Principled Neuro-Functional Connectivity Discovery

Kejun Huang* huang663@umn.edu
Nicholas D. Sidiropoulos* nikos@ece.umn.edu
Evangelos E. Papalexakis† epapalex@cs.cmu.edu
Christos Faloutsos† christos@cs.cmu.edu
Partha Pratim Talukdar‡ ppt@serc.iisc.in
Tom M. Mitchell† tom.mitchell@cmu.edu

Abstract

How can we reverse-engineer the brain connectivity, given the input stimulus, and the corresponding brain-activity measurements, for several experiments? We show how to solve the problem in a principled way, modeling the brain as a linear dynamical system (LDS), and solving the resulting "system identification" problem after imposing sparsity and non-negativity constraints on the appropriate matrices. These are reasonable assumptions in some applications, including magnetoencephalography (MEG).

There are three contributions: (a) *Proof*: We prove that this simple condition resolves the ambiguity of similarity transformation in the LDS identification problem; (b) *Algorithm*: we propose an effective algorithm which further induces *sparse* connectivity in a principled way; and (c) *Validation*: our experiments on semi-synthetic (C. elegans), as well as real MEG data, show that our method recovers the neural connectivity, and it leads to interpretable results.

1 Introduction

In computational neuroscience, one of the major research challenges is estimating the *functional connectivity* of the brain, i.e. a relation between neurons (or groups of neurons) which encodes co-activation of the neurons involved. Functional connectivity is often determined via cross-correlation or mutual information statistics [2, 11], albeit these approaches do not explicitly model neuronal state dynamics. An introductory overview of techniques for estimating brain connectivity can be found in [16].

Here, we want to estimate the functional connectivity of the brain, under the following experimental regime: a human subject is presented with a stimulus (in particular concrete nouns of the English language) as well as a task (such as answering a simple question regarding the shown noun, like *is it alive?*, *can you buy it?* and so on), and their brain activity is measured [1] over a course of a few seconds. Recently, [13] proposed the "GeBM" model, a simple model for the brain, that successfully captures the temporal dynamics of the brain. The "GeBM" model is as follows:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t), \end{aligned} \tag{1.1}$$

where $\mathbf{u}(t)$ is the stimulus signal, $\mathbf{y}(t)$ is the observed brain activity measured via MEG, $\mathbf{x}(t)$ is the latent brain activity, $\mathbf{A}$ is the functional connectivity matrix, $\mathbf{B}$ is the stimulus matrix, and $\mathbf{C}$ models the measurement that maps the internal state of the brain into MEG sensor values. In the original paper, the "GeBM" model is solved using "system identification" from control theory, and an ad-hoc, greedy method, to sparsify the connectivity matrix $\mathbf{A}$.

In this work, we formalize the problem, and provide a principled and theoretically sound treatment of sparse system identification under an additional nonnegativity condition on $\mathbf{C}$, with application to brain functional connectivity estimation. Our main contributions are the following: (a) a *rigorous proof* of identifiability for the constrained problem we propose; (b) an effective, two-stage *algorithm*; and (c) *validation* of both, using semi-synthetic and real (MEG) data.

Figure 1 shows an illustration of our results on real MEG data. The left shows the recovered graph, and the right part shows the corresponding adjacency matrix. Notice that there are several 'white' (= empty) cells, exactly because our algorithm enforces sparsity. See the experiments section for more details.

*University of Minnesota
†Carnegie Mellon University
‡Indian Institute of Science

[1] Measurements may be taken either via Magnetoencephalography (MEG) or functional magnetic resonance imaging (fMRI), although the former offers finer temporal granularity and is preferred when fine grained temporal dynamics are considered.

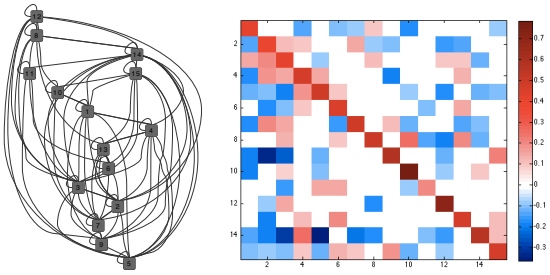

Figure 1: The neural connectivity (left), and its corresponding adjacency matrix (right), obtained from real MEG data.

