An M-Estimator for Reduced-Rank 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]Child Mind Institute

[f]Dept. of Biomedical Engineering and Institute for Computational Medicine, Johns Hopkins University

Abstract

High-dimensional time-series data from a wide variety of domains, such as neuroscience, are being generated every day. 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 data, due to both computational and statistical reasons. We mitigate both kinds of issues by 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 tricks, yields an estimator that is computationally efficient and numerically stable. Simulations and real data examples demonstrate the usefulness of this approach in a variety of problems. In particular, we demonstrate that Mr. Sid can accurately estimate spatial filters, connectivity graphs, and time-courses from native resolution functional magnetic resonance imaging data. Mr. Sid therefore enables big time-series data to be analyzed using standard methods, readying the field for further generalizations including nonlinear and non-Gaussian state-space models.

***keywords*: high dimension, image processing, parameter estimation, state-space model, time series analysis**

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 system (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 tools for time-series prediction with an LDS, given known parameters (Kalman, 1960). In practice, however, for many LDS's, the parameters are unknown and must be estimated in a process often called *system identification* (Ljung, 1998). To the best of our knowledge, currently there does not exist a methodology that provides parameter estimates and predictions from ultra-high-dimensional time-series data (e.g. $p > 10{,}000$).

The challenges associated with high-dimensional time-series estimation and prediction are multifold. First, naïvely, such models include dense $p \times p$ matrices, which are often too large to store, much less invert in memory. Several recent efforts to invert large sparse matrices using a series of computational tricks show promise, though they are still extremely computationally expensive (Hsieh et al., 2013; Banerjee et al., 2013). Second, estimators behave poorly due to numerical instability. Reduced-rank LDS models can partially address this problem by reducing the number of latent states. (CHEN et al., 1989). However, without further constraints, the dimensionality of the latent states would be reduced to such an extent that it would 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 those built with Spark, might help overcome this problem. However, it would lead to 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 (MR. SID). By assuming the dimensionality of the latent state space is small (i.e. reduced-rank), relative to the 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, via 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 reduce the computational burden significantly.

These three techniques combined 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 show numerically that the hyperparameters can be selected to minimize prediction error on held-out data. Finally, we use MR. SID to estimate functional connectomes from the 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 a single unified approach has been used to estimate partitions and functional connectomes directly from the high-dimensional data.

This work presents a new analysis of a model which has only been implemented in low-dimensional settings, and paves the way for high-dimensional implementation. Though primitive, it is a first step for essentially any high-dimensional time series analysis, control system identification, and spatiotemporal analysis. To enable extensions, generalizations, and additional applications, the code for the core functions and generating each of the figures is freely available on Github (https://github.com/shachen/PLDS/).

2 The Model

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

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

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

$$\begin{aligned} P(\boldsymbol{y}_t|\boldsymbol{y}_0,\ldots,\boldsymbol{y}_{t-1},\boldsymbol{x}_0,\ldots,\boldsymbol{x}_t) \approx P(\boldsymbol{y}_t|\boldsymbol{x}_t), \\ P(\boldsymbol{x}_t|\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{t-1}) \approx P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}). \end{aligned} \tag{1}$$

A linear dynamical system (LDS) further assumes that both terms in 1 are linear Gaussian functions, which when written as an iterative random process, yield the standard matrix update rules:

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

where A is a $d \times d$ state transition matrix and C is a $p \times d$ 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 of as a continuous version of the hidden Markov model (HMM), where the columns of C stand for the hidden states and one observes a single state at time t. Unlike HMM, LDS 1 allows one to observe a linear combination of multiple states. A is the analogy of the state transition matrix, which describes 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 LDS model itself is unidentifiable. Three minimal constraints are introduced for identifiability:

Constraint 1: Q is the identity matrix

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

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

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

The logic for Constraint 1 is as follows. Since the 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 Λ 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 $\mathbf{z} = \Lambda^{-1/2}E^T\boldsymbol{x}$, 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 the 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. This is reasonable, because 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, which means that the R matrix is very large. One cannot 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).

The 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 ℓ_1 penalty on A is used to impose the sparsity constraint. 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, it is desirable for 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 ℓ_2 penalty on columns of C is used to enforce smoothness.

