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

**Shaojie Chen, Joshua Vogelstein, Seonjoo Lee, Martin Lindquist, Brian Caffo**

June 2, 2015

## Abstract

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

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

# 1 Introduction

In the past decade functional magnetic resonance imaging (fMRI) has given researchers unprecedented access to the brain in action and provided numerous insights into human brain function. Any given fMRI experiment generates massive amounts of data. For example, a standard experiment collects a few hundred 3D brain images, each consisting of roughly 100,000 uniformly spaced volume elements (voxels) that partition the brain. Intensity values from each individual voxel can be extracted to create a set of time series of length T, where T corresponds to the number of acquired images. The analysis of fMRI data can therefore fruitfully be viewed as a multivariate time series problem. However, the signal of interest is relatively weak and the data exhibits a complicated temporal and spatial noise structure [15].

To date numerous statistical methods have been applied to fMRI data. Many construct separate univariate models at each voxel, thus assuming an improbable independence between voxels. In this work we instead focus on the multivariate statistical methods that have been used to analyze fMRI data. In particular, multivariate decomposition methods, such as Principal Components Analysis (PCA) [2] and Independent Components Analysis (ICA) [6], have been utilized to identify patterns of brain activation [16].

Interestingly, several of these commonly applied statistical techniques for modeling both multivariate data can be seen as variants of state-space models (SSMs). For example, according to Roweis and Ghahramani [22], factor analysis, principal component analysis (PCA), mixture of Gaussian clusters, independent component analysis (ICA), Kalman filter models and hidden Markov models (HMMs) can all be viewed as special cases of SSMs.

In the time-invariant linear case, an SSM is also referred to as a linear dynamical system (LDS) or linear Gaussian model (LGM). In this work, LDS and its extensions are discussed, so we will use LDS and SSM interchangeably in the following sections. The LDS can be seen as a continuous-state analogue of the hidden Markov model (HMM) [20]. The forward step of the forward-backward algorithm used to inference HMMs is equivalent to the well-known Kalman filter used in LDS, and similarly the backward step can be computed using Rauchs recursion [21]. Together these two steps can be employed to perform inference on the posterior probabilities of latent states given the observed sequence.

Likewise, factor analysis and PCA can each be derived from the LDS by applying particular constraints on the latent states dynamics coefficients and the observation error covariance matrix. Specifically, by constraining the latent states dynamics coefficients to $\mathbf{0}$, one gets a static model. Factor analysis can be implemented by further constraining the observation error covariance matrix to be diagonal. PCA can be applied by forcing the observation error covariance matrix to be a multiple of the identity matrix approaching $\mathbf{0}$. A corresponding

detailed review can be found in Roweis and Ghahramani [22].

Finally, LDS can also be represented as a probabilistic graphical model. Here the Kalman filter and smoother are special cases of the belief propagation algorithm that has been developed to analyze general graphical models [14][19].

Because of their flexibility state-space models have found wide usage in a number of different spheres, including time series analysis, statistics, signal processing, control theory and machine learning. In neuroimaging analysis, the LDS exhibits substantial relevance. For example, Harini et al have discussed the applications of HMM in learning functional network dynamics in resting state fMRI [9]. Valdez-Sosa et al used sparse multivariate autoregression to estimate brain functional connectivity [27]. Havlicek et al modeled neuronal responses in fMRI using cubature Kalman filtering along with Kalman filter based Dynamic Granger Causality to evaluate functional connectivity in fMRI data [11]. A systematic framework for functional connectivity measures is proposed by HE Wang et al [29].

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

## 2 The Model

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

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

where $A$ is the $d \times d$ state transition matrix and $C$ is the $p \times d$ generative matrix. $\mathbf{x}_t$ is a $d \times 1$ vector and $\mathbf{y}_t$ is a $p \times 1$ vector. The sequence of vectors $\{\mathbf{y}\} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ are the observed data and $\{\mathbf{x}\} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ represent the unknown hidden states. The output noise covariance $R$ is $p \times p$, while the state noise covariance $Q$ is $d \times d$. Initial state mean $\pi_0$ is $d \times 1$ and covariance $V_0$ is $d \times d$.

Without applying further constraints, the model itself is unidentifiable. Supplemental constraints are are thus introduced to address both identifiability and utility. Three basic constraints are required to make the

model identifiable:

**Constraint 1:** $Q$ is the identity matrix

**Constraint 2:** the ordering of the columns of $C$ is fixed based on their norms

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

Note that the first two constraints follow directly from Roweis and Ghahramani (1999) [22].

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

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

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

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

**Constraint 4:** $R$ is a diagonal matrix

**Constraint 5:** $A$ is sparse

**Constraint 6:** $C$ has smooth columns

Consider the case where the observed data are high dimensional and the $R$ matrix is very large. One can not accurately estimate the many free parameters in $R$ with limited observed data. Therefore some constraints on $R$ will help with inferential accuracy, by virtue of significantly reducing variance while not adding too much bias. In the simplest case, $R$ is set to an identity matrix or its multiple. More generally, one can also constrain matrix $R$ to be diagonal. In the static model with no temporal dynamics, a diagonal $R$ is equivalent to the generic Factor Analysis method, while multiples of the identity $R$ matrix lead to Principal Component Analysis (PCA) [22].

The $A$ matrix is the transition matrix of the hidden states. In our application, it is a central construct of

interest representing a so-called connectivity graph. In many applications, it is desirable for this graph to be sparse. In this work, an L-1 penalty term on $A$ is used to impose sparsity on the connectivity graph..

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

With all those constraints, the model becomes:

$$\begin{aligned} &\mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{w}_t, \quad \mathbf{w}_t \sim N(\mathbf{0}, \mathbf{I}), \quad \mathbf{x}_0 = \pi_0, \quad A \text{ is sparse} \\ &\mathbf{y}_t = C\mathbf{x}_t + \mathbf{v}_t, \qquad \mathbf{v}_t \sim N(\mathbf{0}, R), \quad C \text{ has smooth columns} \end{aligned} \tag{2}$$

For notational convenience, a sequence of $T$ output vectors $(\mathbf{y}_1, \ldots, \mathbf{y}_T)$ is denoted by $\{\mathbf{y}\}$; a subsequence $(\mathbf{y}_{t_0}, \mathbf{y}_{t_0+1}, \ldots, \mathbf{y}_{t_1})$ by $\{\mathbf{y}\}_{t_0}^{t_1}$. Similarly for the latent states. In addition, let $\Theta = \{A, C, R, \pi_0\}$ represents all unknown parameters and $P(\{\mathbf{x}\}, \{\mathbf{y}\})$ be the likelihood for a generic LDS model, then model (2) is equivalent to

$$\hat{\Theta} = \arg\min_{\Theta} \left\{ -\log P(\{\mathbf{x}\}, \{\mathbf{y}\}) + \lambda_1 \|A\|_1 + \lambda_2 \|C\|_2^2 \right\} \tag{3}$$

where $\lambda_1$ and $\lambda_2$ are tuning parameters and $\| \, \mathbf{.} \, \|_p$ represents the $p$-norm of a vector.