2 Identifiability

State-space and, more generally, dynamic latent variable (e.g., Markov) models have a long history across science, and neural data analysis in particular [18, 14]. Even the simplest dynamic models, though, can only be identified up to inherent indeterminacies which generally hide the underlying connectivity pattern. For the linear state-space model in (1.1), for example, this comes in the form of a simplicity transformation that alters the structure of $\mathbf{A}$. Such indeterminacy is inherent to the model in (1.1), and is also borne out of classical subspace-based system identification methods [9, 8], which can only provide $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$ satisfying

$$\hat{\mathbf{A}} = \mathbf{M}\mathbf{A}\mathbf{M}^{-1}, \quad \hat{\mathbf{B}} = \mathbf{M}\mathbf{B}, \quad \hat{\mathbf{C}} = \mathbf{C}\mathbf{M}^{-1},$$

for some unknown non-singular matrix $\mathbf{M}$, due to rotational freedom. The mapping from $(\mathbf{A}, \mathbf{B}, \mathbf{C}) \to (\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}})$ is known as a *similarity transformation*. A key message of this paper is that sparsity and non-negativity of $\mathbf{C}$ can overcome this limitation and render not only $\mathbf{C}$, but also $\mathbf{A}$ and $\mathbf{B}$ identifiable without rotational ambiguity (except the unavoidable permutation and scaling).

For MEG sensors, the assumption that $\mathbf{C}$ is non-negative and sparse can be motivated as follows. The brain activity recorded by MEG is limited to very low frequencies, typically ≤ 30 Hz [12]. Since the corresponding wavelength is so much larger than the size of a skull, the spatial phase variation of the magnetic wave from one sensor to the next is insignificant, i.e., the MEG sensors are approximately *in phase*, hence $\mathbf{C}$ can be assumed non-negative. Furthermore, since field intensity decays very fast as the distance between the source and the sensor increases, a MEG sensor only measures (diffuse) sources that are close to it, leading to a sparse $\mathbf{C}$. Similar arguments can be made for electroencephalography (EEG) [15].

Notation We denote the true system matrices as $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. The matrices $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, and $\hat{\mathbf{C}}$ represent the results obtained from the subspace method, i.e., they are unconstrained. If we take the constraints into consideration, the estimates are denoted $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}$. Generally speaking, when we study identifiability, we compare $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ with $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, $\tilde{\mathbf{C}}$, and when we design algorithms, we use $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$, and the constraints to obtain $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}$.

Since linear dynamical systems can generally only be identified up to a similarity transformation, special transformations can be used to bring the system matrices into certain convenient forms. In the controls field, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are often put into *canonical forms*, such as the controller forms or the observer forms [1, Sec. 3.4], in which case the structure of $\mathbf{A}$ is confined to a specific pattern. Since our main purpose here is to *discover* the underlying structure of the true $\mathbf{A}$, *imposing* structure through transformation to some canonical form is generally inappropriate.

One interesting difference between our model and the classical controls literature is that the dimension of the output is larger than the number of states[2]. In other words, the matrix $\mathbf{C}$ is tall. Moreover, because of the nature of MEG sensors, we can also assume that $\mathbf{C}$ is sparse and takes only non-negative values. In this section, we will first propose a condition under which a non-negative $\mathbf{C}$ is identifiable, and then study the connection between sparsity, non-negativity, and the proposed condition.

Throughout this paper we will assume that $\mathbf{C}^T\mathbf{1} = \mathbf{1}$, because otherwise we can scale the columns of $\mathbf{C}$ to satisfy that, which will result in a similarity transformation with a diagonal matrix, thus does not change the structure of the system.

Since we are given $\hat{\mathbf{C}}$, which we know is a transformed version of a non-negative matrix $\mathbf{C}$, and its columns are scaled to sum up to 1, we should be able to find a matrix $\mathbf{M}$ that satisfies

$$\hat{\mathbf{C}}\mathbf{M} \geq 0, \quad \mathbf{M}^T\hat{\mathbf{C}}^T\mathbf{1} = \mathbf{M}^T\mathbf{1} = \mathbf{1}.$$

In fact, there are clearly infinitely many $\mathbf{M}$ that satisfy that. Therefore, we can set up a criterion and try to find the one with the maximum $|\det \mathbf{M}|$ (one can relate this idea to SVM, in which case there are infinitely many linear separators and we seek for the

[2] As we discovered from rank analysis of experimental MEG data.

one with the maximum "margin"), i.e.,

$$\begin{aligned} &\underset{\mathbf{M}}{\text{maximize}} && |\det \mathbf{M}|, \\ &\text{subject to} && \hat{\mathbf{C}}\mathbf{M} \geq 0, \quad \mathbf{M}^T\mathbf{1} = \mathbf{1}, \end{aligned} \tag{2.2}$$

Geometrically, (2.2) tries to find the simplicial cone that contains all the row vectors of $\hat{\mathbf{C}}$, and with minimum "volume" [10]. Next, we will propose a condition under which the true $\mathbf{C}$ can be recovered by solving (2.2) and then set the estimate as $\hat{\mathbf{C}}\mathbf{M}$.

