional magnetic resonance imaging. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'06)*, volume 2, pages II–993 – II–996, Toulouse, France, May 2006.

[157] M.O. Ulfarsson and V. Solo. Spatially local and temporally smooth PCA for fMRI. In *Proc. IEEE International Conference on Image Processing (ICIP'06)*, Atlanta, Georgia, USA, 2006.

[158] M.O. Ulfarsson and V. Solo. Sparse variable principal component analysis with application to fMRI. In *Proc. IEEE International Symposium on Biomedical Imaging (ISBI'07)*, Washington D.C., 2007.

[159] M.O. Ulfarsson and V. Solo. Rank selection in noisy PCA with SURE and random matrix theory. *IEEE Trans. Signal Proc.*, submitted for publication.

[160] M.O. Ulfarsson and V. Solo. Sparse variable noisy PCA using geodesic descent. *IEEE Trans. Signal Proc.*, submitted for publication.

[161] A. Vazquez and D. Noll. Nonlinear aspects of the BOLD response in functional MRI. *Neuroimage*, 7:108–118, 1998.

[162] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding.* Prentice Hall Inc., Englewood Cliffs N.J., 1996.

[163] R. Viviani, G. Gron, and M. Spitzer. Functional principal component analysis of fMRI data. *Human Brain Mapping*, 24:109–129, 2005.

[164] C.R. Vogel and M.E. Oman. Iterative methods for total variation denoising. *SIAM J. Sci. Comp.*, 17(1):227–238, 1996.

[165] M. Wax and T. Kailath. Detection of signals by information theoretic criteria. *IEEE Trans. Acoustics, Speech and Signal Proc.*, ASSP-33(2), 1985.

[166] R.M. Weisskoff, J. Baker, J. Belliveau, T.L. Davis, K.K. Kwong, M. Cohen, and B.R. Rosen. Power spectrum analysis of functionally weighted MR data: what's in the noise. In *SMRM*, volume 12, page 7, New York, NY, 1993.

[167] P. Whittle. On principal components and least square methods of factor analysis. *Scandinavisk Aktuarietidskrift*, 36:223–239, 1953.

[168] S. Wold. Cross-validatory estimation of the number of components in factor and principal components model. *Technometrics*, 20(4), 1978.

[169] K. Worsley, C. Liao, J. Aston, V. Petre, G. Duncan, F. Morales, and A. Evans. A general statistical analysis for fMRI data. *Neuroimage*, 15:1–15, 2002.

[170] K.J. Worsley, J. Chen, J. Lerch, and A.C. Evans. Comparing functional connectivity via thresholding correlations and SVD. *Phil Trans R Soc Lond B Biol Sci*, 360(1457):913–920, 2005.

[171] C.F.J. Wu. On the convergence properties of the EM algorithm. *Ann. Statist.*, 11:95–103, 1983.

[172] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J.R. Statist. Soc. B.*, 68:49–67, 2006.

[173] W.I. Zangwill. *Nonlinear Programming: A Unified Approach.* Prentice-Hall, 1969.

[174] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J.R. Statist. Soc. B*, 67(2):301–320, 2005.

[175] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *J. Comput. Graphical Stat.*, 15(2):265–286, 2006.