An M-Estimator for Reduced-Rank High-Dimensional Linear Dynamical System Identification

Shaojie Chen[a], Kai Liu[b], Yuguang Yang[c], Yuting Xu[a], Seonjoo Lee[d], Martin Lindquist[a], Brian S. Caffo[a], and Joshua T. Vogelstein[e,f]

[a]Dept. of Biostatistics, Johns Hopkins Bloomberg School of Public Health

[b]Dept. of Neuroscience, Johns Hopkins University

[c]Dept. of Chemical and Biomolecular Engineering, Johns Hopkins University

[d]Dept. of Psychiatry and Department of Biostatistics, Columbia University

[e]Dept. of Biomedical Enginnering and Institute for Computational Medicine, Johns Hopkins University

[f]Child Mind Institute

August 21, 2015

Abstract

High-dimensional time-series data are becoming increasingly abundant across a wide variety of domains, spanning economics, neuroscience, particle physics, and cosmology to name a few. Fitting statistical models to such data, to enable parameter estimation and time-series prediction, is an important computational primitive. Existing methods, however, are unable to cope with the high-dimensional nature of these problems, due to both computational and statistical reasons. We mitigate both kinds of issues via proposing an M-estimator for Reduced-rank System IDentification (MR. SID). A combination of low-rank approximations, ℓ_1 and ℓ_2 penalties, and some numerical linear algebra, yields an estimator that is computationally efficient and numerically stable. Simulations and real data examples demonstrate the utility of this approach in a variety of problems. In particular, we demonstrate that MR. SID can estimate spatial filters, connectivity graphs, and time-courses from native resolution functional magnetic resonance imaging data. Other applications and extensions are immediately available, as our approach is a generalization of the classical Kalman Filter-Smoother Expectation-Maximization algorithm.

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

1 Introduction

High-dimensional time-series data are becoming increasingly abundant across a wide variety of domains, spanning economics, neuroscience, particle physics, and cosmology to name a few. Fitting statistical models to such data, to enable parameter estimation and time-series prediction, is an important computational primitive. Linear dynamical systems (LDS) models are amongst the most popular and powerful, because of their intuitive nature and ease of implementation. The famous Kalman Filter Smoother provides optimal estimates of the Kalman Filter Smoother is one of the most popular and powerful methods for using LDS models. Parameter estimation—often called *system identification* in this domain—is required whenever the parameters (or system is unknown). To date, there does not exist, to our knowledge, a methodology that provides parameter estimates and predictions from ultrahigh-dimensional time-series data (for example, $p > 10,000$).

The challenges associated high-dimensional time-series estimation and prediction are multifold. First, naïvly such models include dense $p \times p$ matrices, which often are too large even to store, much less operate on. Second, estimators behave poorly, due to numerical instability issues. Third, even addressing these problems the time to compute all the necessary quantities can be overly burdensome.

We address all three of these issues with our M-estimator for Reduced-rank System IDentification (MR. SID). By assuming the dimensionality of the latent state space is small (reduced rank), relative to the ambient (or observed space dimensionality), we can significantly improve computational tractability and estimation accuracy. By further penalizing the estimators, with ℓ_1 and/or ℓ_2 penalties, utilizing prior knowledge on the structure of the parameters, we gain further estimation accuracy in this high-dimensional but relatively low-sample size regime. Finally, by employing several numerical linear algebra tricks, we can bring the computational burden down significantly.

These three techniques together enable us to obtain highly accurate estimates in a variety of simulation settings. MR. SID is, in fact, a generalization of the now classic Baum-Welch expectation maximization algorithm, commonly used for system identification in much lower dimensional linear dynamical systems [15]. We numerically show that the hyper-parameters can be selected minimizing prediction error on held-out data. Finally, we use MR. SID to estimate functional connectomes from motor cortex. MR. SID enables us to estimate the regions, rather than imposing some prior parcellation on the data, as well as estimate sparse connectivity between regions. MR. SID reliably estimates these connectomes, as well as predicts the held-out time-series data. To our knowledge, this is the first time anybody has been able to estimate partitions and functional connectomes directly from the high-dimensional data, with a single unified approach. To enable extensions, generalizations, and additional applications, the functions and code for generating each of the figures is freely available from `https://github.com/shachen/PLDS/`.

2 The Model

In statistical data analysis, one often encounter some observed signals and also has some unobserved or latent variables, which are denoted as $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T)$ and $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ respectively. By the Bayesian rule, the joint probability of $\boldsymbol{Y}$ and $\boldsymbol{X}$ is $P(\boldsymbol{X}, \boldsymbol{Y}) = P(\boldsymbol{Y}|\boldsymbol{X})P(\boldsymbol{X})$. The joint conditional distribution $P(\boldsymbol{Y}|\boldsymbol{X})$ and joint prior can both be represented as a product of marginals:

$$\begin{aligned} P(\boldsymbol{Y}|\boldsymbol{X}) &= \prod_{t=1}^{T} P(\boldsymbol{y}_t|\boldsymbol{y}_{0:t-1}, x_{0:t}), \\ P(\boldsymbol{X}) &= P(x_0) \prod_{t=1}^{T} P(x_t|x_{0:t-1}). \end{aligned} \tag{1}$$

The generic time-invariant state-space model makes the following simplifying assumptions

$$\begin{aligned} P(\boldsymbol{y}_t|\boldsymbol{y}_{0:t-1}, x_{0:t}) &\approx P(\boldsymbol{y}_t|\boldsymbol{x}_t), \\ P(x_t|x_{0:t-1}) &\approx P(x_t|x_{t-1}). \end{aligned} \tag{2}$$

Finally, a lineary dynamical system assumes that both of the above terms are linear Gaussian functions, which, when written as an iterative random process, yields the standard matrix update rules:

$$\begin{aligned} &\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + \boldsymbol{Y}'_t, \quad \boldsymbol{Y}'_t \sim N(\boldsymbol{0}, Q), \quad \boldsymbol{x}_0 \sim N(\pi_0, V_0) \\ &\boldsymbol{y}_t = C\boldsymbol{x}_t + \mathbf{v}_t, \qquad \mathbf{v}_t \sim N(\boldsymbol{0}, R), \end{aligned} \tag{3}$$

where A is the $d \times d$ state transition matrix and C is the $p \times d$ generative matrix. $\boldsymbol{x}_t$ is a $d \times 1$ vector and $\boldsymbol{y}_t$ is a $p \times 1$ vector. The output noise covariance R is $p \times p$, while the state noise covariance Q is $d \times d$. Initial state mean π_0 is $d \times 1$ and covariance V_0 is $d \times d$.