# 3 Parameter Estimation

The motivating application requires solving optimization problem (3): given only an observed sequence (or multiple sequences in some applications) of outputs $\{\mathbf{y}\} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$, find the parameters $\Theta = \{A, C, R, \pi_0\}$ that maximize the likelihood of the observed data.

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

Specifically, one way to search for the maximum likelihood solution is through iterative techniques such as expectation maximization (EM) [23]. The detailed EM steps for a generic LDS can be found in Zoubin and Geoffrey (1996) [10]. An alternative approach is to use subspace identification methods such as N4SID and

PCA-ID to compute an asymptotically unbiased solution in closed form [28] [8]. In practice, determining an initial solution with subspace identification and then refining the solution with EM is an effective approach [5].

However, the above solutions can not be directly applied to optimization problem (3) due to the introduced penalty terms. We therefore developed a novel algorithm called Reduced Rank M-Estimation for High-Dimensional Linear Dynamical System Identification (R2SID), as detailed in the following.

By the chain rule, the likelihood in model (2) is

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

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

$$\begin{aligned} P(\mathbf{y}_t|\mathbf{x}_t) &= (2\pi)^{-\frac{p}{2}}|R|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}[\mathbf{y}_t - C\mathbf{x}_t]^T R^{-1}[\mathbf{y}_t - C\mathbf{x}_t]\right\} \\ P(\mathbf{x}_t|\mathbf{x}_{t-1}) &= (2\pi)^{-\frac{d}{2}}\exp\left\{-\frac{1}{2}[\mathbf{x}_t - A\mathbf{x}_{t-1}]^T[\mathbf{x}_t - A\mathbf{x}_{t-1}]\right\}. \end{aligned}$$

Then the log-likelihood, after dropping a constant, is just a sum of quadratic terms

$$\begin{aligned} \log P(\{\mathbf{x}\},\{\mathbf{y}\}) = &-\sum_{t=1}^{T}\big(\frac{1}{2}[\mathbf{y}_t - C\mathbf{x}_t]^T R^{-1}[\mathbf{y}_t - C\mathbf{x}_t]\big) - \frac{T}{2}\log|R| \\ &-\sum_{t=1}^{T}\big(\frac{1}{2}[\mathbf{x}_t - A\mathbf{x}_{t-1}]^T[\mathbf{x}_t - A\mathbf{x}_{t-1}]\big) - \frac{T}{2}\log|\mathbf{I}| + \log(\mathbb{1}_{\pi_0}(\mathbf{x}_0)). \end{aligned} \tag{4}$$

Replace $\log P(\{\mathbf{x}\},\{\mathbf{y}\})$ with equation (4), model (3) is

$$\begin{aligned} \hat{\Theta} = \arg\min_{\Theta}\Big\{ &\sum_{t=1}^{T}\big(\frac{1}{2}[\mathbf{y}_t - C\mathbf{x}_t]^T R^{-1}[\mathbf{y}_t - C\mathbf{x}_t]\big) - \frac{T}{2}\log|R| \\ &+\sum_{t=1}^{T}\big(\frac{1}{2}[\mathbf{x}_t - A\mathbf{x}_{t-1}]^T[\mathbf{x}_t - A\mathbf{x}_{t-1}]\big) - \frac{T}{2}\log|\mathbf{I}| - \log(\mathbb{1}_{\pi_0}(\mathbf{x}_0)) \\ &+\lambda_1\|A\|_1 + \lambda_2\|C\|_2^2\Big\}. \end{aligned} \tag{5}$$

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

### 3.1 E Step

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

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

This quantity depends on three expectations: $E[\mathbf{x}_t|\{\mathbf{y}\}]$, $E[\mathbf{x}_t\mathbf{x}_t^T|\{\mathbf{y}\}]$ and $E[\mathbf{x}_t\mathbf{x}_{t-1}^T|\{\mathbf{y}\}]$. We denote them by the symbols:

$$\hat{\mathbf{x}}_t \equiv E[\mathbf{x_t}|\{\mathbf{y}\}],\ P_t \equiv E[\mathbf{x}_t\mathbf{x}_t^T|\{\mathbf{y}\}],\ P_{t,t-1} \equiv E[\mathbf{x}_t\mathbf{x}_{t-1}^T|\{\mathbf{y}\}]. \tag{6}$$

Expectations (6) are calculated with a Kalman filter/smoother, which is detailed in Appendix 1.

### 3.2 M Step

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

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

$$\begin{aligned}
&\frac{\partial \mathbf{\Phi}}{\partial R^{-1}} = \frac{T}{2}R - \sum_{t=1}^{T}\big(\frac{1}{2}\mathbf{y}_t\mathbf{y}_t^T - C\hat{\mathbf{x}}_t\mathbf{y}_t^T + \frac{1}{2}CP_tC^T\big) = 0 \\
&R = \frac{1}{T}\sum_{t=1}^{T}(\mathbf{y}_t\mathbf{y}_t^T - C^{\text{new}}\hat{\mathbf{x}}_t\mathbf{y}_t^T) \\
&R^{\text{new}} = \text{Diag}\left\{\frac{1}{T}\sum_{t=1}^{T}(\mathbf{y}_t\mathbf{y}_t^T - C\hat{\mathbf{x}}_t\mathbf{y}_t^T)\right\}
\end{aligned} \tag{7}$$

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

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

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

Estimation for transition matrix $C$ also has closed form solution, and the solution can be derived by rear-

ranging the terms properly. Terms relevant to $C$ in equation (5) are

$$f_{\lambda_2}(C;\{\mathbf{x}\},\{\mathbf{y}\}) = \sum_{t=1}^{T}\left(\frac{1}{2}[\mathbf{y}_t - C\mathbf{x}_t]^T R^{-1}[\mathbf{y}_t - C\mathbf{x}_t]\right) + \lambda_2\|C\|_2. \tag{8}$$

