

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

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

<sup>a</sup>Dept. of Biostatistics, Johns Hopkins Bloomberg School of Public Health

<sup>b</sup>Dept. of Neuroscience, Johns Hopkins University

<sup>c</sup>Dept. of Chemical and Biomolecular Engineering, Johns Hopkins University

<sup>d</sup>Dept. of Psychiatry and Department of Biostatistics, Columbia University

<sup>e</sup>Child Mind Institute

<sup>f</sup>Dept. of Biomedical Engineering and Institute for Computational Medicine, Johns Hopkins  
University

## Abstract

High-dimensional time-series data are becoming increasingly abundant across a wide variety of domains, spanning economics, neuroscience, particle physics, and cosmology. Fitting statistical models to such data, to enable parameter estimation and time-series pre-

diction, 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 ( $\text{MR. SID}$ ). A combination of low-rank approximations,  $\ell_1$  and  $\ell_2$  penalties, and some numerical linear algebra tricks, 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  $\text{MR. SID}$  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, imaging processing, fMRI

## 1 Introduction

High-dimensional time-series data are becoming increasingly abundant across a wide variety of domains, spanning economics (Johansen, 1988), neuroscience (Friston et al., 2003), and cosmology (Xie et al., 2013). 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 (Kalman, 1963). The famous Kalman Filter Smoother is one of the most popular and powerful tool for time-series prediction with an LDS, given known parameters (Kalman, 1960). In practice, however, for many

LDSs, the parameters are unknown, and must be estimated, a process often called *system identification* in this domain (Ljung, 1998). 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, naïvely such models include dense  $p \times p$  matrices, which often are too large even to store, and much too large to invert in memory. For large sparse matrices, recently, several efforts to invert them using a series of computational tricks are promising, though still extremely computationally expensive (Hsieh et al., 2013; Banerjee et al., 2013). Second, estimators behave poorly, due to numerical instability issues. Reduced rank LDS models reduce the number of latent dimensions, and therefore partially address this problem (CHEN et al., 1989). However, without further constraints, the dimensionality of the latent states must be so small as to significantly decrease the predictive capacity of the resulting model. Third, even after addressing these problems, the time to compute all the necessary quantities can be overly burdensome. Distributed memory implementations, such as using Spark, might help to overcome this problem, but it would at additional costs and set-up burden, as it would require a Spark cluster (Zaharia et al., 2010).

We address all three of these issues with our M-estimator for Reduced-rank System Identification ( $\text{MR. SID}$ ). 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  $\ell_1$  and/or  $\ell_2$  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 (Rabiner, 1989). 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 on Github. (referred to as the Git Repo in following sections)

## 2 The Model

In statistical data analysis, one often encounter some observed variables, as well as some unobserved latent variables, which we denote as  $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_T)$  and  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$  respectively. By the Bayesian rule, the joint probability of  $\mathbf{Y}$  and  $\mathbf{X}$  is  $P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{Y}|\mathbf{X})P(\mathbf{X})$ . The conditional distribution  $P(\mathbf{Y}|\mathbf{X})$  and prior  $P(\mathbf{X})$  can both be represented as a product

of marginals:

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

$$P(\mathbf{X}) = P(\mathbf{x}_0) \prod_{t=1}^T P(\mathbf{x}_t|\mathbf{x}_0, \dots, \mathbf{x}_{t-1}).$$

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

$$P(\mathbf{y}_t|\mathbf{y}_0, \dots, \mathbf{y}_{t-1}, \mathbf{x}_0, \dots, \mathbf{x}_t) \approx P(\mathbf{y}_t|\mathbf{x}_t),$$

$$P(\mathbf{x}_t|\mathbf{x}_0, \dots, \mathbf{x}_{t-1}) \approx P(\mathbf{x}_t|\mathbf{x}_{t-1}).$$

A linear dynamical system (LDS) further 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:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{w}_t, \quad \mathbf{w}_t \sim N(\mathbf{0}, Q), \quad \mathbf{x}_0 \sim N(\pi_0, V_0)$$

$$\mathbf{y}_t = C\mathbf{x}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim N(\mathbf{0}, R),$$

where  $A$  is a  $d \times d$  state transition matrix and  $C$  is a  $p \times d$  generative matrix.  $\mathbf{x}_t$  is a  $d \times 1$  vector and  $\mathbf{y}_t$  is a  $p \times 1$  vector. The 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  $d \times d$ .

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 time  $t$  is not a single state, but a linear combination of multiple states.  $\mathbf{x}_t$  keeps the weights in the linear combination.  $A$  matrix is the analogy of

the state transition matrix, which describe how the weights  $x_t$  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 LDS model itself is unidentifiable. Three basic constraints are introduced to make it 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:  $V_0 = 0$

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

The logic for Constraint 1 is as follows. Since covariance matrix  $Q$  is symmetric and positive semidefinite, it can be decomposed as  $E\Lambda E^T$ , where  $E$  is a rotation matrix of eigenvectors and  $\Lambda$  is a diagonal matrix of eigenvalues. Then for any model whose  $Q$  is not the identity matrix, one can always generate an equivalent model using a new state vector  $\mathbf{z} = \Lambda^{-1/2} E^T \mathbf{x}$ , with  $A_{\mathbf{z}} = (\Lambda^{-1/2} E^T) A (E \Lambda^{1/2})$  and  $C_{\mathbf{z}} = C (E \Lambda^{1/2})$ . The covariance of new vector  $\mathbf{z}$  is the identity matrix, i.e.,  $Q_{\mathbf{z}} = \mathbf{I}$ . 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. This is reasonable, as in many applications, there is often

only one single chain of time series observed. To estimate  $V_0$  accurately, multiple series of observations are required.