The model can be thought as the continuous version of the hidden Markov model (HMM), where the columns of C stands for the hidden states. The difference is that in this model, what one observe at each time point is not a single state, but a linear combination of multiple states. $\boldsymbol{x}_t$ is the weights in the linear combination. A matrix is the analogy of the state transition matrix, which describe how the weights $\boldsymbol{x}_t$ evolve over time. Another difference is that LDS contains two white noise terms, which are captured by the Q and R matrices.

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: $V_0 = \boldsymbol{0}$

Note that the first two constraints follow directly from Roweis and Ghahramani (1999) [16].

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 Λ is a diagonal matrix of eigenvalues. Thus, for any model where Q is not the identity matrix, one can generate an equivalent model using a new state vector $\boldsymbol{x}^{\intercal} = \Lambda^{-1/2}E^T\boldsymbol{x}$ with $A^{\intercal} = (\Lambda^{-1/2}E^T)A(E\Lambda^{1/2})$ and $C^{\intercal} = C(E\Lambda^{1/2})$ such that the new covariance of $\boldsymbol{x}^{\intercal}$ is the identity matrix, i.e., $Q^{\intercal} = \mathbf{I}$. 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 $\boldsymbol{x}_0 = \pi_0$ 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) [16].

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 neuroimaging data analysis, each column of C can be a signal in the brain. Therefore having the signal spatially smooth can help extract meaningful information from the noisy neuroimaging data. In this context, an ℓ_2 penalty term on C is used to enforce smoothness.

With all those constraints, the model becomes:

$$\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + \boldsymbol{Y}'_t, \quad \boldsymbol{Y}'_t \sim N(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{x}_0 = \pi_0 \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + \mathbf{v}_t, \qquad \mathbf{v}_t \sim N(\mathbf{0}, R). \end{aligned} \tag{4}$$

where A is a sparse matrix and C has smooth columns. Let $\theta = \{A, C, R, \pi_0\}$ represents all unknown parameters and $P(\boldsymbol{X}, \boldsymbol{Y})$ be the likelihood for a generic LDS model, then combing model (4) and the constraints on A and C lead us to an optimization problem

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left\{ -\log P_\theta(\boldsymbol{X}, \boldsymbol{Y}) + \lambda_1 \|A\|_1 + \lambda_2 \|C\|_2^2 \right\} \tag{5}$$

where λ_1 and λ_2 are tuning parameters and $\| \boldsymbol{.} \|_p$ represents the p-norm of a vector. Equivalently, this optimization problem can be written as

$$\begin{aligned} &\text{minimize} && \{-\log P_\theta(\boldsymbol{X}, \boldsymbol{Y})\} \\ &\text{subject to:} && \alpha\|A\|_1 + (1-\alpha)\|C\|_2^2 \leq t \text{ for some } t; \\ & && A \in \mathcal{A}_{d\times d},\ C \in \mathcal{C}_{p\times d}, R \in \mathcal{R}_{p\times p}, \pi_0 \in \pi_{d\times 1}. \end{aligned} \tag{6}$$

where $\alpha = \frac{\lambda_1}{\lambda_1+\lambda_2}$. $\mathcal{A}_{d\times d}$ and $\mathcal{C}_{p\times d}$ are matrix spaces of $d \times d$ and $p \times d$ dimensional respectively. $\mathcal{R}_{p\times p}$ is the $p \times p$ diagonal matrix space and $\pi_{d\times 1}$ is the d dimensional vector space.

3 Parameter Estimation

The motivating application requires solving optimization problem (5): given only an observed sequence (or multiple sequences in some applications) of outputs $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T)$, 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) [17]. The detailed EM steps for a generic LDS can be found in Zoubin and Geoffrey (1996) [9]. An alternative approach is to use subspace identification methods such as N4SID and PCA-ID to compute an asymptotically unbiased solution in closed form [22, 7]. 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 (5) due to the introduced penalty terms. We therefore developed a novel algorithm called M-estimation for Reduced rank System IDentification (MR. SID), as detailed in the following.

By the chain rule, the likelihood in model (4) is

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

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

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

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

$$\begin{aligned} \log P(\boldsymbol{X},\boldsymbol{Y}) = &-\sum_{t=1}^{T}\big(\frac{1}{2}[\boldsymbol{y}_t - C\boldsymbol{x}_t]^{\intercal} R^{-1}[\boldsymbol{y}_t - C\boldsymbol{x}_t]\big) - \frac{T}{2}\log|R| \\ &-\sum_{t=1}^{T}\big(\frac{1}{2}[\boldsymbol{x}_t - A\boldsymbol{x}_{t-1}]^{\intercal}[\boldsymbol{x}_t - A\boldsymbol{x}_{t-1}]\big) - \frac{T}{2}\log|\mathbf{I}| + \log(\mathbb{1}_{\pi_0}(\boldsymbol{x}_0)). \end{aligned} \tag{7}$$

Replace $\log P(\boldsymbol{X},\boldsymbol{Y})$ with equation (7), optimization problem (5) is

$$\begin{aligned} \hat{\theta} = \underset{\theta}{\operatorname{argmin}}\Big\{ &\sum_{t=1}^{T}\big(\frac{1}{2}[\boldsymbol{y}_t - C\boldsymbol{x}_t]^{\intercal} R^{-1}[\boldsymbol{y}_t - C\boldsymbol{x}_t]\big) - \frac{T}{2}\log|R| \\ &+\sum_{t=1}^{T}\big(\frac{1}{2}[\boldsymbol{x}_t - A\boldsymbol{x}_{t-1}]^{\intercal}[\boldsymbol{x}_t - A\boldsymbol{x}_{t-1}]\big) - \frac{T}{2}\log|\mathbf{I}| - \log(\mathbb{1}_{\pi_0}(\boldsymbol{x}_0)) \\ &+\lambda_1\|A\|_1 + \lambda_2\|C\|_2^2\Big\}. \end{aligned} \tag{8}$$

Denote the target function in the curly braces as $\boldsymbol{\Phi}(\theta,\boldsymbol{Y},\boldsymbol{X})$, then $\boldsymbol{\Phi}$ can be optimized with an Expectation-Maximization (EM) algorithm.