In $f_{\lambda_2}(C;\{\mathbf{x}\},\{\mathbf{y}\})$, $C$ is a matrix and need to be vectorized for optimization. Here we follow the methods of Turlach et al [26]. Without loss of generality, assume $R$ is the identity matrix in equation (8); otherwise, one can always write equation (8) as

$$\sum_{t=1}^{T}\left(\frac{1}{2}[R^{-\frac{1}{2}}y_t - R^{-\frac{1}{2}}Cx_t]^T[R^{-\frac{1}{2}y_t} - R^{-\frac{1}{2}}Cx_t]\right) + \lambda_2\|R^{-\frac{1}{2}}C\|$$

Let

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

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

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

where $W = \left(\mathbf{x}_1,\ldots,\mathbf{x}_T\right)$. Finally, vectorize $C^{\text{old}}$ as

$$\mathbf{c}^{\text{old}} = (C_{11}^{\text{old}},\ldots,C_{1d}^{\text{old}},C_{21}^{\text{old}},\ldots,C_{2d}^{\text{old}},C_{p1}^{\text{old}},\ldots,C_{pd}^{\text{old}})^T \tag{9}$$

where $C_{ij}$ is the element at row $i$ and column $j$ of $C$. With these new notations, the equation (8) is equivalent to

$$f_{\lambda_2}(C;\{\mathbf{x}\},\{\mathbf{y}\}) = \|\mathbf{w} - \mathbf{W}\mathbf{c}\|_2^2 + \lambda_2\|\mathbf{c}\|_2^2. \tag{10}$$

With the Tikhonov regularization [25], equation (10) has closed form solution

$$\begin{aligned} \mathbf{c}^{\text{new}} &= (\mathbf{W}^T\mathbf{W} + \lambda_2\mathbf{I})^{-1}\mathbf{W}^T\mathbf{w} \\ C^{\text{new}} &= \text{Rearrange } \mathbf{c}^{\text{new}} \text{ by equation } (9) \end{aligned} \tag{11}$$

Now let's look at parameter $A$. Terms involving $A$ in equation (5) are,

$$f_{\lambda_1}(A;\{\mathbf{x}\},\{\mathbf{y}\}) = \sum_{t=1}^{T}\big(\frac{1}{2}[\mathbf{x}_t - A\mathbf{x}_{t-1}]^T[\mathbf{x}_t - A\mathbf{x}_{t-1}]\big) + \lambda_1\|A\|_1. \tag{12}$$

Similar to what we have done to $C$, equation (12) is equivalent to

$$f_{\lambda_1}(A;\{\mathbf{x}\},\{\mathbf{y}\}) = \|\mathbf{z} - \mathbf{Z}\mathbf{a}\|_2^2 + \lambda_1\|\mathbf{a}\|_1. \tag{13}$$

where $\mathbf{z}$ is a $Td \times 1$ vector from rearranging $\{\mathbf{x}\}$ and $\mathbf{Z}$ is a block diagonal matrix with diagonal component $Z^T = (\mathbf{x}_0, \ldots, \mathbf{x}_{T-1})^T$. Unfortunately, equation (13) does not have closed form solution due to the $L-1$ term.

Though not having a closed form solution, $f_{\lambda_1}(A;\{\mathbf{x}\},\{\mathbf{y}\})$ can be solved numerically with a Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [4]. FISTA is an accelerated version of the Iterative Shrinkage-Threshholding Algorithm (ISTA) [7]. ISTA is linearly convergent while FISTA is quadratic convergent. Steps of a general FISTA algorithm can be found in Appendix 2.

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

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

With FISTA and $L$, matrix $A$ can be updated:

$$A^{\text{new}} = \text{FISTA}(\|\mathbf{Z}^T\mathbf{a}^{\text{old}} - \mathbf{z}\|_2^2, \quad \lambda_1) \tag{14}$$

## 3.3 The Complete EM

The complete EM algorithm for PLDS is addressed as follows.

**Algorithm** EM Algorithm for PLDS

**M Step**

1. $R^{\text{new}} = \text{Diag}\left\{ \frac{1}{T} \sum_{t=1}^{T} (\mathbf{y}_t \mathbf{y}_t^T - C^{\text{old}} \hat{\mathbf{x}}_t \mathbf{y}_t^T) \right\}$, as in equation (7)

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

3. Update $C^{\text{new}}$, as in equation (11)

4. Update $A^{\text{new}}$ with FISTA, as in equation (14)

**E Step**

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

1. Update the expectations in (6) with the Kalman filter smoother in Appendix 1

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

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

$$\mathbf{Y} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathbf{T}} \approx \mathbf{U}_{p\times d}\mathbf{D}_{d\times d}\mathbf{V}_{d\times T}^T = \mathbf{U}_{p\times d}\mathbf{X}_{d\times T} \tag{15}$$

where $\mathbf{U}_{p\times d}$ is the first $d$ columns of $\mathbf{U}$ and $\mathbf{D}_{d\times d}$ is the upper left block of $\mathbf{D}$. This notation also applies to $\mathbf{V}_{d\times T}^T$.

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

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

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

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

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

$$K_t = V_t^{t-1} C^T (C V_t^{t-1} C^T + R)^{-1}$$

which involves the inverse of a large square matrix of dimension p by p. As stated previously, such a matrix ex-

ceeds available memory when p is high. The Woodbury Matrix Identity is employed to turn a high dimensional inverse to low dimensional problem:

$$(CV_t^{t-1}C^T + R)^{-1} = R^{-1} - R^{-1}C[(V_t^{t-1})^{-1} + C^T R^{-1} C]^{-1} C^T R^{-1}$$