The following three 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, the  $R$  matrix can be very large. One can not accurately estimate the many free parameters in  $R$  with a limited amount of observations. 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  $R$  to be diagonal. In the static model with no temporal dynamics, a diagonal  $R$  is equivalent to the generic Factor Analysis method, while multiples of the identity  $R$  matrix lead to Principal Component Analysis (PCA) (Roweis and Ghahramani, 1999).

$A$  matrix is the transition matrix of the hidden states. In many applications, it is desirable for  $A$  to be sparse. In this work, an  $\ell_1$  penalty on  $A$  is used to impose the sparsity constraints. In the applications that follow,  $A$  is a central construct of interest representing a so-called connectivity graph, and the graph is expected to be sparse.

Similarly, in 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. Having the signals spatially smooth can help extract meaningful information from the noisy neuroimaging data. In this context, an  $\ell_2$  penalty on columns of  $C$  is used to enforce smoothness.

With all those constraints, the model becomes:

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

where  $A$  is a sparse matrix and  $C$  has smooth columns. Let  $\theta = \{A, C, R, \pi_0\}$  represents all unknown parameters and  $P(\mathbf{X}, \mathbf{Y})$  be the full likelihood, then combining model (1) and the constraints on  $A$  and  $C$  leads us to an optimization problem

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

where  $\lambda_1$  and  $\lambda_2$  are tuning parameters and  $\|\cdot\|_p$  represents the  $p$ -norm of a vector. Equivalently, this problem has the following dual form

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

where  $\alpha = \frac{\lambda_1}{\lambda_1 + \lambda_2}$ .  $\mathcal{A}_{d \times d}$  and  $\mathcal{C}_{p \times d}$  are  $d \times d$  and  $p \times d$  dimensional matrix spaces 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

Parameter estimation requires solving optimization problem (2): given only one observed sequence (or multiple sequences in some applications) of outputs  $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_T)$ , find the parameters  $\theta = \{A, C, R, \pi_0\}$  that maximize the likelihood of observations.



Parameter estimation for LDS has been investigated extensively in statistics, machine learning, control theory and signal processing research. For example, in machine learning, exact and variational inference algorithms for general Bayesian networks can be applied to LDS. In control theory, the corresponding area of study is known as system identification.

Specifically, one way to search for the maximum likelihood estimation (MLE) is through iterative methods such as expectation maximization (EM) (Shumway and Stoffer, 1982). The EM algorithm for a standard LDS is detailed in Zoubin and Geoffrey (1996) (Ghahramani and Hinton, 1996). An alternative is to use subspace identification methods such as N4SID and PCA-ID, which give asymptotically unbiased closed-form solutions (Van Overschee and De Moor, 1994; Doretto et al., 2003). In practice, determining an initial solution with subspace identification and then refining it with EM is an effective approach (Boots, Boots).

However, the above approaches are not directly applicable to optimization problem (2) due to the introduced penalty terms. We therefore developed a novel algorithm called M-estimation for Reduced rank System IDentification ( $\text{MR. SID}$ ), as detailed in the following.

By the chain rule, the full likelihood is

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

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

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

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

$$\begin{aligned} \log P(\mathbf{X}, \mathbf{Y}) = & - \sum_{t=1}^T \left( \frac{1}{2} [\mathbf{y}_t - C\mathbf{x}_t]^\top R^{-1} [\mathbf{y}_t - C\mathbf{x}_t] \right) - \frac{T}{2} \log |R| \\ & - \sum_{t=1}^T \left( \frac{1}{2} [\mathbf{x}_t - A\mathbf{x}_{t-1}]^\top [\mathbf{x}_t - A\mathbf{x}_{t-1}] \right) - \frac{T}{2} \log |\mathbf{I}| + \log(\mathbb{1}_{\pi_0}(\mathbf{x}_0)). \end{aligned} \quad (3)$$

Replace  $\log P(\mathbf{X}, \mathbf{Y})$  in problem (2) with equation (3), one gets

$$\begin{aligned} \hat{\theta} = \operatorname{argmin}_{\theta} \Big\{ & \sum_{t=1}^T \left( \frac{1}{2} [\mathbf{y}_t - C\mathbf{x}_t]^\top R^{-1} [\mathbf{y}_t - C\mathbf{x}_t] \right) - \frac{T}{2} \log |R| \\ & + \sum_{t=1}^T \left( \frac{1}{2} [\mathbf{x}_t - A\mathbf{x}_{t-1}]^\top [\mathbf{x}_t - A\mathbf{x}_{t-1}] \right) - \frac{T}{2} \log |\mathbf{I}| - \log(\mathbb{1}_{\pi_0}(\mathbf{x}_0)) \\ & + \lambda_1 \|A\|_1 + \lambda_2 \|C\|_2^2 \Big\}. \end{aligned} \quad (4)$$

Denote the target function in the curly braces as  $\Phi(\theta, \mathbf{Y}, \mathbf{X})$ , then  $\Phi$  can be optimized with `MR. SID`, a generalized Expectation-Maximization (EM) algorithm.

### 3.1 E Step

The E step of EM requires computation of the expected log likelihood,  $\Gamma = E[\log P(\mathbf{X}, \mathbf{Y} | \mathbf{Y})]$ . This quantity depends on three expectations:  $E[\mathbf{x}_t | \mathbf{Y}]$ ,  $E[\mathbf{x}_t \mathbf{x}_t^\top | \mathbf{Y}]$  and  $E[\mathbf{x}_t \mathbf{x}_{t-1}^\top | \mathbf{Y}]$ . For simplicity, we denote their finite sample estimators by:

$$\hat{\mathbf{x}}_t \equiv E[\mathbf{x}_t | \mathbf{Y}], \quad \hat{P}_t \equiv E[\mathbf{x}_t \mathbf{x}_t^\top | \mathbf{Y}], \quad \hat{P}_{t,t-1} \equiv E[\mathbf{x}_t \mathbf{x}_{t-1}^\top | \mathbf{Y}]. \quad (5)$$

Expectations (5) are estimated with a Kalman filter/smoothing, which is detailed in the Appendix on the Git Repo. Notice that all expectations are taken with respect to the current

estimations of parameters.