With all those constraints, the model becomes:

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

where A is a sparse matrix and C has smooth columns. Let $\theta = \{A, C, R, \pi_0\}$ represent all unknown parameters, while $P(\boldsymbol{X}, \boldsymbol{Y})$ represents the full likelihood. Then, combining model 2 and the constraints on A and C leads us to an optimization problem:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left\{ -\log P_\theta(\boldsymbol{X}, \boldsymbol{Y}) + \lambda_1 \|A\|_1 + \lambda_2 \|C\|_2^2 \right\} \tag{3}$$

where λ_1 and λ_2 are tuning parameters and $\| \mathbf{.} \|_p$ represents the p-norm of a vector. Equivalently, this problem has the following dual form:

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

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 3: given only one 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 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 3 due to the introduced penalty terms. We therefore developed an algorithm called M-estimation for Reduced-rank System IDentification (MR. SID), as detailed in the following.

By the chain rule, the full likelihood is

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

where $\mathbb{1}_{\pi_0}(\boldsymbol{x}_0)$ is the indicator function. 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:

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

By replacing $\log P(\boldsymbol{X}, \boldsymbol{Y})$ in problem 3 with Eq. 4, one gets

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

Let the target function in the curly braces be denoted as $\boldsymbol{\Phi}(\theta, \boldsymbol{Y}, \boldsymbol{X})$. Then $\boldsymbol{\Phi}$ 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, $\Gamma = E[\log P(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{Y})]$. This quantity depends on three expectations: $E[\boldsymbol{x}_t|\boldsymbol{Y}]$, $E[\boldsymbol{x}_t\boldsymbol{x}_t^{\intercal}|\boldsymbol{Y}]$ and $E[\boldsymbol{x}_t\boldsymbol{x}_{t-1}^{\intercal}|\boldsymbol{Y}]$. For simplicity, we denote their finite sample estimators by:

$$\hat{\boldsymbol{x}}_t \equiv E[\mathbf{x_t}|\boldsymbol{Y}],\ \hat{P}_t \equiv E[\boldsymbol{x}_t\boldsymbol{x}_t^{\intercal}|\boldsymbol{Y}],\ \hat{P}_{t,t-1} \equiv E[\boldsymbol{x}_t\boldsymbol{x}_{t-1}^{\intercal}|\boldsymbol{Y}]. \tag{6}$$

Expectations 6 are estimated with a Kalman filter/smoother, which is detailed in the Appendix. 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 $\boldsymbol{\Phi}(\theta, \boldsymbol{Y}, \boldsymbol{x})$, setting them to zero, and then solving the equations.

Let the estimations from the previous step be denoted as $\theta^{\text{old}} = \{A^{\text{old}}, C^{\text{old}}, R^{\text{old}}, \pi_0^{\text{old}}\}$ and the current estimations as $\theta^{\text{new}} = \{A^{\text{new}}, C^{\text{new}}, R^{\text{new}}, \pi_0^{\text{new}}\}$. The estimation for the R matrix has a closed form, as follows:

$$\begin{aligned}\frac{\partial \mathbf{\Phi}}{\partial R^{-1}} &= \frac{T}{2}R - \sum_{t=1}^{T}\big(\frac{1}{2}\boldsymbol{y}_t\boldsymbol{y}_t^{\intercal} - C\hat{\boldsymbol{x}}_t\boldsymbol{y}_t^{\intercal} + \frac{1}{2}C\hat{P}_tC^{\intercal}\big) = 0 \\ \implies R &= \frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{y}_t\boldsymbol{y}_t^{\intercal} - C^{\text{new}}\hat{\boldsymbol{x}}_t\boldsymbol{y}_t^{\intercal}) \\ \implies R^{\text{new}} &= \operatorname{diag}\Big\{\frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{y}_t\boldsymbol{y}_t^{\intercal} - C\hat{\boldsymbol{x}}_t\boldsymbol{y}_t^{\intercal})\Big\}\end{aligned} \tag{7}$$

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

The estimation for π_0 has a closed form. The relevant term $\log(\mathbb{1}_{\pi_0}(\hat{\boldsymbol{x}}_0))$ is minimized only when $\pi_0^{\text{new}} = \hat{\boldsymbol{x}}_0$.