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

# 4 Result

## 4.1 Parameter Estimation

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

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

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

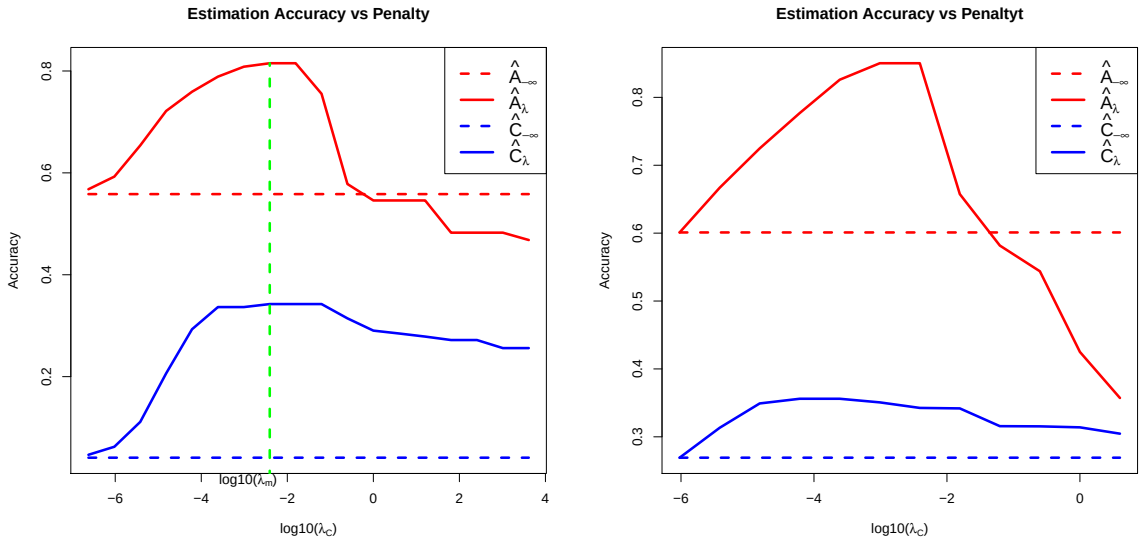
To evaluate the accuracy of estimations, some distance measure should be first defined. Here the distance between two matrices $A$ and $B$ is defined as follows

$$d(A, B) = -\log(\frac{1}{n} \max_{P \in P(n)} \text{Tr}\big(P \times C_{A,B}\big))$$

where $C_{A,B}$ is the correlation matrix between columns of A and B, $P(n)$ is the collection of all the permutation matrices of order n and $P$ is a permutation matrix.

As a result of the way it's defined, $d(A, B)$ is invariant to the scales of columns of $A$ and $B$. It is also invariant to a permutation of columns of either matrix. The calculation of $d(A, B)$ is exactly a linear assignment problem and can be solved in polynomial time with the Hungarian algorithm [12].

Both the generic LDS and the penalized LDS are applied to the simulation data. As the true parameters




(a) Low dimensional setting

(b) High dimensional setting

Figure 1: x axis is tuning parameter $\lambda_C$ under log scale and y axis is the distance between truth and estimations; $\lambda_A$ is increasing proportionally with $\lambda_C$

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

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

Estimation accuracies are plotted against penalty size $\lambda_C$ in Figure 1. Results from LDS and PLDS are overlayed in one plot for comparison. As the figure shows, PLDS converges to the LDS when the penalties are approaching zero. Estimation accuracies first increase with penalty size and then decrease due to over-shrinkage.

As a concrete example, estimations from both methods are compared to the true values of parameters in Figure 2. One can see that true values in each column of $C$ matrix are decreasing smoothly. $\hat{C}_{\lambda_m}$, which is estimated with optimal penalties $\lambda_C = \lambda_m$ and $\lambda_A = k\lambda_m$, shows similar pattern. In terms of $A$, the true value is sparse with many 0 (blue) values. PLDS estimation $\hat{A}_{\lambda_m}$ is also sparse, denoted by the off-diagonal blue values. However, LDS estimation $\hat{A}_{\lambda_{-\infty}}$ is not sparse, with many yellow and red off-diagonal values.

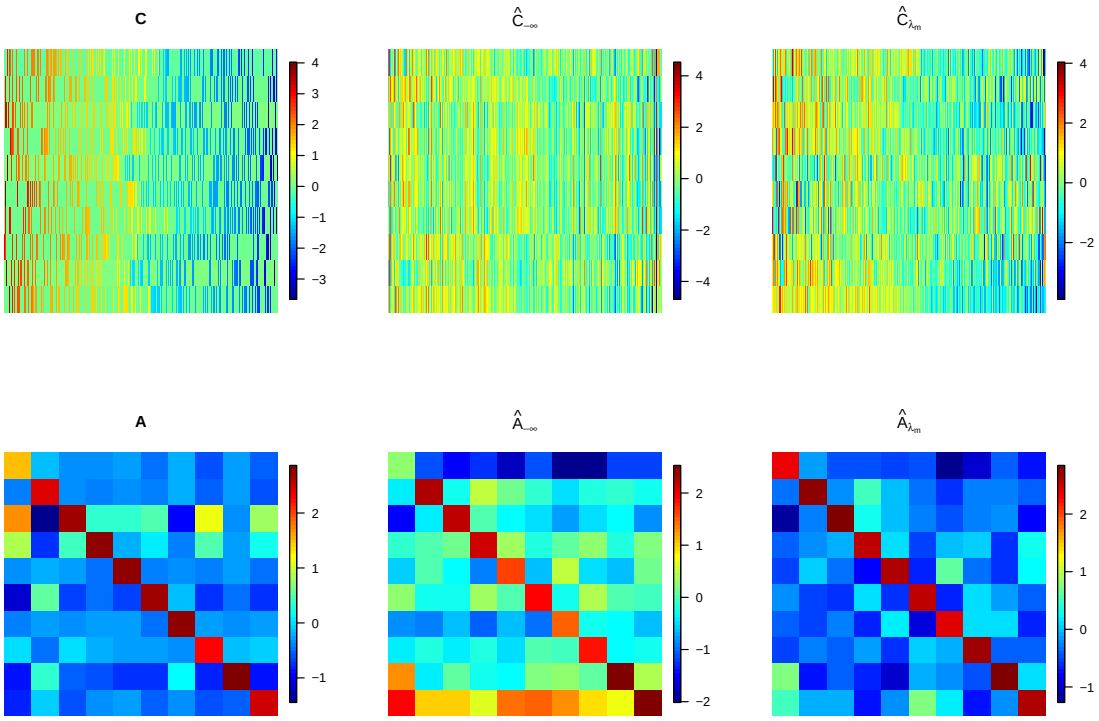



Figure 2: Row 1: A truth; non-penalized estimation of A; optimally penalized estimation of A. Row 2: C truth; non-penalized estimation of C; optimally penalized estimation of C.

In addition to the improved estimation accuracy, the proposed algorithm is also computational efficiency and highly scalable. As a demonstrate, we measure the running times of multiple simulation scenarios and summarize them in Table 1. When both $p$ and $d$ are high dimensional, the algorithm can still solve the problem in a reasonable time.

Table 1: PLDS Running Time

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

## 4.2 Making Predictions

Another perspective when considering the PLDS model is its ability to make predictions. When the parameters $\mathbf{\Theta}$ and the latent states $x_T$ are estimated, one can first use estimated $x_T$ to predict $x_{T+1}$ and use $x_{T+1}$ to predict $y_{T+1}$. Similarly, more predictions $y_{T+2}, \ldots, y_{T+k}$ can be made. Intuitively, properly chosen penalties

give better estimations and good estimations should give more accurate predictions. This idea is demonstrated with a simulation. The parameter settings for this simulation follow Section 4.1. The correlation between the predicted signal and true signal is used as a measure of prediction accuracy. The prediction accuracy over penalty size is shown in Figure 3.