### 3.2 M Step

Each of the parameters in  $\theta = \{A, C, R, \pi_0\}$  is estimated by taking the corresponding partial derivatives of  $\Phi(\theta, \mathbf{Y}, \mathbf{x})$ , 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  $R$  matrix has closed form,

$$\begin{aligned} \frac{\partial \Phi}{\partial R^{-1}} &= \frac{T}{2} R - \sum_{t=1}^T \left( \frac{1}{2} \mathbf{y}_t \mathbf{y}_t^\top - C \hat{\mathbf{x}}_t \mathbf{y}_t^\top + \frac{1}{2} C \hat{P}_t C^\top \right) = 0 \\ \implies R &= \frac{1}{T} \sum_{t=1}^T (\mathbf{y}_t \mathbf{y}_t^\top - C^{\text{new}} \hat{\mathbf{x}}_t \mathbf{y}_t^\top) \\ \implies R^{\text{new}} &= \text{diag} \left\{ \frac{1}{T} \sum_{t=1}^T (\mathbf{y}_t \mathbf{y}_t^\top - C \hat{\mathbf{x}}_t \mathbf{y}_t^\top) \right\} \end{aligned} \quad (6)$$

In the bottom line,  $\text{diag}$  extracts only the diagonal of the in-bracket term, as we constrain  $R$  to be diagonal in Constraint 4.

Estimation for  $\pi_0$  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  $C$  matrix also has closed form. Terms involving  $C$  in equation (4) are

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

In  $f_{\lambda_2}(C; \mathbf{X}, \mathbf{Y})$ ,  $C$  is a matrix, to simplify notation and optimization, we vectorized it to a vector  $\mathbf{c}$  following the methods of Turlach et al. (2005). A closed form solution for  $\mathbf{c}$ ,  $\mathbf{c}^{\text{new}}$ , is given by the Tikhonov regularization (Tikhonov, 1943). Rearrange the elements in  $\mathbf{c}^{\text{new}}$ ,

one gets an estimation of matrix  $C$ . That is,

$$C^{\text{new}} = \text{Rearrange } c^{\text{new}} \quad (7)$$

The details of estimating  $C$  can be found in the Git Repo Appendix.

Now let's look at matrix  $A$ . Terms involving  $A$  in equation (4) are,

$$f_{\lambda_1}(A; \mathbf{X}, \mathbf{Y}) = \sum_{t=1}^T \left( \frac{1}{2} [\mathbf{x}_t - A\mathbf{x}_{t-1}]^\top [\mathbf{x}_t - A\mathbf{x}_{t-1}] \right) + \lambda_1 \|A\|_1.$$

Similar to what we have done to  $C$ ,  $f_{\lambda_1}(A; \mathbf{X}, \mathbf{Y})$  is equivalent to

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

where  $\mathbf{z}$  is a  $Td \times 1$  vector from rearranging  $\mathbf{X}$  and  $\mathbf{Z}$  is a block diagonal matrix with diagonal component  $\mathbf{Z}^\top = (\mathbf{x}_0, \dots, \mathbf{x}_{T-1})^\top$ .

$f_{\lambda_1}(A; \mathbf{X}, \mathbf{Y})$  does not have closed form solution due to the  $\ell_1$  term. Fortunately, it can be solved numerically with a Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) (Beck and Teboulle, 2009). The FISTA algorithm is detailed in the Appendix on the Git Repo.

With FISTA, matrix  $A$  can be updated:

$$A^{\text{new}} = \text{FISTA}(\|\mathbf{Z}^\top \mathbf{a}^{\text{old}} - \mathbf{z}\|_2^2, \lambda_1) \quad (8)$$

Table 1: The Complete EM Algorithm

<b>Algorithm</b> EM Algorithm for <small>MR. SID</small>	
<b>M Step</b>	
1.	$R^{\text{new}} = \text{diag} \left\{ \frac{1}{T} \sum_{t=1}^T (\mathbf{y}_t \mathbf{y}_t^\top - C^{\text{old}} \hat{\mathbf{x}}_t \mathbf{y}_t^\top) \right\}$ , as in equation (6)
2.	$\pi_0^{\text{new}} = \hat{\mathbf{x}}_0$
3.	Update $C^{\text{new}}$ , as in equation (7)
4.	Update $A^{\text{new}}$ with FISTA, as in equation (8)
5.	Stop when difference between estimations from this step and previous step is less than tolerance or maximum number of iterations reached.
<b>E Step</b>	
0.	Initialize $\theta = \{A, C, R, \pi_0\}$ as in Section 3.3, if first loop
1.	Update the expectations in (5) with the Kalman filter smoother

### 3.3 Initialization

$R$  matrix is initialized as the identify matrix, while  $\pi_0$  as 0 vector. For  $A$  and  $C$ , denote  $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_T]$ , a  $p \times T$  matrix, then the singular value decomposition (SVD) of  $\mathbf{Y}$  is  $\mathbf{Y} = \mathbf{U} \mathbf{D} \mathbf{V}^\top \approx \mathbf{U}_{p \times d} \mathbf{D}_{d \times d} \mathbf{V}_{d \times T}^\top = \mathbf{U}_{p \times d} \mathbf{X}_{d \times T}$ , 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  $\mathbf{V}_{d \times T}^\top$ .  $C$  is then initialized 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  $A$ .