3.1 E Step

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

$$\Gamma = E[\log P(\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{Y})].$$

This quantity depends on three expectations: $E[\boldsymbol{x}_t|\boldsymbol{Y}]$, $E[\boldsymbol{x}_t\boldsymbol{x}_t^{\intercal}|\boldsymbol{Y}]$ and $E[\boldsymbol{x}_t\boldsymbol{x}_{t-1}^{\intercal}|\boldsymbol{Y}]$. We denote their finite sample estimators by:

$$\hat{\boldsymbol{x}}_t \equiv E[\mathbf{x_t}|\boldsymbol{Y}],\ \hat{P}_t \equiv E[\boldsymbol{x}_t\boldsymbol{x}_t^{\intercal}|\boldsymbol{Y}],\ \hat{P}_{t,t-1} \equiv E[\boldsymbol{x}_t\boldsymbol{x}_{t-1}^{\intercal}|\boldsymbol{Y}]. \tag{9}$$

Expectations (9) are estimated with a Kalman filter/smoother, which is detailed in Appendix 1. Notice that all expectations are taken with respect to the current estimations of parameters.

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 $\mathbf{\Phi}(\theta, \boldsymbol{Y}, \boldsymbol{x})$, 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,

$$
\begin{aligned}
\frac{\partial \mathbf{\Phi}}{\partial R^{-1}} &= \frac{T}{2} R - \sum_{t=1}^{T} \Big(\frac{1}{2} \boldsymbol{y}_t \boldsymbol{y}_t^{\intercal} - C \hat{\boldsymbol{x}}_t \boldsymbol{y}_t^{\intercal} + \frac{1}{2} C \hat{P}_t C^{\intercal} \Big) = 0 \\
\implies R &= \frac{1}{T} \sum_{t=1}^{T} (\boldsymbol{y}_t \boldsymbol{y}_t^{\intercal} - C^{\text{new}} \hat{\boldsymbol{x}}_t \boldsymbol{y}_t^{\intercal}) \\
\implies R^{\text{new}} &= \operatorname{diag} \Big\{ \frac{1}{T} \sum_{t=1}^{T} (\boldsymbol{y}_t \boldsymbol{y}_t^{\intercal} - C \hat{\boldsymbol{x}}_t \boldsymbol{y}_t^{\intercal}) \Big\}
\end{aligned}
\tag{10}
$$

In the bottom line, diag denotes to only extract the diagonal terms of the matrix R, as we constrain R to be diagonal in Constraint 4.

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

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

Estimation for transition matrix C also has closed form solution, and the solution can be derived by rearranging the terms properly. Terms relevant to C in equation (8) are

$$f_{\lambda_2}(C; \boldsymbol{X}, \boldsymbol{Y}) = \sum_{t=1}^{T} \left(\frac{1}{2} [\boldsymbol{y}_t - C \boldsymbol{x}_t]^{\intercal} R^{-1} [\boldsymbol{y}_t - C \boldsymbol{x}_t] \right) + \lambda_2 \|C\|_2. \tag{11}$$

In $f_{\lambda_2}(C; \boldsymbol{X}, \boldsymbol{Y})$, C is a matrix, we vectorized it to ease optimization and notation. Here we follow the methods of Turlach et al. [20]. Without loss of generality, assume R is the identity matrix in equation (11); otherwise, one can always write equation (11) as

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

Let

$$\boldsymbol{Y}' = (y_{11}, \ldots, y_{T1}, y_{12}, \ldots, y_{T2}, \ldots, y_{1p}, \ldots, y_{Tp})^{\intercal}$$

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

$$\boldsymbol{X}' = \begin{pmatrix} \boldsymbol{X}^{\intercal} & & \\ & \ddots & \\ & & \boldsymbol{X}^{\intercal} \end{pmatrix}_{pT \times pd}.$$

Finally, vectorize C^{old} as

$$\mathbf{c}^{\text{old}} = (C_{11}^{\text{old}}, \ldots, C_{1d}^{\text{old}}, C_{21}^{\text{old}}, \ldots, C_{2d}^{\text{old}}, C_{p1}^{\text{old}}, \ldots, C_{pd}^{\text{old}})^{\intercal} \tag{12}$$

where C_{ij} is the element at row i and column j of C. With these new notations, the equation (11) is equivalent to

$$f_{\lambda_2}(C; \boldsymbol{X}, \boldsymbol{Y}) = \|\boldsymbol{Y}' - \mathbf{X}'\mathbf{c}\|_2^2 + \lambda_2 \|\mathbf{c}\|_2^2. \tag{13}$$

With the Tikhonov regularization [19], equation (13) has closed form solution

$$\begin{aligned} \mathbf{c}^{\text{new}} &= (\boldsymbol{X}'^{\intercal}\boldsymbol{X}' + \lambda_2 \mathbf{I})^{-1} \boldsymbol{X}'^{\intercal}\boldsymbol{Y}' \\ C^{\text{new}} &= \text{Rearrange } \mathbf{c}^{\text{new}} \text{ by equation } (12) \end{aligned} \tag{14}$$

Now let's look at parameter A. Terms involving A in equation (8) are,

$$f_{\lambda_1}(A; \boldsymbol{X}, \boldsymbol{Y}) = \sum_{t=1}^{T} \big(\frac{1}{2} [\boldsymbol{x}_t - A\boldsymbol{x}_{t-1}]^{\intercal} [\boldsymbol{x}_t - A\boldsymbol{x}_{t-1}] \big) + \lambda_1 \|A\|_1. \tag{15}$$

Similar to what we have done to C, equation (15) is equivalent to

$$f_{\lambda_1}(A; \boldsymbol{X}, \boldsymbol{Y}) = \|\mathbf{z} - \mathbf{Z}\mathbf{a}\|_2^2 + \lambda_1 \|\mathbf{a}\|_1. \tag{16}$$

where $\mathbf{z}$ is a $Td \times 1$ vector from rearranging $\boldsymbol{X}$ and $\mathbf{Z}$ is a block diagonal matrix with diagonal component $Z^{\intercal} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{T-1})^{\intercal}$. Unfortunately, equation (16) does not have closed form solution due to the

$$\ell_1$$

term.