Denote $\text{cone}(\mathbf{C}^T)^*$ as the polyhedral cone parametrized by $\mathbf{C}$,

$$\text{cone}(\mathbf{C}^T)^* = \{\mathbf{x} | \mathbf{C}\mathbf{x} \geq 0\},$$

and $\mathcal{K}$ as the second-order cone [3],

$$\mathcal{K} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{1}^T\mathbf{x} \geq \|\mathbf{x}\|_2\},$$

our proposed assumption on $\mathbf{C}$ is the following.

ASSUMPTION 1. *The non-negative matrix $\mathbf{C}$ satisfies the following conditions:*

1. *cone$(\mathbf{C}^T)^* \subseteq \mathcal{K}$;*
2. *cone$(\mathbf{C}^T)^* \cap \mathbf{bd}\mathcal{K} = \{\lambda\mathbf{e}_i | \lambda \geq 0, i = 1, 2, \ldots, n\}$.*

In Assumption 1, $\mathbf{bd}\mathcal{K}$ means the boundary of $\mathcal{K}$, i.e., $\mathbf{bd}\mathcal{K} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{1}^T\mathbf{x} = \|\mathbf{x}\|_2\}$, and $\mathbf{e}_i$ is the all 0 vector except for the i-th element being 1. An interpretation of the second requirement of Assumption 1 is the following: obviously the first requirement means every element in $\text{cone}(\mathbf{C}^T)^*$ is also contained in $\mathcal{K}$, and the second requirement further constrains the elements of $\text{cone}(\mathbf{C}^T)^*$ to be contained in the interior of $\mathcal{K}$, except for the $\mathbf{e}_i$'s and their positively scaled versions. In other words, if $\mathbf{C}$ satisfies Assumption 1, and there is a point $\mathbf{x}$ such that $\mathbf{x} \in \text{cone}(\mathbf{C}^T)^*$ and $\mathbf{1}^T\mathbf{x} = \|\mathbf{x}\|_2$, then $\mathbf{x} = \lambda\mathbf{e}_i$.

THEOREM 2.1. *Suppose $\mathbf{C}$ is non-negative ($\mathbf{C} \geq 0$) and each column sums up to 1 ($\mathbf{C}^T\mathbf{1} = 1$), and we are given a transformed version of it, $\hat{\mathbf{C}}$. If $\mathbf{C}$ satisfies Assumption 1, then by solving (2.2) optimally, we can recover $\mathbf{C}$ up to permutation of its columns.*

Proof. Cf. Appendix A.

This condition that we imposed on $\mathbf{C}$ was first proposed in a different context in [6], and used to prove uniqueness of non-negative matrix factorization. It was soon thereafter extended to the minimum volume enclosing simplicial cone problem in [4], under a similar setting (they assumed, using our notation, that the *rows* of $\mathbf{C}$ sum up to 1). As it turns out, the same type of result holds in our present setup, with an even simpler proof.

It is shown in [6] that if $\mathbf{C}$ satisfies Assumption 1, then each column of $\mathbf{C}$ contains at least $n-1$ zeros. However, checking the condition exactly is NP-hard. In Appendix B, we propose a method to approximately check Assumption 1, and show empirically that sparse, non-negative tall $\mathbf{C}$ satisfies Assumption 1 with high probability.

3 Proposed Method

Theoretically, if $\mathbf{C}$ is sparse, non-negative and tall, in which case Assumption 1 is satisfied with high probability, it is enough to only work on $\hat{\mathbf{C}}$ to figure out the true similarity transformation, thus successfully identifying the true $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, up to permutation of the states. We will first introduce an algorithm that approximately solves (2.2), under a noiseless scenario. In practice, when the measurements are noisy, we found that the aforementioned formulation is very sensitive to noise. We therefore use a modified robust formulation, followed by a least-squares refinement procedure to make the resulting $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}$ sparse (and non-negative, if/as appropriate).

3.1 Algorithm for (2.2) and a robust formulation The absolute value of the determinant of a non-symmetric matrix is proportional to the volume of a simplex defined by the columns of that matrix (and the origin) [10]. The objective function of (2.2) is non-convex, therefore (2.2) is presumably hard to solve. Two approaches that can be used to handle this type of non-convexity are successive linearization, and block coordinate descent—see [10] and the references therein. In our experiments we found that the block coordinate descent method works better in the noiseless case, therefore this method is briefly explained next.

If we fix all columns of $\mathbf{M}$ but one, the objective is linear over that column,

$$\det \mathbf{M} = \sum_{i=1}^{n} (-1)^{i+j} \mathbf{m}_j(i) \det \mathbf{M}_{i,j},$$

where $\mathbf{m}_j(i)$ means the i-th entry of the j-th column of $\mathbf{M}$, and $\mathbf{M}_{i,j}$ is obtained by deleting the i-th row and j-th column of $\mathbf{M}$. Thus, the update of one column of $\mathbf{M}$ becomes

$$\begin{aligned} &\underset{\mathbf{m}_j}{\text{maximize}} && \left| \sum_{i=1}^{n} (-1)^{i+j} \mathbf{m}_j(i) \det \mathbf{M}_{i,j} \right|, \\ &\text{subject to} && \hat{\mathbf{C}}\mathbf{m}_j \geq 0, \quad \mathbf{m}_j^T\mathbf{1} = \mathbf{1}. \end{aligned} \tag{3.3}$$

Now (3.3) is still non-convex, but we can get rid of the absolute value and solve two linear programming problems instead (maximizing and minimizing the linear objective function), then set $\mathbf{m}_j$ as the one that gives the larger absolute value.