Combining the initialization, E-step and M-step, a complete EM algorithm for MR. SID is addressed in Table 1. Notice that all the terms involving  $\mathbf{X}$  in the M-step are approximated with the conditional expectations calculated in E-step.

### 3.4 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.

In addition, the diagonal is directly calculated instead of first calculating the full matrix  $R$  in M step.

Finally, in E-step, the term  $K_t = V_t^{t-1} C^T (C V_t^{t-1} C^T + R)^{-1}$  involves the inverse of a large square matrix of  $p$  by  $p$ , which might be intractable. The Woodbury Matrix Identity is employed to turn a high dimensional matrix inverse to low dimensional one:  $(C V_t^{t-1} C^T + R)^{-1} = R^{-1} - R^{-1} C [(V_t^{t-1})^{-1} + C^T R^{-1} C]^{-1} C^T R^{-1}$ .

Also note that quantities like  $R^{-1}$  and  $C^T R^{-1} C$  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.

## 4 Simulations

### 4.1 Simulation Setup

Two simulations of different dimensions are performed to demonstrate parameter estimations, computational efficiency and predicting ability of `MR.SID`.

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 truncated to zero to make it sparse. After that a multiple of the identity matrix is added to  $A$ . Finally  $A$  is scaled to make sure its eigenvalues fall within  $[-1, 1]$ , thus avoiding diverging time series. Matrix  $C$  is generated as follows. Each column contains random samples from a standard Gaussian. Then each column is sorted in ascending order. Covariance  $Q$  is the identity matrix and covariance  $R$  is a multiple of the identity matrix. Initial state  $\pi_0 = 0$  is a zero vector. Pseudocode for data generation can be found in the Appendix on Git Repo.

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  $B$  as

$$d(A, B) = \operatorname{argmin}_{P \in P(n)} \left\{ \log \left[ \frac{n}{\operatorname{Trace}(P \times C_{A,B})} \right] \right\}. \quad (9)$$

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

As a result of its definition,  $d(A, B)$  is invariant to the scales of  $A$  and  $B$ . It is also invariant to a permutation of columns of either matrix. The calculation of  $d(A, B)$  is a linear assignment problem and can be solved in polynomial time with the Hungarian algorithm (Kuhn, 1955).

## 4.2 Parameter Estimation

Both the standard LDS and the penalized LDS are applied to the simulation data. Since  $A$  is sparse and  $C$  is smooth, we expect the penalized algorithms to yield better estimations

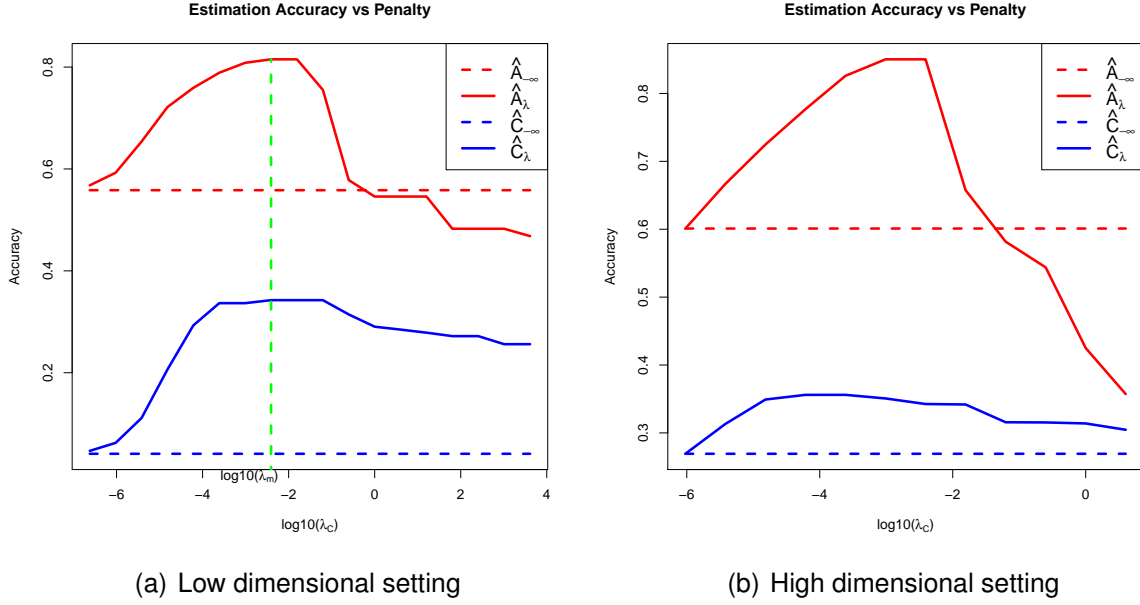


Figure 1:  $x$  axis is tuning parameter  $\lambda_C$  under log scale and  $y$  axis is the distance between truth and estimations;  $\lambda_A$  is increasing proportionally with  $\lambda_C$ . One can see that in both the low dimensional and high dimensional setting, estimation accuracies for  $A$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

with properly chosen penalty size. When the penalties are approaching 0, the penalized algorithm should converge to the non-penalized algorithm. 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  $\lambda_C$  are tested, 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 MR. SID are overlaid 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