Though not having a closed form solution, $f_{\lambda_1}(A; \boldsymbol{X}, \boldsymbol{Y})$ 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) [6]. 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 $\nabla \mathbf{g}(\mathbf{z}) = \mathbf{Z}^{\mathsf{T}}(\mathbf{Z}\mathbf{a} - \mathbf{z})$, where $\mathbf{g}(\mathbf{z}) = \|\mathbf{Z}^{\mathsf{T}}\mathbf{a} - \mathbf{z}\|_2^2$. Denote $\|Z\|$ as the induced norm of matrix Z, then L is

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

With FISTA and L, matrix A can be updated:

$$A^{\text{new}} = \text{FISTA}(\|\mathbf{Z}^{\mathsf{T}}\mathbf{a}^{\text{old}} - \mathbf{z}\|_2^2, \quad \lambda_1) \tag{17}$$

3.3 Initialization

R matrix is initialized as the identify matrix, while π_0 as $\mathbf{0}$ vector. For A and C, denote $\boldsymbol{Y} = [\mathbf{y_1}, \cdots, \mathbf{y_T}]$, a $p \times T$ matrix, then the singular value decomposition (SVD) of $\boldsymbol{Y}$ is

$$\boldsymbol{Y} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}} \approx \mathbf{U}_{p\times d}\mathbf{D}_{d\times d}\mathbf{V}^{\mathsf{T}}_{d\times T} = \mathbf{U}_{p\times d}\boldsymbol{X}_{d\times T} \tag{18}$$

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 $\mathbf{V}^{\mathsf{T}}_{d\times T}$.

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

3.4 The Complete EM

The complete EM algorithm for MR. SID is addressed as follows.

Algorithm EM Algorithm for MR. SID

M Step

1. $R^{\text{new}} = \text{diag}\left\{ \frac{1}{T} \sum_{t=1}^{T} (\boldsymbol{y}_t \boldsymbol{y}_t^{\top} - C^{\text{old}} \hat{\boldsymbol{x}}_t \boldsymbol{y}_t^{\top}) \right\}$, as in equation (10)
2. $\pi_0^{\text{new}} = \hat{\boldsymbol{x}}_0$
3. Update C^{new}, as in equation (14)
4. Update A^{new} with FISTA, as in equation (17)
5. Stop when difference between estimations from this step and previous step is less than tolerance or maximum number of iterations reached.

E Step

0. Initialize $\theta = \{A, C, R, \pi_0\}$ as in Section 3.3, if first loop
1. Update the expectations in (9) with the Kalman filter smoother in Appendix 1

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

3.5 Improving Computational Efficiency

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 consumes over 100 gigabyte of 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,

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

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

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

Also note that quantities like R^{-1} and $C^{\top} R^{-1} C$ can be pre-computed and reused throughout the E step. With

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

3.6 The Data

The Kirby 21 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 [11]. 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.

The Human Connectome Project is a systematic effort to map macroscopic human brain circuits and their relationship to behavior in a large population of healthy adults [21, 13, 8]. MR scanning includes four imaging modalities, acquired at high resolutions: structural MRI, resting-state fMRI (rfMRI), task fMRI (tfMRI), and diffusion MRI (dMRI). All 1200 subjects are scanned using all four of these modalities on a customized 3 T scanner. All scans consist of 1200 time points.

4 Results

4.1 Parameter Estimation

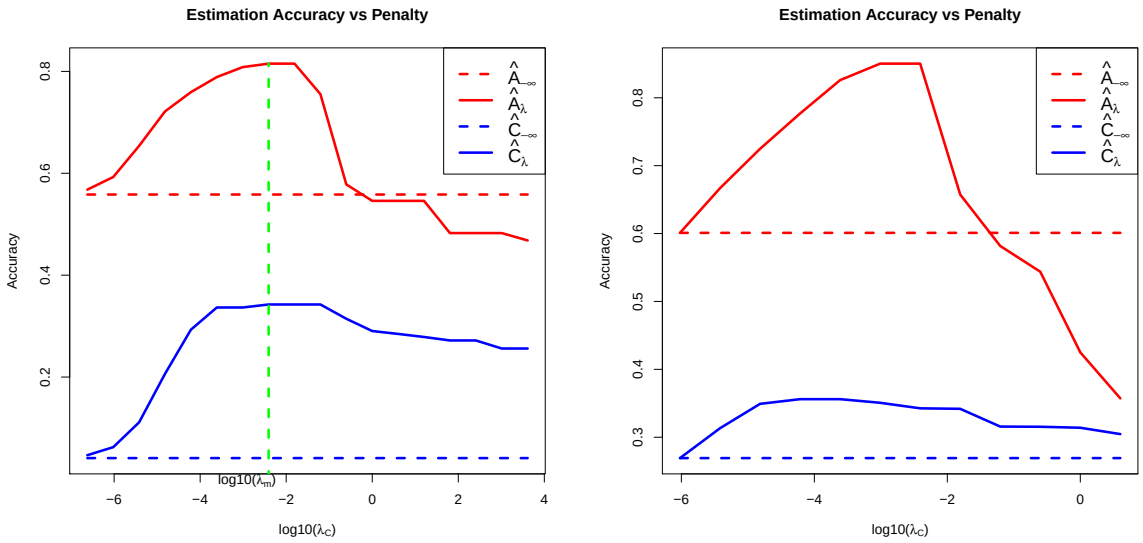
Two simulations of different dimensions are performed to demonstrate the model and its parameter estimations.

In the low dimensional setting, $p = 300$, $d = 10$ and $T = 100$. A is first generated from a random matrix, then elements with small absolute values are then truncated such that 20 percent of elements are zeros. After that a multiple of the identity matrix is added to A. A is then scaled to make sure its eigenvalues fall 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 each column 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 $\mathbf{x_0}$. Pseudocode for data generation is included in Appendix 4.

In the high-dimensional setting, $p = 10000$, $d = 30$ and $T = 100$. The parameter are generated in a the same manner.

To evaluate the accuracy of estimations, we elect to define the distance between two matrices A and B is defined as follows

$$d(A, B) = \underset{P \in P(n)}{\operatorname{argmin}} \left\{ \log\left[\frac{n}{\mathsf{Trace}(P \times C_{A,B})} \right] \right\}. \tag{19}$$



(a) Low dimensional setting

(b) High dimensional setting

Figure 1: x axis is tuning parameter λ_C under log scale and y axis is the distance between truth and estimations; λ_A is increasing proportionally with λ_C. One can see that in both the low dimensional and hight dimensional setting, estimation accuracies for A and C first increase then decrease as penalty increases.

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 [10].