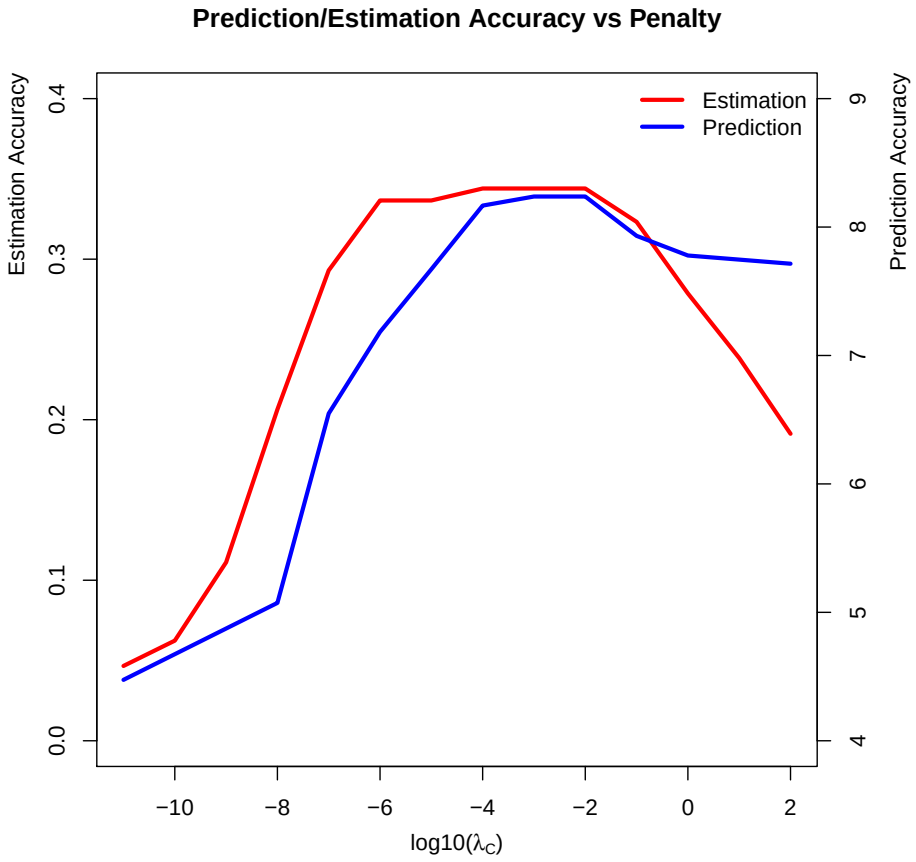



Figure 3: Estimation and prediction accuracies.

Observations and findings from these plots include:

- The prediction accuracy first improves then drops when the penalties increase
- The prediction accuracy peaks when the penalty coefficient $\lambda_A$ and $\lambda_C$ are around $10^{-3}$. This make sense as the same $\lambda$ pair also gives the best estimation for coefficients $A$ and $C$, as in Figure 1.

This second observation provides us a way to pick tuning parameters in real applications, as detailed in Section 5.

# 5 Application

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

The PLDS is applied to analyze the motor cortex of human brains from the KIRBY 21 Data. These data are resting-state fMRI scans consisting of a test-retest dataset previously acquired at the FM Kirby Research Center at the Kennedy Krieger Institute, Johns Hopkins University [13]. Twenty-one healthy volunteers with no history of neurological disease each underwent two separate resting state fMRI sessions on the same scanner: a 3T MR scanner utilizing a body coil with a 2D echoplanar (EPI) sequence and eight channel phased array SENSitivity Encoding (SENSE; factor of 2) with the following parameters: TR 2s; 3mm$\times$3mm in plane resolution; slice gap 1mm; and total imaging time of 7 minutes and 14 seconds.

In this application, test-retest scans from two subjects are analyzed. The imaging data are first preprocessed with FSL, a comprehensive library of analysis tools for fMRI, MRI and DTI brain imaging data [24]. FSL is used for spatial smoothing with Gaussian kernel. Then PLDS is applied on the smoothed data.

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

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

The number of latent states, d, can be manually selected based on related research that maps the primary motor region to human activities. For instance, Meier et. al mapped the motor region to 9 human organs: tongue, lips, squint, fingers, wrist, forearm, elbow, foot and saccade [17].

A more flexible technique to choose the number of latent states involves the profile likelihood method proposed by Zhu et. al [30]. As a first step, eigenvalues of the data matrix are calculated with Principal Component Analysis (PCA). The cumulative eigenvalues as a percentage of the sum of all eigenvalues are then plotted - see Figure 4. Visually one notes that the first 10 eigenvalues take over $80\%$ of all variations. The number of latent states can be selected as the smallest number (of eigenvalues) that explains over $80\%$ of total

variation in the data. However, the drawback of this method is clear: the choice of threshold percentage (here $80\%$) is highly subjective. The profile likelihood method overcomes this problem and could pick the dimension automatically.

The above method assumes that the first $k$ eigen-values are samples from a Gaussian distribution $N(\mu_1, \sigma^2)$, while the rest are from a different Gaussian distribution $N(\mu_2, \sigma^2)$. Then the profile likelihood can be calculated given $k$, for all $k = 1, \cdots, T$ and selecting the optimal $k$ as the one with the highest profile likelihood. As shown in Figure 4, when the profile likilihood method is applied to the first scan of subject one, d = 11 is selected. Apply the method to all four scans, the numbers of latents states selected are 6, 11, 14 and 15 respectively. Their average, $d = 11$, is used.