If $\hat{\mathbf{C}}$ is not exactly a transformed version of $\mathbf{C}$, but includes some noise due to subspace estimation errors, one potential problem with formulation (2.2) is that it may not even be feasible. As a trade-off between the maximum determinant criterion and non-negativity of $\mathbf{C}$, we can use instead

$$\begin{aligned} &\underset{\mathbf{M}}{\text{maximize}} \quad \log|\det \mathbf{M}| - \lambda \sum_{i,j} [\hat{\mathbf{C}}\mathbf{M}]_{-}, \\ &\text{subject to} \quad \mathbf{M}^T\mathbf{1} = \mathbf{1}, \end{aligned} \tag{3.4}$$

where $[\cdot]_-$ sums the negative elements of its argument. We put a log to the determinant term because we found that otherwise, in order to make the second term large, the algorithm tends to make $\mathbf{M}$ singular, even if the regularization parameter λ is very small. By taking the log of the determinant term, it will decrease the objective function sharply when $\mathbf{M}$ is close to singular, while increasing it slowly when it is not. Algorithmically, we can still update $\mathbf{M}$ column by column, since the sub-problem can still be cast as two convex optimization problems.

3.2 Sparse refinement While our ultimate goal is to estimate $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, our identifiability results show that a tall non-negative and sparse of $\mathbf{C}$ can be enough to guarantee identifiability, in the noiseless case. In practice, when the noiseless scenario is not realistic, what we observe is that while the robust formulation (3.4) is able to recover $\mathbf{C}$ with minor errors, the resulting $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ are not close to $\mathbf{A}$ and $\mathbf{B}$, even when the perturbation in the data is quite small. We therefore propose to refine the result from (3.4) by solving the problem given in (3.5), which also takes into account possible sparsity in $\mathbf{A}$ and $\mathbf{B}$. In (3.5), the operation $\|\cdot\|_0$ returns the cardinality of the argument. Notice that we have introduced an auxiliary variable $\mathbf{M}_{\text{inv}}$ and a penalty term $\lambda\|\mathbf{M}_{\text{inv}}\mathbf{M}-\mathbf{I}\|_F^2$ to make it close to $\mathbf{M}^{-1}$, instead of working directly with both $\mathbf{M}$ and $\mathbf{M}^{-1}$, which is presumably hard.

This formulation is still non-convex; but it is amenable to block coordinate descent—the updates for $\mathbf{M}$ and $\mathbf{M}_{\text{inv}}$ are classical linear least-squares, whereas the updates for $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}$ are simple projections. Although the cardinality constraints are not convex, the corresponding projections are very easy—we only need to keep the entries with largest (absolute) values and zero out the others. Note that there is no reason to use an l_1 norm surrogate of the cardinality constraints, as hard projection is in fact easier here, and the $\mathbf{M}_{\text{inv}}\mathbf{M}$ term makes the problem non-convex, regardless. Note that, due to non-convexity, the block coordinate descent algorithm can get stuck at a local minimum. This is why initializing it with the result of (3.4) is crucial.

3.3 Summary of the proposed method The method proposed for principled neuro-functional connectivity discovery (NFCD) is summarized in Alg. 1, which consists of three steps: i) A system identification step to get $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, and $\hat{\mathbf{C}}$ from the input-output data $\mathbf{U}$ and $\mathbf{Y}$, and a given system order n; ii) a determinant maximization step as discussed in §3.1; and iii) a sparse refinement step as discussed in §3.2.

In the system identification step, an interesting observation is that our model has more outputs than states, unlike typical LDS models in automatic control where the number of outputs is smaller than the number of states. Having more outputs than states makes system identification easier. As described in the first step of Alg. 1, we only need to take the 'thin' SVD of the output samples, and then solve a linear least-squares problem. In line 3, $\mathbf{T}$ is a random $n \times n$ matrix—we found by simulations that this improves the conditioning. In line 4, $\hat{\mathbf{X}}_0$ is the first $N-1$ columns of $\hat{\mathbf{X}}$, and $\hat{\mathbf{X}}_1$ is the last $N-1$ columns of $\hat{\mathbf{X}}$, where N is the number of samples, i.e., the number of columns of $\hat{\mathbf{X}}$.