Both the generic LDS and the penalized LDS are applied to the simulation data. As the true parameters are sparse, we expect that the penalized algorithms would yield better estimations with appropriately chosen penalty parameters. When the penalties are approaching 0, the penalized algorithm should converge to the generic model. In addition, when the penalties are too big or too small compared to the optimal values, estimations might be less accurate.

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

Estimation accuracies are plotted against penalty size λ_C in Figure 1. Results from LDS and MR. SID are overlayed in one plot for comparison. As the figure shows, MR. SID 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. MR. SID 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.

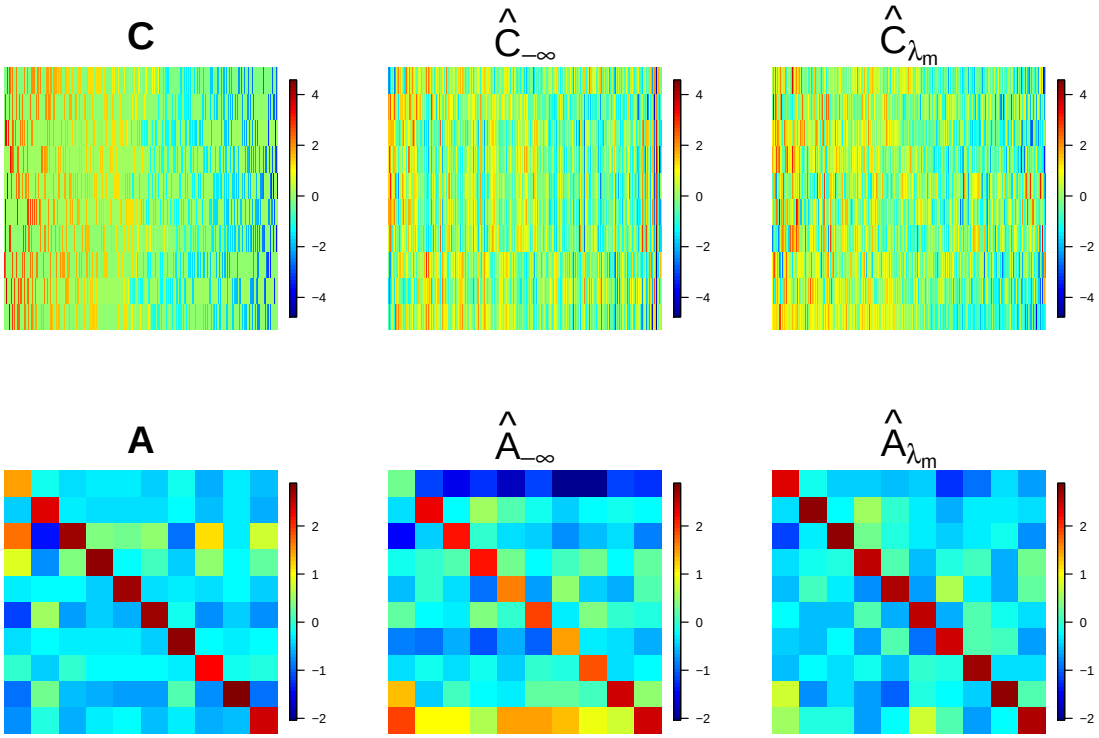


Figure 2: Row 1: A truth; non-penalized estimation of A; optimally penalized estimation of A. Row 2: C truth; non-penalized estimation of C; optimally penalized estimation of C.

In addition to the improved estimation accuracy, the proposed algorithm is also computational efficiency and highly scalable. As a demonstration, 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: MR. SID Running Time

p	100	1000	10000	100000	100000
d	10	30	50	100	500
T	100	300	500	1000	1000
Time (min)	0.07	0.30	5.15	111.38	1127.18

4.2 Making Predictions

Another perspective when considering MR. SID model is its ability to make predictions. When the parameters θ 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 simu-

lation. 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.

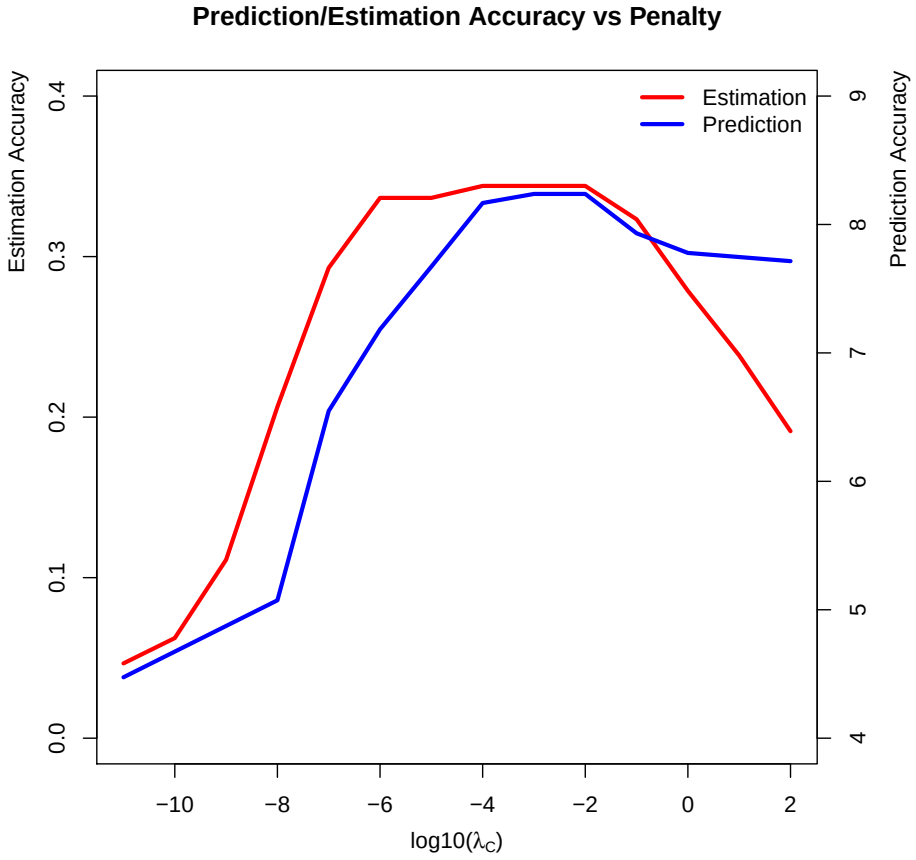


Figure 3: Estimation and prediction accuracies. The x-axis is the penalty size under log scale, while y-axis is the estimation and prediction accuracies. One can see that the penalty that yields the most accurate estimation also gives the best predictions.