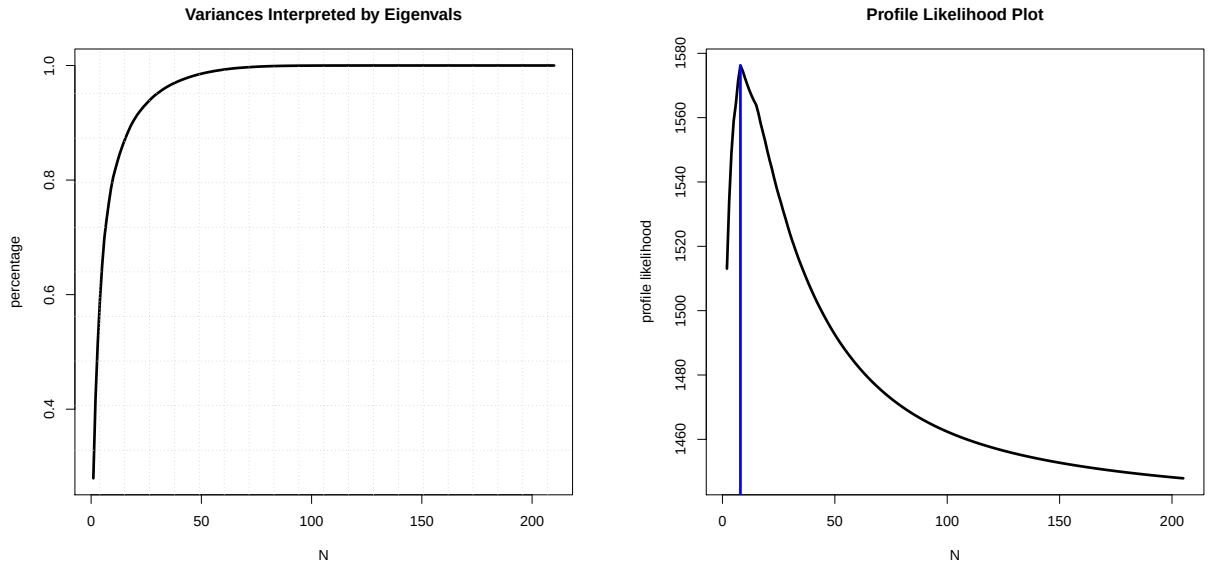



Figure 4: Eigen-values and Corresponding Profile Likelihood Plot

The $A$ matrices as connectivity graphs are first plotted in Figure 5. One can group the scans correctly with the $A$ matrices. Specifically, denote the $A$ matrix estimation for the first scan of subject one as $A_{11}$. Similary notations apply to the other scans. Then the canonical correlations among the four matrices are summarized in Table 2. Another permutation invariant measure of square matrix similarity, the Amari error, is also provided in the table [1]. Notice a higher $d(A, B)$ or a smaller Amari error means more similarity. From both measures, one can group the four scans correctly. This implies that the graph contains subject-specific information.

Table 2: Similarities Among Estimated $A$ Matrices

| $d(\cdot,\cdot)$(Amari Error) | $A_{11}$ | $A_{12}$ | $A_{21}$ | $A_{22}$ |
|---|---|---|---|---|
| $A_{11}$ | 0 | | | |
| $A_{12}$ | $\mathbf{0.076(0.88)}$ | 0 | | |
| $A_{21}$ | 0.105(1.05) | 0.095(1.08) | 0 | |
| $A_{22}$ | 0.095(1.02) | 0.095(1.09) | $\mathbf{0.085(0.98)}$ | 0 |

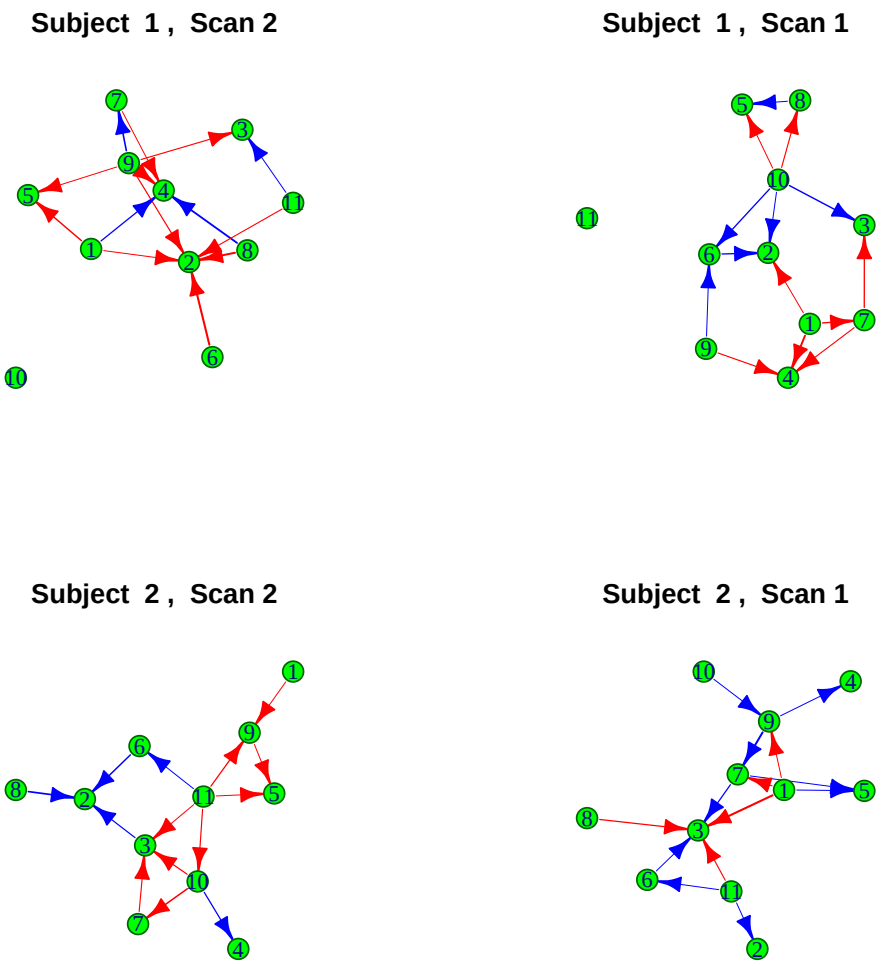



Figure 5: Connectivity Graph: The wider edge means stronger connectivity; the red edge means negative connectivity and blue edge means positive connectivity.

The similarities are also shown in Figure (6).