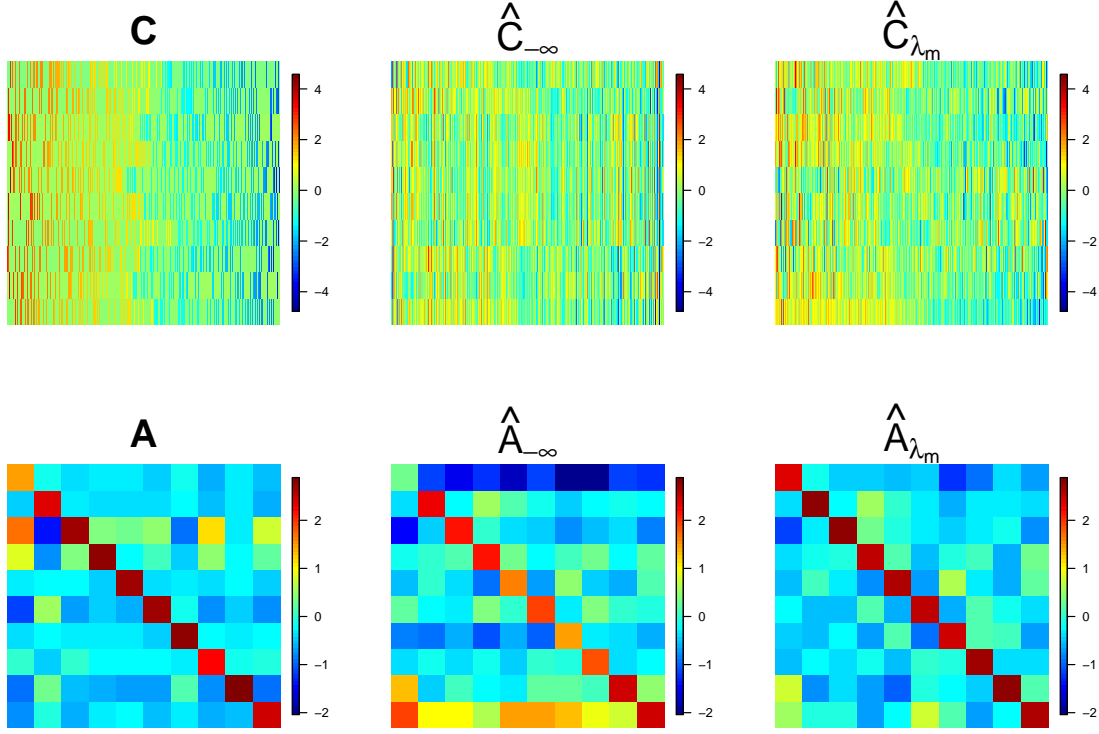


Figure 2: Row 1:  $C$  truth; non-penalized estimation of  $C$ ; optimally penalized estimation of  $C$ . Row 2:  $A$  truth; non-penalized estimation of  $A$ ; optimally penalized estimation of  $A$ .

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 0 (blue) values. However, LDS estimation  $\hat{A}_{\lambda_{-\infty}}$  is not sparse, with many positive (yellow and red) off-diagonal values.

In addition to the improved estimation accuracy, the proposed algorithm is also computational efficient and highly scalable. As a demonstration, the running times of multiple simulation scenarios are summarized in Table 2. When both  $p$  and  $d$  are high dimensional,

Table 2:  $\text{MR. SID}$  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

the algorithm can still solve the problem in a reasonable time.

### 4.3 Making Predictions

Another perspective when considering  $\text{MR. SID}$  is its ability to make predictions. When the parameters  $\theta$  and the latent state  $x_T$  are estimated, one can first use  $x_T$  to predict  $x_{T+1}$  and then use  $x_{T+1}$  to predict  $y_{T+1}$ . Similarly, more predictions  $y_{T+2}, \dots, y_{T+k}$  can be made. As shown above, properly chosen penalties give good estimations. With good estimations, one can expect accurate predictions. This idea is demonstrated with a simulation. The same data are used as in Section 4.1. 80% of the data are used for estimation, while the other 20% used for prediction. Correlations between predictions and true signals are used as a measure of accuracy. The prediction accuracy over penalty size is plotted in Figure 3.

From the plots one can see, 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 makes good sense as the same  $(\lambda_A, \lambda_C)$  pair also gives the best estimations of  $A$  and  $C$ , as in Figure 1. The latter observation provides us a way to pick tuning parameters in real applications: one can use a collection of tuning parameter pairs  $(\lambda_A, \lambda_C)$  for estimations and then predictions, the pair that gives most accurate predictions is picked. This trick is used in application Section 5.

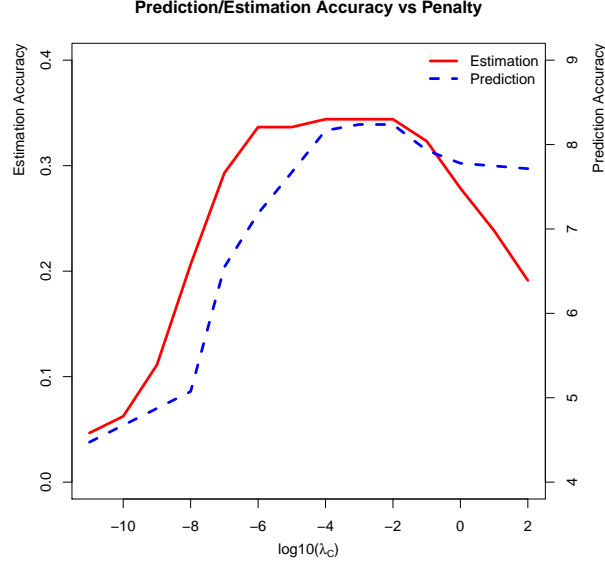


Figure 3: Estimation and prediction accuracies.  $x$ -axis is the penalty size under  $\log$  scale.  $y$ -axis represents the estimation and prediction accuracies. One see that the penalty which yields the most accurate estimation also gives the best predictions.

## 5 Application

### 5.1 Data and Motivation

MR. SID is applied to two datasets in this section: the Kirby 21 data and the Human Connectome Project (HCP) data.