Observations and findings from these plots include:

- The prediction accuracy first improves then drops when the penalties increase
- The prediction accuracy peaks when the penalty coefficient λ_A and λ_C are around 10^{-3}. This make sense as the same λ pair also gives the best estimation for coefficients A and C, as in Figure 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 $\boldsymbol{y}_t$ is a time step including the entire brain volume. Each column of the C matrix is interpreted as a time-invariant brain "point spread function". At each time point, the observed brain image, $\boldsymbol{y}_t$, is a linear mixture of latent co-assemblies of neural activity $\boldsymbol{x}_t$. Matrix A describes how $\boldsymbol{x}_t$ evolves over time. A can also be viewed as a directed graph if each neural assembly is treated as a vertex. Each neural assembly is spatially smooth and connectivities across them are empirically sparse. This naturally fits into the sparsity and smoothness assumptions in MR. SID.

MR. SID is applied to analyze the motor cortex of human brains from the KIRBY 21 Data. 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 [18]. FSL is used for spatial smoothing with Gaussian kernel. Then MR. SID 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: $\lambda_A = \lambda_C = 10^{-5}$.[1]
- 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 [12].

A more flexible technique to choose the number of latent states involves the profile likelihood method proposed by Zhu et al. [23]. 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 eigenvalues 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 likelihood 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 11, 6, 14 and 15 respectively. Their average,

[1] Different combinations of λ_A and λ_C, where $\lambda_A = \lambda_C$ is applied to the data. The values range from 10^{-10} to 10^4. Then the estimations are used to make predictions. The combination that gives the best predictions is used here. One can also use a grid of combinations, but it is time consuming.

$d = 11$, is used.

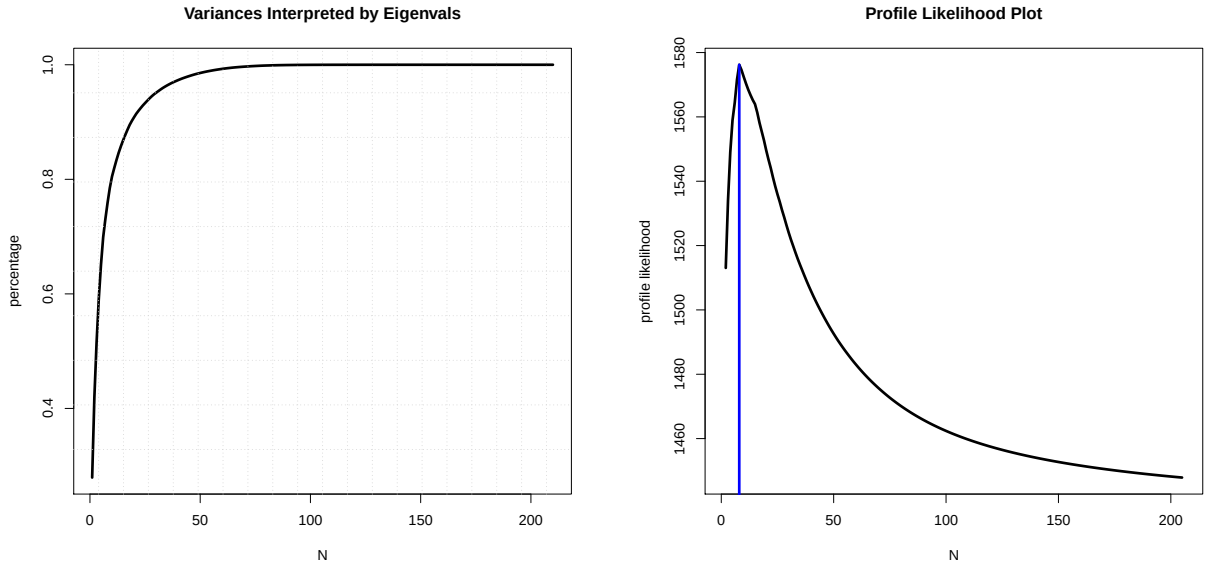


Figure 4: Eigen-values and corresponding profile likelihood plot from the first scan of subject one. For this data, the profile likelihood picks $d = 11$ as the number of latent states.

Denote the A matrix estimation for the second scan of subject one as A_{12}. Similar notations apply to the other scans. Then the canonical correlations among the four matrices are summarized in Table 2. The distance measure in Equation (19) is used. Another permutation invariant measure of distance between two square matrices, the Amari error [2], is also provided in the table [2]. Notice a smaller $d(A, B)$ or Amari error means more similarity. Among the six pairs from A_{11}, A_{12}, A_{21} and A_{22}, it is expected that the pairs (A_{11}, A_{12}) and (A_{21}, A_{22}) give the smallest distances, as each pair comes from two scans of the same subject. This idea is validated by Table 2. A direct application of this result is to correctly cluster the four scans into two group, each group corresponding to a subject. This implies that the A matrices contains subject-specific information. The A matrices as connectivity graphs are plotted in Figure 5.

[2]The Amari error between A and $\hat{A}$: $E(A, \hat{A}) = \sum_{i=1}^{n}(\sum_{j=1}^{n}\frac{|p_{ij}|}{\max_k |p_{ik}|} - 1) + \sum_{j=1}^{n}(\sum_{i=1}^{n}\frac{|p_{ij}|}{\max_k |p_{kj}|} - 1)$, where $P = (p_{ij}) = A^{-1}\hat{A}$.