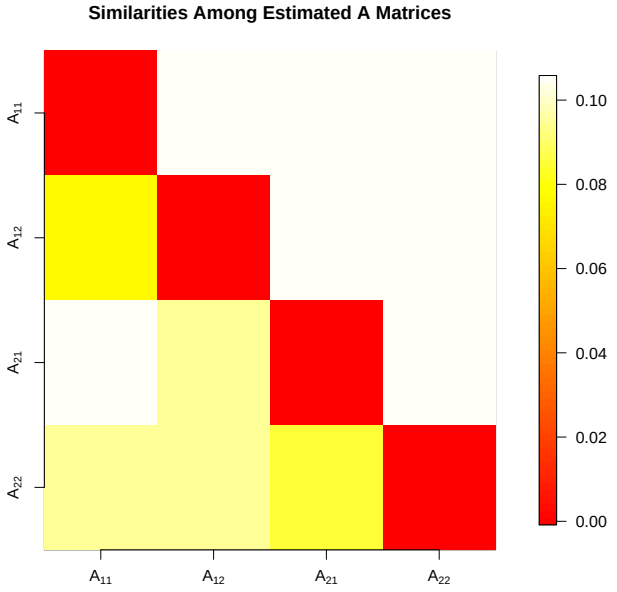



Figure 6: Similarities among the four estimated $A$ matrices. The distance $d(\cdot, \cdot)$ is used in this figure. As one can see, the two yellow off-diagonal pixels has the minimum distances, which correspond to the pairs of $(A_{11}, A_{12})$ and $(A_{21}, A_{22})$ respectively.

As an example, the 3D renderings of the columns of matrix $C$ from the first scan of subject one are shown in Figure 7 (after thresholding). The biological meaning of those regions need to be further validated. It is helpful to compare those regions to other existing parcellations of the mortor cortex. As an example, the blue region above accurately matches the DM (dorselmedical) parcel of the five-region parcellation proposed by Nebel MB et al. [18].

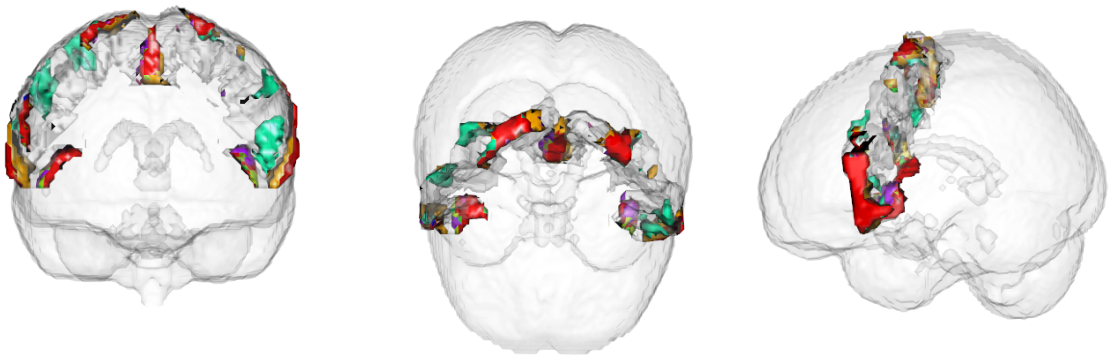


Figure 7: 3D Rendering of Columns of Matrix $C$

Another application of the algorithm is predicting brain signals. To demonstrate this, the algorithm is applied to the Human Connectome Project (HCP) data. (**Descriptions of data and citations required here**) Using the profile likelihood method, $d = 149$ is picked. The data has $T = 1200$ time points. The first $N = 1000$ are picked as train data, while the rest are used as test data. Then both the SVD method in Equation and the PLDS algorithm are used for prediction. The prediction accuracies are shown in Figure 8. The first observation is that, the PLDS algorithm is giving significantly better predictions for the first 150 predictions compared to the SVD method. As the SVD method is also used to intialize the PLDS algorithm, this shows that the PLDS algorithm improves estimations from the SVD method in terms of short-term predictions. Another observation is, the PLDS algorithm's performance get worse when one predicts into the "long" future ($> 150$ steps). This is reasonable, as there is no way that we can predict the two noise terms in the model, therefore the prediction errors from each step will accumulate and yields deteriorating predictions.

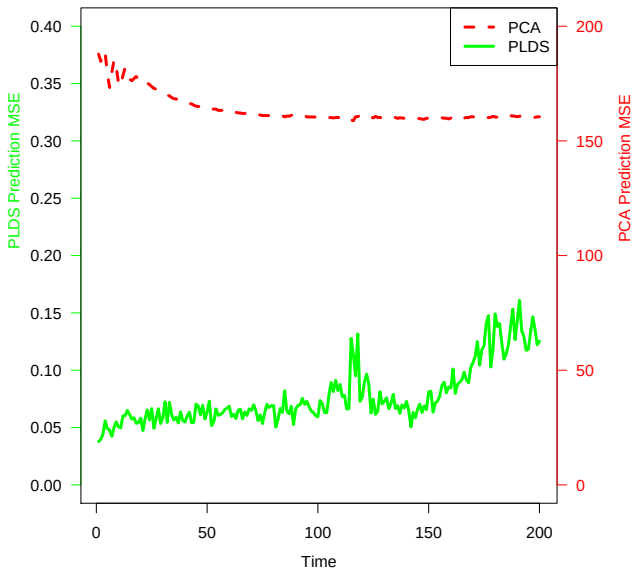



Figure 8: Prediction accuracies comparison on HCP data

A sample plot of the true time series and predicted values are shown in Figure 9. We see that the PLDS is giving more accurate predictions and the true signal lies in the confidence band giving ty the PLDS model. Another observation is that the confidence band is getting wider as we predict into the future, which is a result of the accumulated errors.

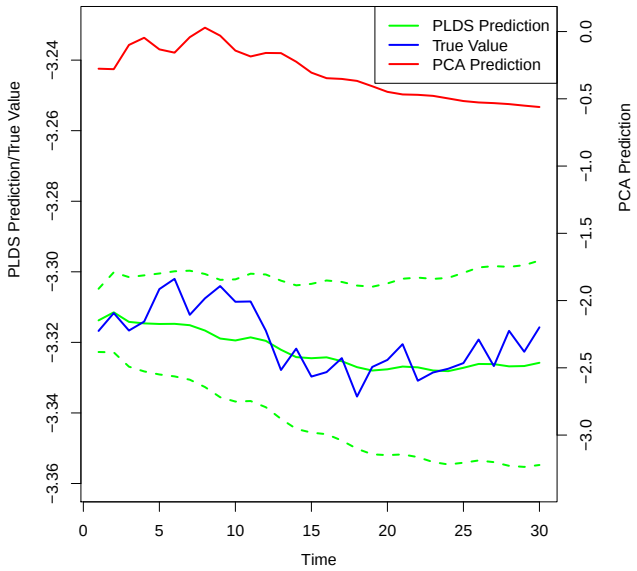



Figure 9: Sample time series plot. The dotted green curve stands for the $60\%$ confidence band given by the PLDS model. The time series is the average of signals from 20 randomly picked voxels.