In the 3rd step of Alg.1, we used the (hard) thresholding operator $\mathcal{T}_t(\cdot)$ parameterized by t, and its non-negative version $\mathcal{T}_t^+(\cdot)$, which are defined as:

$$\mathcal{T}_t(z) = \begin{cases} z \text{ , if } |z| \geq t \\ 0 \text{ , else} \end{cases}, \quad \mathcal{T}_t^+(z) = \begin{cases} z \text{ , if } z \geq t \\ 0 \text{ , else} \end{cases}.$$

A brief discussion on the complexity of NFCD is useful at this point. For the MEG data that we consider in this paper, m (the input dimension) and p (the number of MEG sensors) are no more than a few hundreds, and, as shown in [13], for n ranging from 10 to 30 the LDS model is able to capture most of the brain dynamics. Therefore, steps 2 and 3 of NFCD are relatively small scaled. The only number that can possibly go large is N, the number of samples collected by the MEG sensors, which only comes into play in step 1. Notice that for both SVD and least-squares, complexity grows linearly in the large dimension (times the small dimension squared). Thus, even if N is very large, NFCD is able to scale well.

$$
\begin{aligned}
&\underset{\substack{\tilde{\mathbf{A}},\tilde{\mathbf{B}},\tilde{\mathbf{C}},\\ \mathbf{M},\mathbf{M}_{\text{inv}}}}{\text{minimize}} \quad \|\tilde{\mathbf{A}} - \mathbf{M}_{\text{inv}}\hat{\mathbf{A}}\mathbf{M}\|_F^2 + \|\tilde{\mathbf{B}} - \mathbf{M}_{\text{inv}}\hat{\mathbf{B}}\|_F^2 + \|\tilde{\mathbf{C}} - \hat{\mathbf{C}}\mathbf{M}\|_F^2, + \lambda\|\mathbf{M}_{\text{inv}}\mathbf{M} - \mathbf{I}\|_F^2 \\
&\text{subject to} \quad \|\tilde{\mathbf{A}}\|_0 \le s_A, \quad \|\tilde{\mathbf{B}}\|_0 \le s_B, \quad \|\tilde{\mathbf{C}}\|_0 \le s_C, \quad \tilde{\mathbf{C}} \ge 0,
\end{aligned} \tag{3.5}
$$

Algorithm 1 Neuro-Functional Connectivity Discovery (NFCD)

procedure SystemIdentification($\mathbf{U}$, $\mathbf{Y}$, n)
  $\mathbf{Y} \approx \mathbf{U}_n\mathbf{\Sigma}_n\mathbf{V}_n^T$
  $\hat{\mathbf{C}} \leftarrow \mathbf{U}_n\mathbf{T}^{-1}, \quad \hat{\mathbf{X}} \leftarrow \mathbf{T}\mathbf{\Sigma}_n\mathbf{V}_n^T$
  $\begin{bmatrix}\hat{\mathbf{A}} & \hat{\mathbf{B}}\end{bmatrix} = \hat{\mathbf{X}}_1 \begin{bmatrix}\hat{\mathbf{X}}_0 \\ \mathbf{U}\end{bmatrix}^\dagger$
  Scale the columns of $\hat{\mathbf{C}}$ to sum up to 1, and then counter-scale $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ accordingly
end procedure

procedure Solve Problem (3.4)
  initialize $\mathbf{M}$ as a random matrix
  repeat
    for $j = 1, \ldots, n$ **do**
      Solve (3.4) with respect to $\mathbf{m}_j$.
    end for
  until convergence
end procedure

procedure Solve Problem (3.5)
  initialize $\mathbf{M}$ as the result obtained from the previous step, and $\mathbf{M}_{\text{inv}}$ as its inverse
  repeat
    $t \leftarrow$ the s_A-th largest value in $|\mathbf{M}_{\text{inv}}\hat{\mathbf{A}}\mathbf{M}|$
    $\tilde{\mathbf{A}} \leftarrow \mathcal{T}_t(\mathbf{M}_{\text{inv}}\hat{\mathbf{A}}\mathbf{M})$,
    $t \leftarrow$ the s_B-th largest value in $|\mathbf{M}_{\text{inv}}\hat{\mathbf{B}}|$
    $\tilde{\mathbf{B}} \leftarrow \mathcal{T}_t(\mathbf{M}_{\text{inv}}\hat{\mathbf{B}})$,
    $t \leftarrow \max(0,$the s_C-th largest value in $\hat{\mathbf{C}}\mathbf{M})$
    $\tilde{\mathbf{C}} \leftarrow \mathcal{T}_t^+(\hat{\mathbf{C}}\mathbf{M})$,
    $\mathbf{M}_{\text{inv}} \leftarrow \begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}} & \lambda\mathbf{I}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{A}}\mathbf{M} & \hat{\mathbf{B}} & \lambda\mathbf{M}\end{bmatrix}^\dagger$
    $\mathbf{M} \leftarrow \begin{bmatrix}\mathbf{M}_{\text{inv}}\hat{\mathbf{A}} \\ \hat{\mathbf{C}} \\ \lambda\mathbf{M}_{\text{inv}}\end{bmatrix}^\dagger \begin{bmatrix}\tilde{\mathbf{A}} \\ \tilde{\mathbf{C}} \\ \lambda\mathbf{I}\end{bmatrix}$
  until convergence
end procedure