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

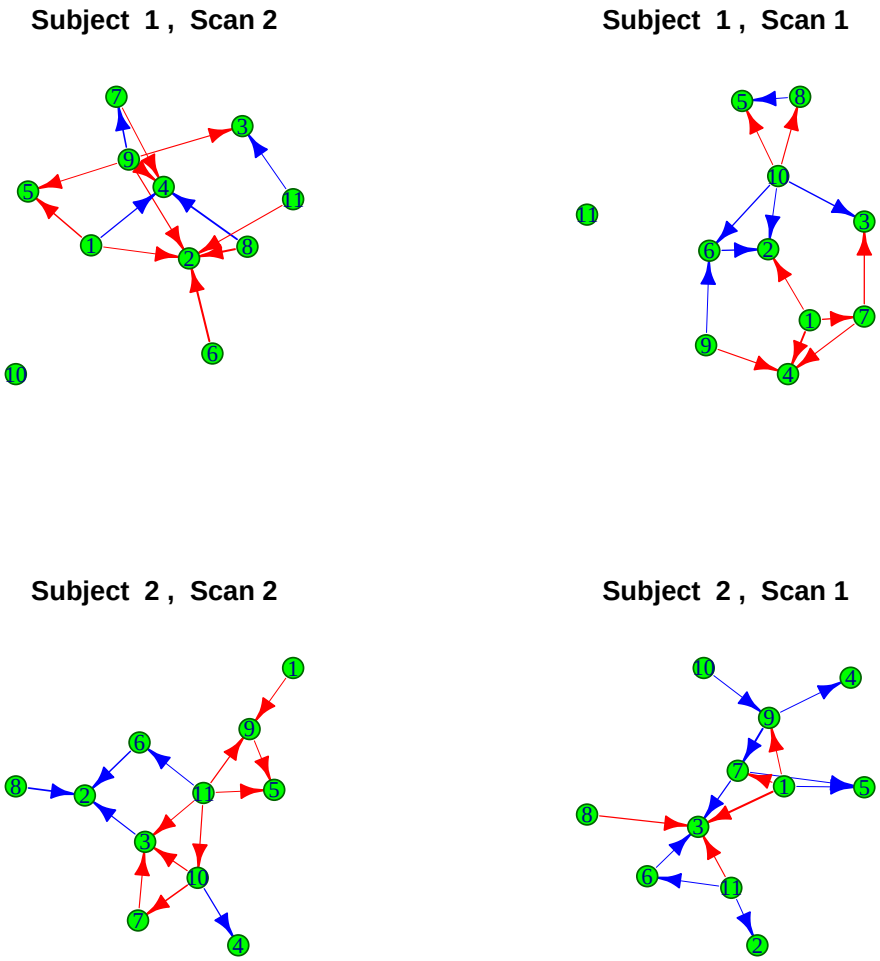


Figure 5: Connectivity Graph: The wider edge means stronger connectivity; the red edge means negative connectivity and blue edge means positive connectivity.

The similarities among A matrices are also shown in Figure (6) as a heatmap.

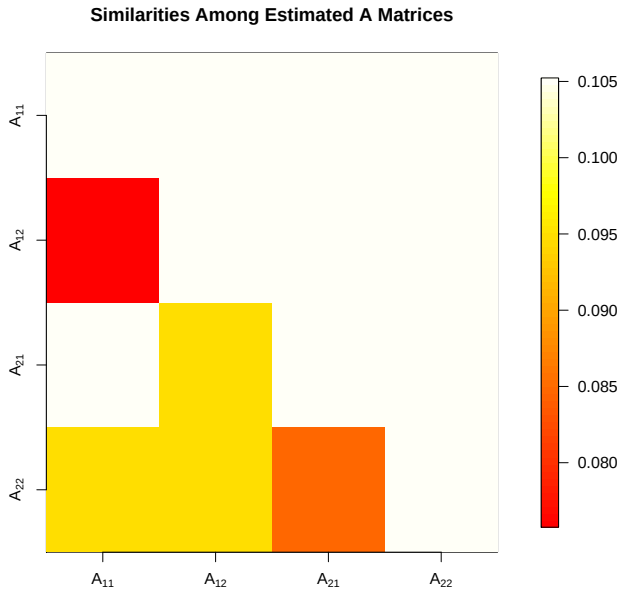


Figure 6: Similarities among the four estimated A matrices. The distance $d(\cdot, \cdot)$ is used in this figure. As one can see, the two red/orange off-diagonal pixels has the minimum distances, which correspond to the pairs of (A_{11}, A_{12}) and (A_{21}, A_{22}) respectively. With this similarity map, one can tell which two scans are from the same subject.

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). It is helpful to compare those regions to other existing parcellations of the mortor cortex. As an example, the blue region in Figure 7 accurately matches the dorselmedical (DM) parcel of the five-region parcellation proposed by Nebel MB et al. [14].

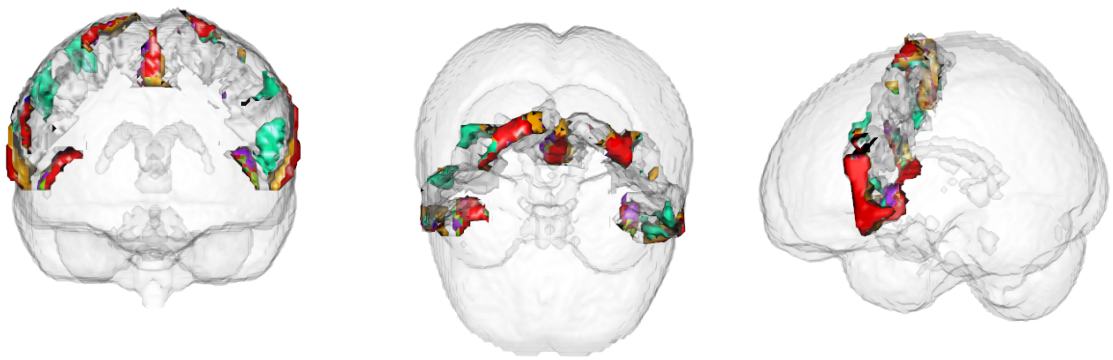


Figure 7: 3D rendering of columns of matrix C: estimation from the first scan of subject one shown in this plot.

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

the SVD method in Section 3.3 and MR. SID algorithm are used for estimations. Then k-step ahead predictions are made with equations (4) and estimations from both methods. Pseudocode for k-step ahead predictions is given in Appendix 3. The prediction accuracies are shown in Figure 8 (left panel). The first observation is that, MR. SID 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 MR. SID algorithm, this shows that MR. SID algorithm improves estimations from the SVD method in terms of short-term predictions. Another observation is, MR. SID algorithm's performance get worse when one predicts into the "long" future (> 150 steps). This is reasonable because the prediction errors from each step will accumulate and yields deteriorating predictions as the number of steps increase.