The Kirby 21 data are resting-state fMRI scans acquired at the FM Kirby Research Center at the Kennedy Krieger Institute, Johns Hopkins University (Landman et al., 2011). Twenty-one healthy volunteers with no history of neurological disease each underwent two separate resting state fMRI sessions on the same scanner: a 3T MR scanner utilizing a body coil with a 2D echoplanar (EPI) sequence and eight channel phased array SENSitivity Encoding (SENSE; factor of 2) with the following parameters: TR 2s;  $3\text{mm} \times 3\text{mm}$  in plane

resolution; slice gap 1mm; and total imaging time of 7 minutes and 14 seconds. The imaging data are first preprocessed with FSL, a comprehensive library of analysis tools for fMRI, MRI and DTI brain imaging data (Smith et al., 2004). FSL is used for spatial smoothing with Gaussian kernel. Then  $\text{MR. SID}$  is applied on the smoothed data.

The Human Connectome Project (HCP) is a systematic effort to map macroscopic human brain circuits and their relationship to behavior in a large population of healthy adults (Van Essen et al., 2013; Moeller et al., 2010; Feinberg et al., 2010). 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. A comprehensive introduction of the dataset is given by Van Essen et al. (2013).

For both datasets, the motor cortex, which contains  $p = 7396$  voxels is analyzed instead of the whole brain.

Extensive research has been done to analyze the above datasets. Methods such as PCA and ICA (independent component analysis) have been applied to get spatial decompositions of the brain, as well as the functional connectivity among the decomposed regions. Motivated by these research, we are applying  $\text{MR. SID}$  to the Kirby 21 data and wish to get a spatial decomposition and a connectivity graph at the same time. It would also be very helpful, especially in medical sciences, if one can predict the brain activity from already-collected data. Therefore, as a second application,  $\text{MR. SID}$  is applied to the HCP data to predict brain activities.

## 5.2 Results

$\text{MR. SID}$  is first applied to the Kirby 21 data.

When applied to fMRI data, the model has very good interpretability. Each  $y_t$  is a snapshot of brain activity at time  $t$ . The columns of  $C$  are interpreted as time-invariant brain “point spread functions”. At each time point, the observed brain image,  $y_t$ , is a linear mixture of latent co-assemblies of neural activity  $x_t$ . Matrix  $A$  describes how  $x_t$  evolves over time.  $A$  is a directed graph if one treats each neural assembly as a vertex. Each neural assembly is spatially smooth, and connectivity across them are empirically sparse. This naturally fits into the sparsity and smoothness assumptions of MR. SID.

In this application, the number of voxels,  $p = 7396$  and number of scans,  $T = 210$ . Tuning parameters:  $\lambda_A = \lambda_C = 10^{-5}$ . Number of latent states,  $d = 11$ . Max number of iterations for EM and the regularized subproblems are both 30 steps.

To pick optimal penalty size, different combinations of  $\lambda_A$  and  $\lambda_C$ , where  $\lambda_A = \lambda_C$  are tried. The values range from  $10^{-10}$  to  $10^4$ . Then the estimations from each combination are used to make predictions. The combination that gives the best prediction is used. One can also use a grid of combinations to search for even better penalties.

To determine the number of latent states, the profile likelihood method proposed by Zhu et al. (Zhu and Ghodsi, 2006) is adopted. The method assumes eigenvalues of the data matrix come from a mixed Gaussian and use profile likelihood to pick the optimal number of latent states. Apply the method to all four scans, the numbers of latent states are 11, 6, 14 and 15 respectively. Their average,  $d = 11$ , is used.

First look at estimations of the  $A$  matrix. Let  $A_{12}$  stands for the estimated  $A$  matrix for the second scan of subject one. Similar logic applies to  $A_{11}$ ,  $A_{21}$ ,  $A_{22}$  and  $C$  matrices. These matrices contain subject-specific information. There are 6 different pairs among the 4 matrices - intuitively, the pair  $(A_{11}, A_{12})$  and  $(A_{21}, A_{22})$  should have the highest similarity, as each comes from two scans of the same subject. This idea is validated by Table 3 and

Table 3: Similarities Among Estimated  $A$  Matrices

$d(\cdot, \cdot)$ (Amari Error)	$A_{11}$	$A_{12}$	$A_{21}$	$A_{22}$
$A_{11}$	0			
$A_{12}$	<b>0.076(0.88)</b>	0		
$A_{21}$	0.105(1.05)	0.095(1.08)	0	
$A_{22}$	0.095(1.02)	0.095(1.09)	<b>0.085(0.98)</b>	0

Figure 4, which summarize similarities among the 4 matrices. The distance measure in Equation (9) is used. Another permutation-invariant measure of similarity, the Amari error (Amari et al., 1996), is also provided. The Amari error between matrices  $A$  and  $\hat{A}$  is defined as:  $E(A, \hat{A}) = \sum_{i=1}^n (\sum_{j=1}^n \frac{|p_{ij}|}{\max_k |p_{ik}|} - 1) + \sum_{j=1}^n (\sum_{i=1}^n \frac{|p_{ij}|}{\max_k |p_{kj}|} - 1)$ , where  $P = (p_{ij}) = A^{-1}\hat{A}$ . Notice a smaller  $d(A, B)$  or Amari error means higher similarity.

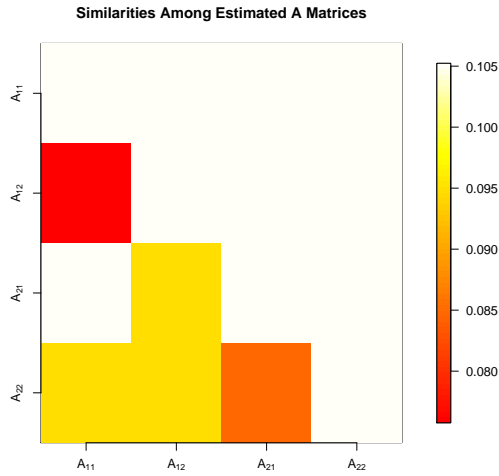


Figure 4: Similarities among the four estimated  $A$  matrices. The distance  $d(\cdot, \cdot)$  is used in this figure. The two red/orange off-diagonal pixels has the minimum distances, which correspond to the pairs of  $(A_{11}, A_{12})$  and  $(A_{21}, A_{22})$  respectively. With this similarity map, one can tell which two scans are from the same subject.