4 Experiments

We next present some numerical results to corroborate our theoretical claims and illustrate the robustness of our methods. The convex optimization subproblems are solved by using CVX, a package for specifying and solving convex programs [5].

4.1 Synthetic data We start by experimenting with synthetically generated data, where we know $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and we can check whether our proposed method is able to recover them from input-output data. We begin by assuming that the system is noiseless, and simply use (2.2) without refinement. The true $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are generated randomly, and whether an entry is zero or not is determined by drawing from an i.i.d. Bernoulli distribution. The non-zeros entries of $\mathbf{A}$ and $\mathbf{B}$ are drawn from an i.i.d. Gaussian distribution, whereas the non-zeros of $\mathbf{C}$ are drawn from an i.i.d. exponential distribution, to ensure non-negativity of $\mathbf{C}$. Then $\mathbf{A}$ is scaled down by its spectral radius to ensure stability of the system, and the columns of $\mathbf{C}$ are scaled to sum up to 1. The inputs $\mathbf{u}(t), t = 1, \ldots, N$, as well as the initial state $\mathbf{x}(0)$, are generated from an i.i.d. Gaussian distribution. Then the inputs are sent into the system in (1.1) to obtain the outputs $\mathbf{y}(t), t = 1, \ldots, N$.

As one particular example, with $n = 30$, $m = 50$, $p = 300$, and approximately 50% of the entries of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ being zero, 10^4 input-output pairs are used to do subspace system identification, and then the estimated $\hat{\mathbf{C}}$ is fed to (2.2). The convergence of the proposed block coordinate descent method is shown in Figure 2. Notice that the horizontal axis starts at 30 because $\mathbf{M}$ becomes feasible and non-singular only after the first round of column updates, therefore it is meaningless to show the objective before 30.

As shown in Figure 2, the algorithm converges very fast. In fact, considering that the first round of column updates tries to find a feasible $\mathbf{M}$, it converges even before the second round of column updates finishes. Let $\mathbf{M}_*$ be the result obtained from solving (2.2); in this noiseless case we simply set

$$\tilde{\mathbf{A}} = \mathbf{M}_*^{-1}\hat{\mathbf{A}}\mathbf{M}_*, \quad \tilde{\mathbf{B}} = \mathbf{M}_*^{-1}\hat{\mathbf{B}}, \quad \tilde{\mathbf{C}} = \hat{\mathbf{C}}\mathbf{M}_*.$$

Before we compare $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}$ with the ground

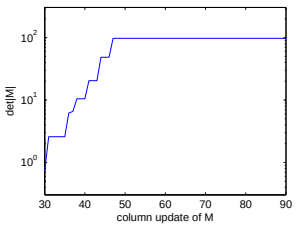

Figure 2: Column update of $\mathbf{M}$ for approximately solving (2.2).

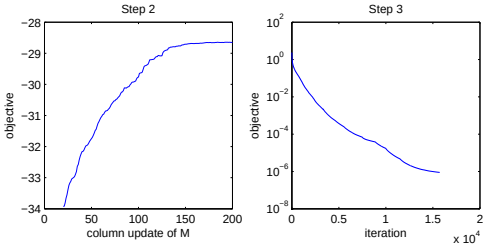

Figure 3: Convergence of Algorithm 1, step 2 (left) and step 3 (right).

Table 1: System matrices recovery in the noiseless case.

	$n = 30$ $s = 0.5$	$n = 30$ $s = 0.3$	$n = 15$ $s = 0.5$
$\frac{\|\mathbf{A}-\tilde{\mathbf{A}}\|_F}{\|\mathbf{A}\|_F}$	7.33e-07	5.85e-06	1.03e-05
$\frac{\|\mathbf{B}-\tilde{\mathbf{B}}\|_F}{\|\mathbf{B}\|_F}$	7.11e-07	5.49e-06	1.50e-05
$\frac{\|\mathbf{C}-\tilde{\mathbf{C}}\|_F}{\|\mathbf{C}\|_F}$	5.93e-07	5.14e-06	1.30e-05

Table 2: System matrices recovery of the C. elegans system with noisy data.