The estimation for the C matrix also has a closed form. Terms involving C in Eq. 5 are

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

In $f_{\lambda_2}(C; \boldsymbol{X}, \boldsymbol{Y})$, 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 $\mathbf{c}$, denoted $\mathbf{c}^{\text{new}}$, is given by the Tikhonov regularization (Tikhonov, 1943). By rearranging the elements in $\mathbf{c}^{\text{new}}$, one gets an estimation of matrix C. That is,

$$C^{\text{new}} = \text{Rearrange } \mathbf{c}^{\text{new}} \tag{8}$$

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

Now consider matrix A. Terms involving A in Eq. 5 are

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

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

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

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

$f_{\lambda_1}(A; \boldsymbol{X}, \boldsymbol{Y})$ does not have a closed form solution due to the ℓ_1 term. However, 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.

With FISTA, matrix A can be updated as follows:

$$A^{\text{new}} = \text{FISTA}(\|\mathbf{Z}^{\intercal}\mathbf{a}^{\text{old}} - \mathbf{z}\|_2^2, \quad \lambda_1) \tag{9}$$

3.3 Initialization

The R matrix is initialized as the identity matrix, while π_0 is initialized as the $\mathbf{0}$ 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 $\boldsymbol{Y} = \mathbf{U}\mathbf{D}\mathbf{V}^{\intercal} \approx \mathbf{U}_{p\times d}\mathbf{D}_{d\times d}\mathbf{V}^{\intercal}_{d\times T} = \mathbf{U}_{p\times d}\boldsymbol{X}_{d\times T}$, 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 $\mathbf{V}^{\intercal}_{d\times T}$. C is then initialized 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.

Combining the initialization, E-step, and M-step, a complete EM algorithm for MR. SID is ad-

Table 1: The Complete EM Algorithm

```
Algorithm EM Algorithm for Mr. Sid
1. Initialize $\theta = \{A, C, R, \pi_0\}$ as in Section 3.3
2. While convergence criteria are unmet
E Step
3. Update the expectations in Eq. 6 with the Kalman filter-smoother
M Step
4. $R^{\text{new}} = \text{diag}\left\{\frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{y}_t\boldsymbol{y}_t^{\intercal} - C^{\text{old}}\hat{\boldsymbol{x}}_t\boldsymbol{y}_t^{\intercal})\right\}$, as in Eq. 7
5. $\pi_0^{\text{new}} = \hat{\boldsymbol{x}}_0$
6. Update $C^{\text{new}}$, as in Eq. 8
7. Update $A^{\text{new}}$ with FISTA, as in Eq. 9
```

dressed in Table 1. Notice that all the terms involving $\boldsymbol{X}$ in the M-step are approximated with the conditional expectations calculated in the 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 the covariance matrix R, which is a $p \times p$ matrix. The following computational techniques are utilized to make the code highly efficient and scalable. For the covariance matrix R, with constraint 4 (i.e. the diagonal assumption), we employ a sparse matrix to represent R, and only the diagonal elements are directly calculated. In the E-step, the term $K_t = V_t^{t-1}C^{\intercal}(CV_t^{t-1}C^{\intercal} + R)^{-1}$ involves the inverse of a large square $p \times p$ matrix, which might be intractable. The Woodbury Matrix Identity is employed to turn a high dimensional matrix inverse to a low dimensional one: $(CV_t^{t-1}C^{\intercal} + R)^{-1} = R^{-1} - R^{-1}C[(V_t^{t-1})^{-1} + C^{\intercal}R^{-1}C]^{-1}C^{\intercal}R^{-1}$. Note that quantities like R^{-1} and $C^{\intercal}R^{-1}C$ 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