Next let's turn to the  $C$  matrix. 3D renderings of the columns of  $C_{11}$  are shown in Figure 5 (after thresholding). These regions are comparable to existing parcellations of the mortor

cortex. As an example, the blue region in Figure 5 accurately matches the dorsolateral (DM) parcel of the five-region parcellation proposed by Nebel MB et al. (Nebel et al., 2014).

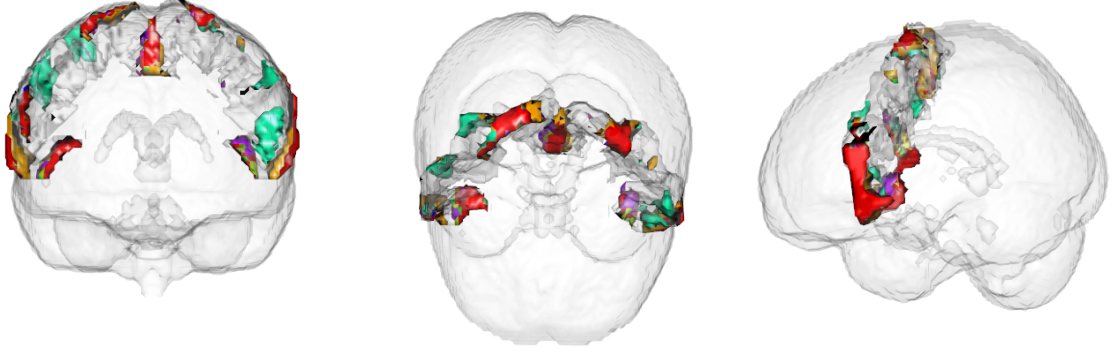


Figure 5: 3D rendering of columns of matrix  $C_{11}$ : estimation for the first scan of subject one.

As a summary,  $\text{MR. SID}$  gives a spatial decomposition of the motor cortex, as well as the sparse connectivity among the decomposed regions. The connectivity graph contains subject-specific information and can correctly group scans by subject. The decomposed regions are spatially smooth and are comparable to existing parcellations of the motor cortex.

As a second application,  $\text{MR. SID}$  is applied to the HCP data to predict brain activities.

Using the profile likelihood method,  $d = 149$  is picked. HCP data has  $T = 1200$  time points. The first  $N = 1000$  are used as training data, while the rest are used for test.

$\text{MR. SID}$  is used to predict brain activity from training data. As a comparison, the SVD method in Section 3.3 is also tried. Both methods are first used for parameter estimations, then the estimated parameters are fed into equations (1) to make  $k$ -step ahead predictions.

Pseudocode for  $k$ -step ahead predictions is given in Appendix on Git Repo.

The prediction accuracies are shown in Figure 6 (left panel). One can see `MR.SID` algorithm is giving significantly better predictions for the first 150 predictions compared to the SVD method. Considering the SVD method is also used to initialize `MR.SID` algorithm, this observation shows that `MR.SID` algorithm improves estimation accuracy significantly on top of the SVD method. 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.

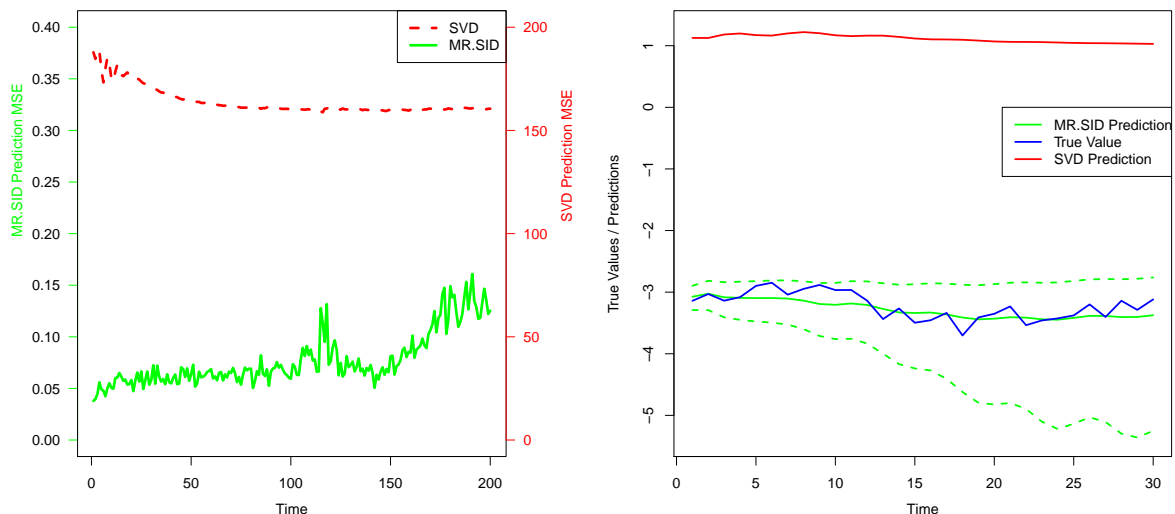


Figure 6: Prediction accuracies comparison on HCP data. *Left*: The mean squared error (MSE) is used as the accuracy measure. *Right*: Sample time series plot. The dotted green curve stands for the 60% confidence band given by `MR.SID` model. The true time series is averaged signals from a subsample 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 6 (right



panel). We see that  $\text{MR. SID}$  is giving more accurate predictions and the true signal lies in the confidence band giving by  $\text{MR. SID}$  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  $R$  (Allen et al., 2014). 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 tridiagonal and upper triangular could also be considered. Mohammad et al. have discussed the impact of auto correlation on functional connectivity, which also provides us a direction for extension (Arbabshirani et al., 2014).