	$n = 10$	$n = 15$	$n = 20$
$\frac{\|\mathbf{A}-\tilde{\mathbf{A}}\|_F}{\|\mathbf{A}\|_F}$	3.71e-04	0.0650	0.0299
$\frac{\|\mathbf{B}-\tilde{\mathbf{B}}\|_F}{\|\mathbf{B}\|_F}$	4.77e-04	0.0455	0.0153
$\frac{\|\mathbf{C}-\tilde{\mathbf{C}}\|_F}{\|\mathbf{C}\|_F}$	3.73e-04	0.0375	0.0135

truth, we need to be aware that there is still a permutation ambiguity, i.e., the similarity transformation can be a permutation matrix, which does not affect the true structure of the system, but only relabels the states. We resolve this by first matching the columns of $\tilde{\mathbf{C}}$ with $\mathbf{C}$, i.e.,

$$\min_{\mathbf{P}\in\Pi} \|\mathbf{C} - \tilde{\mathbf{C}}\mathbf{P}\|_F^2,$$

where Π indicates the set of permutation matrices. This problem can be cast as a linear assignment problem, which be solved optimally by the Hungarian method [7] [3]. After obtaining the best permutation $\mathbf{P}$, the rows and/or columns of $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}$ are permuted accordingly.

In Table 1, we provide the normalized estimation error of the system matrices for various settings, where s indicates the ratio of nonzero entries. Sometimes the algorithm fails to generate a non-singular matrix, in which case a different initialization is used and the algorithm is run again. For each setting, 10 Monte-Carlo trials are performed, and we only show the largest error. In all cases, $m = 50$, and $p = 300$. As we can see, this simulation justifies the claim in Theorem 2.1 that sparse, non-negative and tall $\mathbf{C}$ yield an identifiable system.

4.2 Semi-synthetic data Next we try noisy data. Instead of synthetically generating the whole system, the $\mathbf{A}$ matrix we use here comes from real data – the neural connectivity of C. elegans[4]. Specifically, we take the connectivity of the first $10 \sim 20$ C. elegans neurons as the matrix $\mathbf{A}$ (those neurons are relatively more densely connected), again scaled down by its spectral radius to ensure stability. Then we synthetically generate $\mathbf{B}$ and $\mathbf{C}$, similar to the previous experiment. The inputs and the initial state of the system are generated as before, but now we introduce state and measurement noise, i.e.,

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{v}(t),$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{w}(t),$$

where $\mathbf{v}(t)$ and $\mathbf{w}(t)$ are white Gaussian, with standard deviation $\sigma = 10^{-3}$. The matrices $\mathbf{B}$ and $\mathbf{C}$ are generated with $m = 50$, $p = 300$, and approximately 50% zeros. Then Algorithm 1 is applied to the input-output data. In step 2, we set $\lambda = 0.5$, and in step 3, we set $\lambda = 100$. The cardinality constraints in step 3 are set to be approximately 10% more than the true density. For $n = 20$, the convergence of these two steps is shown in Figure 3.

Finally, the resulting $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}$ are compared with the true system $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, after column matching using the Hungarian method, and the normalized errors for different values of n are shown in

[3] A MATLAB inplementation of the Hungarian method is used and available at `http://www.mathworks.com/matlabcentral/fileexchange/11609-hungarian-algorithm`

[4] available at `http://www.wormatlas.org/neuronalwiring.html`.

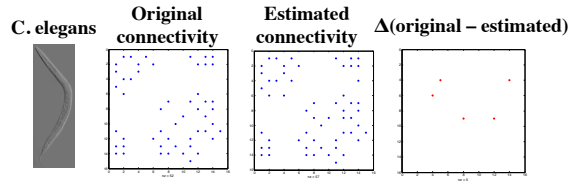

Figure 4: Recovery of C. elegans neural connectivity.

Table 2. Notice that the recovery is almost perfect when $n = 10$. For $n = 15$, the estimated connectivity matrix is compared with the true connectivity in Figure 4, in which case we managed to recover all the true connectivity, with only a few redundant ones. In fact, if we set the sparsity constraint in (3.5) to be the exact one, the connectivity is recovered perfectly.

4.3 Real data Next we apply our proposed method to a set of real input-output data. The experiment was conducted by asking a yes/no question about a particular word to a human subject, and then his/her brain activities are measured by the 306 MEG sensors. Approximately 20 questions were asked for 60 words, and then 340 MEG measurements were collected for each particular question/word. We sampled 10 samples for each experiment, and also added 2 dimensions to indicate the time the subject responded to the question, and the answer given. This forms the output matrix $\mathbf{Y}$, with $p = 308$ and $N \approx 12000$. The input dimension $m = 40$, which is a subset of the 218-questions description of those 20 words conducted by Amazon Mechanical Turks. For more details on the dataset, see [13, 17].