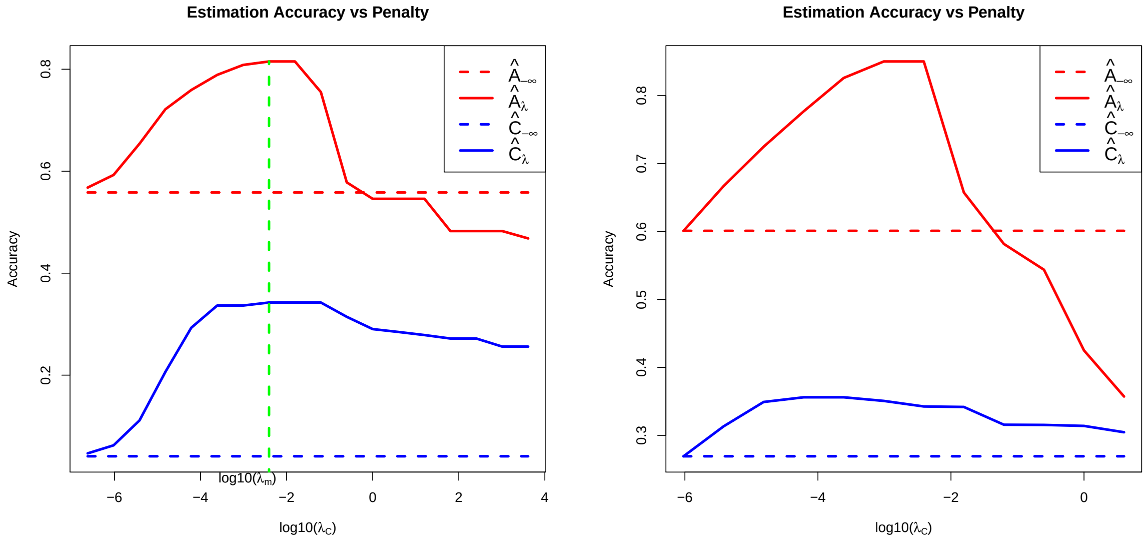
Two simulations of different dimensions are performed to demonstrate the parameter estimations, computational efficiency, and predicting ability of MR. SID. In the low dimensional setting, $p = 300$, $d = 10$, and $T = 100$. A is first generated from a random matrix, then elements with small absolute values are truncated to zero to make it sparse. Afterwards, 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 then 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 $\pi_0 = \mathbf{0}$ is a zero vector. Pseudocode for data generation can be found in the Appendix.

In the high-dimensional setting, $p = 10000$, $d = 30$, and $T = 100$. The parameters are generated in the same manner. To evaluate the accuracy of estimations, we elect to define the distance between two matrices A and B as

$$d(A, B) = \underset{P \in P(n)}{\operatorname{argmin}} \left\{ \log\Big[\frac{n}{\text{Trace}(P \times C_{A,B})}\Big] \right\} \tag{10}$$

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.

Both the standard LDS and MR. SID are applied to the simulation data. Estimation accuracies are plotted against penalty sizes in Figure 1. From the plot, one sees that the prediction accuracy



(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.

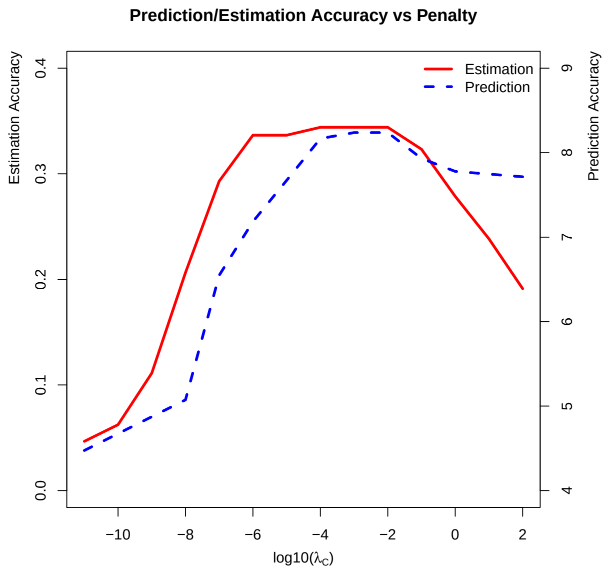


Figure 2: Estimation and prediction accuracies. The x-axis represents the penalty size on a $\log$ scale. The y-axis represents the estimation and prediction accuracies. Note that the penalty which yields the most accurate estimation also gives the best prediction.

first improves, then drops when the penalties increase. MR. SID is also used for time series prediction, and the result is plotted in Figure 2. The prediction accuracy peaks when the penalty coefficients λ_A and λ_C are around 10^{-3}. This makes sense, as the same (λ_A, λ_C) pair also gives the best estimations of A and C, as seen 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 (λ_A, λ_C) for estimations (with train data) and subsequently for predictions (with test data). The pair that gives the most accurate out-of-sample predictions is picked. This trick is used in Section 5.