Finally, the work can also be extended on the application side. Currently, only data from a few subjects are analyzed. As a next step, the model can be extended to a group version and be used to analyze more subjects. In addition,  $A$  matrix estimated by  $\text{MR. SID}$  is a potential candidate as a measure of fMRI scan reproducibility.

## References

- Allen, G. I., L. Grosenick, and J. Taylor (2014). A generalized least-square matrix decomposition. *Journal of the American Statistical Association* 109(505), 145–159.
- Amari, S.-i., A. Cichocki, H. H. Yang, et al. (1996). A new learning algorithm for blind signal separation. *Advances in neural information processing systems*, 757–763.
- Arbabshirani, M. R., E. Damaraju, R. Phlypo, S. Plis, E. Allen, S. Ma, D. Mathalon, A. Preda, J. G. Vaidya, T. Adali, et al. (2014). Impact of autocorrelation on functional connectivity. *NeuroImage*.
- Banerjee, A., J. T. Vogelstein, and D. B. Dunson (2013, December). Parallel inversion of huge covariance matrices. *arXiv preprint*.
- Beck, A. and M. Teboulle (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences* 2(1), 183–202.
- Boots, B. Learning stable linear dynamical systems.
- CHEN, S., S. A. BILLINGS, and W. LUO (1989, November). Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control* 50(5), 1873–1896.
- Doretto, G., A. Chiuso, Y. N. Wu, and S. Soatto (2003). Dynamic textures. *International Journal of Computer Vision* 51(2), 91–109.
- Feinberg, D. A., S. Moeller, S. M. Smith, E. Auerbach, S. Ramanna, M. Gunther, M. F. Glasser, K. L. Miller, K. Ugurbil, and E. Yacoub (2010). Multiplexed echo planar imaging for sub-second whole brain fmri and fast diffusion imaging. *PloS one* 5(12), e15710.

- Friston, K., L. Harrison, and W. Penny (2003, August). Dynamic causal modelling. *NeuroImage* 19(4), 1273–1302.
- Ghahramani, Z. and G. E. Hinton (1996). Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science.
- Hsieh, C.-J., M. A. Sustik, I. S. Dhillon, P. K. Ravikumar, and R. Poldrack (2013). BIG & QUIC: Sparse Inverse Covariance Estimation for a Million Variables. In *Advances in Neural Information Processing Systems*, pp. 3165–3173.
- Johansen, S. r. (1988, June). Statistical analysis of cointegration vectors. *Journal of Economic Dynamics and Control* 12(2-3), 231–254.
- Kalman, R. E. (1960, March). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering* 82(1), 35.
- Kalman, R. E. (1963, January). Mathematical Description of Linear Dynamical Systems. *Journal of the Society for Industrial and Applied Mathematics Series A Control* 1(2), 152–192.
- Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly* 2(1-2), 83–97.
- Landman, B. A., A. J. Huang, A. Gifford, D. S. Vikram, I. A. L. Lim, J. A. Farrell, J. A. Bogovic, J. Hua, M. Chen, S. Jarso, et al. (2011). Multi-parametric neuroimaging reproducibility: A 3-t resource study. *Neuroimage* 54(4), 2854–2866.

- Ljung, L. (1998). System Identification. In A. Procházka, J. Uhlí, P. W. J. Rayner, and N. G. Kingsbury (Eds.), *Signal Analysis and Prediction*, Applied and Numerical Harmonic Analysis. Boston, MA: Birkhäuser Boston.
- Moeller, S., E. Yacoub, C. A. Olman, E. Auerbach, J. Strupp, N. Harel, and K. Uğurbil (2010). 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.
- Nebel, M. B., S. E. Joel, J. Muschelli, A. D. Barber, B. S. Caffo, J. J. Pekar, and S. H. Mostofsky (2014). Disruption of functional organization within the primary motor cortex in children with autism. *Human brain mapping* 35(2), 567–580.
- Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2), 257–286.
- Roweis, S. and Z. Ghahramani (1999). A unifying review of linear gaussian models. *Neural computation* 11(2), 305–345.
- Shumway, R. H. and D. S. Stoffer (1982). An approach to time series smoothing and forecasting using the em algorithm. *Journal of time series analysis* 3(4), 253–264.
- Smith, S. M., M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. Behrens, H. Johansen-Berg, P. R. Bannister, M. De Luca, I. Drobnjak, D. E. Flitney, et al. (2004). Advances in functional and structural mr image analysis and implementation as fsl. *Neuroimage* 23, S208–S219.
- Tikhonov, A. N. (1943). On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, Volume 39, pp. 195–198.

- Turlach, B. A., W. N. Venables, and S. J. Wright (2005). Simultaneous variable selection. *Technometrics* 47(3), 349–363.
- Van Essen, D. C., S. M. Smith, D. M. Barch, T. E. Behrens, E. Yacoub, K. Ugurbil, W.-M. H. Consortium, et al. (2013). The wu-minn human connectome project: an overview. *Neuroimage* 80, 62–79.
- Van Overschee, P. and B. De Moor (1994). N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica* 30(1), 75–93.
- Xie, Y., J. Huang, and R. Willett (2013, February). Change-Point Detection for High-Dimensional Time Series With Missing Data. *IEEE Journal of Selected Topics in Signal Processing* 7(1), 12–27.
- Zaharia, M., M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica (2010). Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, Volume 10, pp. 10.
- Zhu, M. and A. Ghodsi (2006). Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics & Data Analysis* 51(2), 918–930.