# 6 Discussion

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

In the future, this work could be extended in two important directions. First, assumptions on the covariance structures in the observation equation could be generalized. Prior knowledge could be incorporated to covariance $R$. The general rule is that $R$ should be general enough to be flexible while sufficiently restricted to make the model useful. A lot of other platforms such as tridiagnol and upper triangular could also be considered. Mohammad et. al have discussed the impact of auto correlation on functional connectivity, which also provides us a direction for extension [3].

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

# Appendix 1

**Algorithm** Standard Kalman Filter Smoother for estimating the moments required in the E-step of an EM algorithm for a linear dynamical system

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

1. Forward Recursions:

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

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

$K_t = \mathbf{V}_t^{t-1} C^T (CV_t^{t-1} C^T + R)^{-1}$

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

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

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

2. Backward Recursions:

$J_{t-1} = V_{t-1}^{t-1} A^T (V_t^{t-1})^{-1}$

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

$V_{t-1}^T = V_{t-1}^{t-1} + J_{t-1}(V_t^T - V_t^{t-1})J_{t-1}^T$

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

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

# Appendix 2

In general, FISTA optimize a target function

$$\min_{x\in\mathcal{X}} \quad \mathbf{F}(\mathbf{x};\lambda) = \mathbf{g}(\mathbf{x}) + \lambda\|\mathbf{x}\|_{\mathbf{1}} \tag{16}$$

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

A FISTA algorithm with constant step is detailed below

---

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

---

1. Input an initial guess $\mathbf{x_0}$ and Lipschitz constant $\mathbf{L}$ for $\nabla\mathbf{g}$, set $\mathbf{y_1} = \mathbf{x_0}, t_1 = 1$
2. Choose $\tau \in (0, 1/\mathbf{L}]$.
3. Set $k \leftarrow 0$.
4. **loop**
5. Evaluate $\nabla\mathbf{g}(\mathbf{y_k})$
6. Compute $\mathbf{x_1}$= $\mathbf{S}_{\tau\lambda}(\mathbf{y_k} - \tau\nabla\mathbf{g}(\mathbf{y_k}))$
7. Compute $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
8. $\mathbf{y_{k+1}} = \mathbf{x_k} + \left(\frac{t_k-1}{t_{k+1}}\right)\left(\mathbf{x_k} - \mathbf{x_{k-1}}\right)$
9. Set $k \leftarrow k + 1$
10. **end loop**

---

In the above

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

# References


[1] Shun-ichi Amari, Andrzej Cichocki, Howard Hua Yang, et al. A new learning algorithm for blind signal separation. *Advances in neural information processing systems*, pages 757–763, 1996.

[2] Anders H Andersen, Don M Gash, and Malcolm 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.

[3] Mohammad R Arbabshirani, Eswar Damaraju, Ronald Phlypo, Sergey Plis, Elena Allen, Sai Ma, Daniel Mathalon, Adrian Preda, Jatin G Vaidya, Tülay Adali, et al. Impact of autocorrelation on functional connectivity. *NeuroImage*, 2014.

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

[5] Byron Boots. Learning stable linear dynamical systems.

[6] Vince D Calhoun, Jingyu Liu, and Tülay Adalı. A review of group ica for fmri data and ica for joint inference of imaging, genetic, and erp data. *Neuroimage*, 45(1):S163–S172, 2009.

[7] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004.

[8] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.

[9] Harini Eavani, Theodore D Satterthwaite, Raquel E Gur, Ruben C Gur, and Christos Davatzikos. Unsupervised learning of functional network dynamics in resting state fmri. In *Information Processing in Medical Imaging*, pages 426–437. Springer, 2013.

[10] Zoubin Ghahramani and Geoffrey E Hinton. Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, 1996.

[11] Martin Havlicek, Karl J Friston, Jiri Jan, Milan Brazdil, and Vince D Calhoun. Dynamic modeling of neuronal responses in fmri using cubature kalman filtering. *Neuroimage*, 56(4):2109–2128, 2011.

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

[13] Bennett A Landman, Alan J Huang, Aliya Gifford, Deepti S Vikram, Issel Anne L Lim, Jonathan AD Farrell, John A Bogovic, Jun Hua, Min Chen, Samson Jarso, et al. Multi-parametric neuroimaging reproducibility: A 3-t resource study. *Neuroimage*, 54(4):2854–2866, 2011.

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

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

[16] Martin J McKeown, Tzyy-Ping Jung, Scott Makeig, Greg Brown, Sandra S Kindermann, Te-Won Lee, and Terrence J Sejnowski. Spatially independent activity patterns in functional mri data during the stroop color-naming task. *Proceedings of the National Academy of Sciences*, 95(3):803–810, 1998.

[17] Jeffrey D Meier, Tyson N Aflalo, Sabine Kastner, and Michael SA Graziano. Complex organization of human primary motor cortex: a high-resolution fmri study. *Journal of neurophysiology*, 100(4):1800–1812, 2008.

[18] Mary Beth Nebel, Suresh E Joel, John Muschelli, Anita D Barber, Brian S Caffo, James J Pekar, and Stewart H Mostofsky. Disruption of functional organization within the primary motor cortex in children with autism. *Human brain mapping*, 35(2):567–580, 2014.

[19] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[20] Lawrence Rabiner and Biing-Hwang Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.

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

[22] Sam Roweis and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural computation*, 11(2):305–345, 1999.

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

[24] Stephen M Smith, Mark Jenkinson, Mark W Woolrich, Christian F Beckmann, Timothy EJ Behrens, Heidi Johansen-Berg, Peter R Bannister, Marilena De Luca, Ivana Drobnjak, David E Flitney, et al. Advances in functional and structural mr image analysis and implementation as fsl. *Neuroimage*, 23:S208–S219, 2004.

[25] Andrey Nikolayevich Tikhonov. On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, volume 39, pages 195–198, 1943.

[26] Berwin A Turlach, William N Venables, and Stephen J Wright. Simultaneous variable selection. *Technometrics*, 47(3):349–363, 2005.

[27] Pedro A Valdés-Sosa, Jose M Sánchez-Bornot, Agustín Lage-Castellanos, Mayrim Vega-Hernández, Jorge Bosch-Bayard, Lester Melie-García, and Erick Canales-Rodríguez. Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1457):969–981, 2005.

[28] Peter Van Overschee and Bart De Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

[29] Huifang E Wang, Christian G Bénar, Pascale P Quilichini, Karl J Friston, Viktor K Jirsa, and Christophe Bernard. A systematic framework for functional connectivity measures. *Frontiers in neuroscience*, 8, 2014.

[30] Mu Zhu and Ali Ghodsi. Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics & Data Analysis*, 51(2):918–930, 2006.