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 were acquired from the FM Kirby Research Center at the Kennedy Krieger Institute, an affiliate of 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 an eight-channel phased-array coil with SENSitivity Encoding (SENSE; factor of 2) with the following parameters: TR 2s; 3mm×3mm in plane resolution; slice gap 1mm; and total imaging time of 7 minutes and 14 seconds. The imaging data were first preprocessed with FSL, a comprehensive library of analysis tools for fMRI, MRI, and DTI brain imaging data (Smith et al., 2004). FSL was used for spatial smoothing with a Gaussian kernel. Then MR. SID was applied on the smoothed data. The number of scans was $T = 210$.

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 1,200 subjects were scanned using all four of these modalities on a customized 3T scanner. All scans consist of 1,200 time points. A comprehensive introduction of the dataset is given by Van Essen et al. (2013).

Extensive research has been done to analyze the above datasets. Methods such as PCA and ICA (Independent Component Analysis) have been applied to obtain spatial decompositions of the brain, as well as the functional connectivity among the decomposed regions. Thus, for our first application, we applied MR. SID to the Kirby 21 data with the intent of obtaining both a spatial decomposition graph and a connectivity graph. As a second application, MR. SID was applied to the HCP data to predict brain activities. For both datasets, the motor cortex, which contains $p = 7396$ voxels, is analyzed instead of the whole brain.

5.2 Results

MR. SID was first applied to the Kirby 21 data. The max number of iterations for EM and the regularized subproblems were both 30 steps. To pick the optimal penalty size, different values of $\lambda_A = \lambda_C$, were attempted. The values ranged from 10^{-10} to 10^4. Then the estimations from each combination were used to make predictions. We determined that the best value was 10^{-5}, as it gives the most accurate out-of-sample predictions. One can also try 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) was adopted. The method assumes eigenvalues of the data matrix come from a mixed Gaussian, and uses 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, $d = 11$, was used.

First, we will consider estimations of the A matrix. Let A_{12} stand for the estimated A matrix for the second scan of subject 1. Similar logic applies to the 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 2 and Figure 3, which summarize similarities among the 4 matrices. The distance measure in Eq. 10 was used. The Amari error (Amari et al., 1996), which is another permutation-invariant mea-

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

sure of similarity, 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.

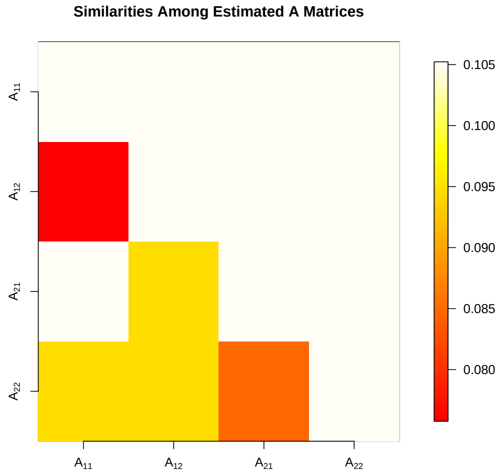


Figure 3: Similarities among the four estimated A matrices. The distance $d(\cdot,\cdot)$ was used in this figure. The two off-diagonal pixels that have the minimum distances, i.e. the red pixel and the orange pixel, correspond to the pairs of (A_{11},A_{12}) and (A_{21},A_{22}) respectively. With this similarity map, one can tell which two scans are from the same subject.

Next, consider the C matrix. 3D renderings of the columns of C_{11} after thresholding are shown in Figure 4. These regions are comparable to existing parcellations of the motor cortex. As an example, the blue region in Figure 4 accurately matches the dorsomedial (DM) parcel of the five-region parcellation proposed by Nebel MB et al. (Nebel et al., 2014).