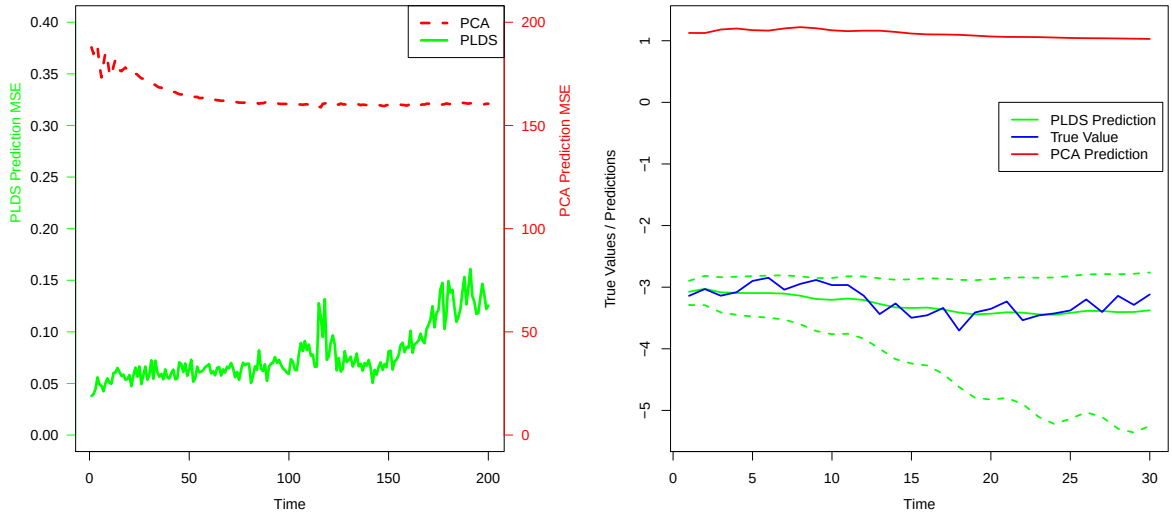


Figure 8: Prediction accuracies comparison on HCP data. *Left:* The mean squared error (MSE) is used as the accuracy measure. *Right:* Sample time series plot. The dotted green curve stands for the 60% confidence band given by MR. SID model. The true time series is averaged signals from a subsample of voxels. The predictions are also averaged over the same subsample. The confidence band is estimated based on the covariance matrix of these voxels. A subsample of 20 voxels are picked in this experiment to avoid big covariance matrices calculation. All values are log-scaled for plotting purpose.

A sample plot of the true time series and predicted values are shown in Figure 8 (right panel). We see that MR. SID is giving more accurate predictions and the true signal lies in the confidence band giving by MR. SID 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.

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 [1]. 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 $\boldsymbol{x}_t^\tau$ = E($\boldsymbol{x}_t|\boldsymbol{Y}_1^\tau$),$\mathbf{V}_t^\tau$ = Var($\boldsymbol{x}_t|\boldsymbol{Y}_1^\tau$), $\hat{\boldsymbol{x}}_t \equiv \boldsymbol{x}_t^T$ and $\hat{P}_t \equiv V_t^T + \boldsymbol{x}_t^T \boldsymbol{x}_t^{T^\mathsf{T}}$

1. Forward Recursions:

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

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

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

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

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

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

2. Backward Recursions:

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

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

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

$\hat{P}_{t,t-1} \equiv V_{t,t-1}^T + \boldsymbol{x}_t^T \boldsymbol{x}_t^{T^\mathsf{T}}$

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

Appendix 2

In general, FISTA optimize a target function

$$\min_{x\in\mathcal{X}} \quad \mathbf{F}(\mathbf{x};\lambda) = \mathbf{g}(\mathbf{x}) + \lambda\|\mathbf{x}\|_{\mathbf{1}} \tag{20}$$

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. $\mathbf{y_{k+1}} = \mathbf{x_k} + \left(\frac{t_k - 1}{t_{k+1}}\right)\Big(\mathbf{x_k} - \mathbf{x_{k-1}}\Big)$
9. Set $k \leftarrow k + 1$
10. **end loop**

In the above

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

Appendix 3

Algorithm k-step predictions with PCA and MR. SID

1. Denote the estimated parameters with PCA and MR. SID as $A_{pca}, C_{pca}, A_{plds}$, and C_{plds} respectively.
2. PCA Estimated latent states at $t = 1000$: $x_{1000,pca} =$ the 1000th column of $\boldsymbol{X}_{d\times T}$ from Section 3.3
3. MR. SID Estimated latent states at $t = 1000$: $x_{1000,pls}$ is from E step of the EM algorithm in Section 3.4
4. **for i = 1 to k**
5. $\quad x_{1000+k,pca} = A_{pca}\, x_{999+k,pca}$
6. $\quad y_{1000+k,pca} = C_{pca}\, x_{1000+k,pca}$
7. $\quad x_{1000+k,plds} = A_{plds}\, x_{999+k,plds}$
8. $\quad y_{1000+k,plds} = C_{plds}\, x_{1000+k,plds}$
9. **end**

Appendix 4

Algorithm Simulation Data Generation

1. Denote the dimensions as p, d and T respectively
2. Generate a $p \times d$ matrix C_0 from a standard Gaussian distribution
3. Sort each column of C_0 in ascending order to get matrix C
4. Generate a $d \times d$ matrix A_0 from a standard Gaussian distribution
5. Add a multiple of the identity matrix to A_0
6. Replace entries in A_0 with small absolute values with 0
7. Scale A_0 to make sure its eigen values are between -1 and 1; use A_0 as the A matrix
8. Let R be a diagonal matrix with positive diagonal entries and Q be the identity matrix
9. Generate simulation data with A, C, Q and R
10. **end**

References

[1] Genevera I Allen, Logan Grosenick, and Jonathan Taylor. A generalized least-square matrix decomposition. *Journal of the American Statistical Association*, 109(505):145–159, 2014.

[2] 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.

[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] 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.

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

[8] David A Feinberg, Steen Moeller, Stephen M Smith, Edward Auerbach, Sudhir Ramanna, Matthias Gunther, Matt F Glasser, Karla L Miller, Kamil Ugurbil, and Essa Yacoub. Multiplexed echo planar imaging for sub-second whole brain fmri and fast diffusion imaging. *PloS one*, 5(12):e15710, 2010.

[9] 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.

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

[11] 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.

[12] 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.

[13] Steen Moeller, Essa Yacoub, Cheryl A Olman, Edward Auerbach, John Strupp, Noam Harel, and Kâmil Uğurbil. Multiband multislice ge-epi at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fmri. *Magnetic Resonance in Medicine*, 63(5):1144–1153, 2010.

[14] 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.

[15] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

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

[17] 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.

[18] 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.

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

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

[21] David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, WU-Minn HCP Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

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

[23] 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.