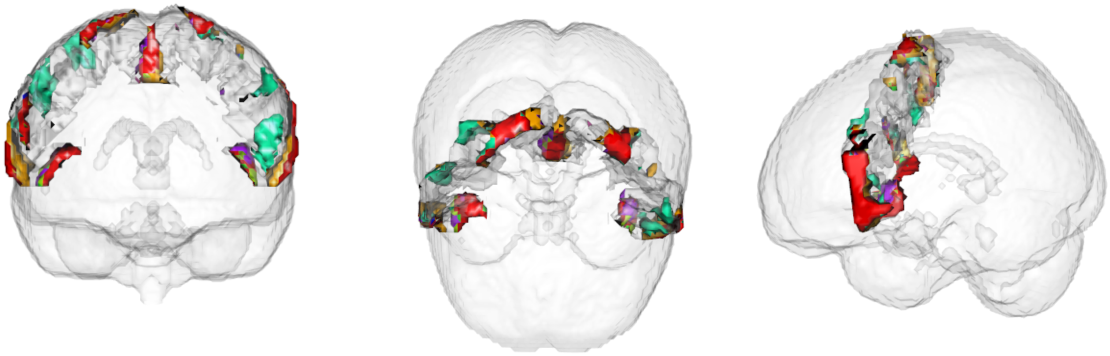


Figure 4: 3D rendering of columns of matrix C_{11}: estimation for the first scan of subject one.

When applied to fMRI data, the model has very good interpretability. Each $\boldsymbol{y}_t$ 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, $\boldsymbol{y}_t$, 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 is a directed graph if one treats each neural assembly as a vertex. Each neural assembly is spatially smooth, and connectivity across them is empirically sparse. This naturally fits into the sparsity and smoothness assumptions of MR. SID.

To summarize, 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.

For the second application, MR. SID was 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 $N = 1000$ (about 80%) were used as training data, while the rest were used as out-of-sample test data. MR. SID was used to predict brain activity from the training data. As a comparison, the SVD method from Section 3.3 was also attempted. Both methods were first used for parameter estimations, then the estimated parameters were fed into equations 2 to make k-step ahead predictions. Pseudocode for k-step ahead predictions is given in the Appendix.

The prediction accuracies are shown in Figure 5 (left panel). One can see that the MR. SID algorithm gives significantly better predictions for the first 150 predictions compared to the SVD method. Considering the SVD method is also used to intialize the MR. SID algorithm, this observation shows that the MR. SID algorithm improves estimation accuracy significantly compared to using the SVD method alone. Another observation is that the performance of the MR. SID algorithm suffers when one predicts too far into the future (> 150 steps). This is reasonable because the prediction errors from each step will accumulate, yielding deteriorating predictions as the number of steps increase.

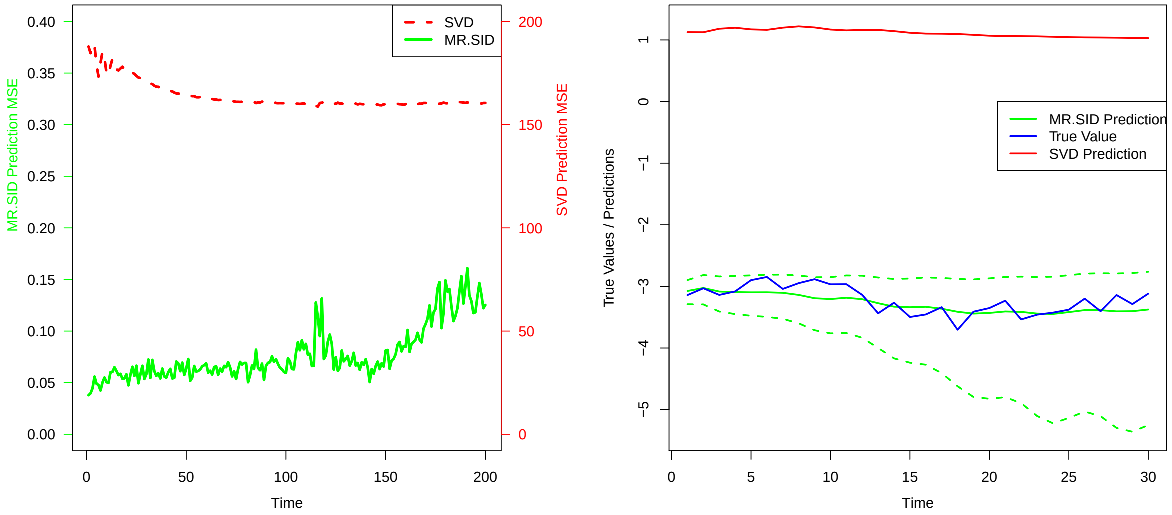


Figure 5: Comparison of prediction accuracies on HCP data. *Left:* Accuracies of Mr. Sid and SVD predictions over time, with accuracy measured through mean squared error (MSE). *Right:* Sample time series plot. The dotted green curve represents the 60% confidence band given by the Mr. Sid model. The true time series consists of averaged signals from a subsample of voxels. The predictions were also averaged over the same subsample. The confidence band was estimated based on the covariance matrix of these voxels. A subsample of 20 voxels were selected for this experiment, to avoid the calculation of large covariance matrices. All values were log-scaled for plotting purposes.

A sample plot of the true time series and predicted values are shown in Figure 5 (right panel). We see that Mr. Sid gives more accurate predictions, and the true signal lies in the confidence band given by the Mr. Sid model. Note that the confidence band is wider for predictions farther into the future, which is a result of the accumulated errors discussed previously.

6 Discussion

We have taken a first step towards the modeling and estimation of high-dimensional time-series data. The proposed method balances both statistical and computational considerations. Indeed, much like the Kalman Filter-Smoother for modeling time-series data, and the Baum-Welch algorithm for system identification act as “primitives” for time-series data analysis, Mr. Sid can act as a primitive for similar time-series analysis when the dimensionality is significantly larger than the number of time steps. Via simulations we demonstrated the efficacy of our methods. Then, by applying the proposed approach to fMRI scans of the motor cortex of healthy adults, we identified limited sub-regions (networks) from 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 into the covariance matrix R (Allen et al., 2014). The idea is that R should be general enough to be flexible, but sufficiently restricted to make the model useful. Many other methods, e.g.

those that use tridiagonal and upper triangular matrices, could also be considered. Mohammad et al. have discussed the impact of autocorrelation on functional connectivity, which also provides some 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 have been analyzed. As a next step, the model can be extended to a group version and be used to analyze more subjects. In addition, the A matrix estimated by MR. SID could potentially be used as a measure of fMRI scan reproducibility. All of the work presented herein is available from our github repository,

Acknowledgements

This work is graciously supported by the Defense Advanced Research Projects Agency (DARPA) SIMPLEX program through SPAWAR contract N66001-15-C-4041 and DARPA GRAPHS N66001-14-1-4028.

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. 22

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. 18

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*. 23

Banerjee, A., J. T. Vogelstein, and D. B. Dunson (2013, December). Parallel inversion of huge covariance matrices. *arXiv preprint*. 3

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. 12

Boots, B. Learning stable linear dynamical systems. 8

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. 3

Doretto, G., A. Chiuso, Y. N. Wu, and S. Soatto (2003). Dynamic textures. *International Journal of Computer Vision 51*(2), 91–109. 8

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. 17

Friston, K., L. Harrison, and W. Penny (2003, August). Dynamic causal modelling. *NeuroImage 19*(4), 1273–1302. 2

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. 8

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. 3

Johansen, S. r. (1988, June). Statistical analysis of cointegration vectors. *Journal of Economic Dynamics and Control 12*(2-3), 231–254. 2

Kalman, R. E. (1960, March). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering 82*(1), 35. 2

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. 2

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. 17

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. 2

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. 17

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. 19

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*(2), 257–286. 3

Roweis, S. and Z. Ghahramani (1999). A unifying review of linear gaussian models. *Neural computation 11*(2), 305–345. 7

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. 8

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. 17

Tikhonov, A. N. (1943). On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, Volume 39, pp. 195–198. 11

Turlach, B. A., W. N. Venables, and S. J. Wright (2005). Simultaneous variable selection. *Technometrics 47*(3), 349–363. 11

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. 17

Van Overschee, P. and B. De Moor (1994). N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica 30*(1), 75–93. 8

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. 2

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. 3

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. 18