As mentioned earlier, the LDS (Linear Dynamical System) modeling of the brain provides good input-output predictions. However, the ultimate goal is not simply to predict outputs, but also to study the functional connectivity of the brain. As we have argued in Section 1, for MEG sensors the measurement matrix $\mathbf{C}$ is non-negative and sparse, therefore satisfies Assumption 1 with high probability. Using the identifiability results and the algorithm developed in this paper, we can analyze this dataset and see if we obtain interpretable results.

We tried this real input-output data using Algorithm 1 to fit a 15-state LDS, with the same λ values as in the previous simulation, assuming 50% sparsity of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. The regularization parameters in the optimization problem of step 2 and 3 are set equal to the previous simulation for the C. elegans data. The resulting $\tilde{\mathbf{A}}$ is represented as a graph to show the functional connectivity, in Figure 1 in the introduction. As expected, the (hidden) functional connectivity matrix obtained from the MEG experiments is quite sparse, and diagonally dominant. In lieu of ground truth data, we gain confidence in our model because of the fact that under our assumptions, our algorithms are able to recover a sparse functional connectivity matrix which successfully (and in a stable and robust manner) models MEG brain activity in the least squares sense. We omit the corresponding figures due to space restrictions, however, in our experiments we observed robust reconstruction of the MEG recorded brain activity using the obtained model.

5 Conclusions

Our goal is to solve the linear dynamical system (LDS) model of the brain by tackling the subtle, identifiability issue as well as the sparsity and non-negativity constraints, in a principled, effective way. Our contributions are the following:

- *Proof* that our proposed conditions resolve the identifiability issue.
- *Algorithm*: our two-stage algorithm is carefully designed. We give a robust problem reformulation when the data is noisy; and we propose a refinement step to sparsify the connectivity matrix.
- *Validation*, using real and synthetic data. For the semi-synthetic data, we used a subset of the neuro-connectivity of the C. elegans as the system to simulate a set of noisy input-output data, and managed to recover the true neuro-connectivity with high accuracy. On real data measured by MEG, our method produced interpretable results.

Acknowledgments

Research was funded by the National Science Foundation under Grants No. IIS-1247632, IIS-1247489. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

A Proof of Theorem 2.1

We use the following lemma to prove Theorem 2.1.

LEMMA A.1. *Suppose the matrix* $\mathbf{C}$ *satisfies that* $\mathbf{C} \geq 0$, *and* $\mathbf{C}^T\mathbf{1} = \mathbf{1}$. *If it further satisfies Assumption 1, then for any* $\tilde{\mathbf{C}} = \mathbf{CT}$ *that maintains*

non-negativity and the scaling, i.e.,

$$\tilde{\mathbf{C}} \geq 0, \quad \tilde{\mathbf{C}}^T\mathbf{1} = \mathbf{1},$$

we have that $|\det \mathbf{T}| \leq 1$. *Furthermore, equality holds if and only if* $\mathbf{T}$ *is a permutation matrix.*

Proof. First of all, since $\mathbf{C}^T\mathbf{1} = \mathbf{1}$, and $\mathbf{T}^T\mathbf{C}^T\mathbf{1} = \tilde{\mathbf{C}}^T\mathbf{1} = \mathbf{1}$, obviously $\mathbf{T}^T\mathbf{1} = \mathbf{1}$.

The condition $\mathbf{CT} = \tilde{\mathbf{C}} \geq 0$ geometrically means

$$\mathbf{t}_i \in \text{cone}(\mathbf{C})^*, \quad i = 1, \ldots, n,$$

where $\mathbf{t}_i$ is the i-th column of $\mathbf{T}$. Since $\mathbf{C}$ satisfies Assumption 1, i.e., $\text{cone}(\mathbf{C})^* \subseteq \mathcal{K}$, we have

$$\mathbf{t}_i \in \mathcal{K},$$

which means $\mathbf{t}_i$ satisfies that $\mathbf{t}_i^T\mathbf{1} \geq \|\mathbf{t}_i\|_2$. Therefore,

$$|\det \mathbf{T}| \leq \prod_{i=1}^{n} \|\mathbf{t}_i\|_2 \leq \prod_{i=1}^{n} \mathbf{t}_i^T\mathbf{1} = 1, \tag{A.1}$$

where the first inequality is Hadamard's inequality. From the discussion about the second requirement of Assumption 1, the second inequality of (A.1) holds as an equality if and only if the $\mathbf{t}_i$'s are all standard vectors (because they have to sum up to one, the scalings are all 1); in this case, the first inequality holds as an equality if the $\mathbf{t}_i$'s are all different standard vectors, in other words, when the matrix $\mathbf{T}$ is a permutation matrix. Thus, $|\det \mathbf{T}| = 1$ if and only if $\mathbf{T}$ is a permutation matrix. **QED**

Proof. [Proof of Theorem 2.1] By contradiction. Suppose $\mathbf{M}_*$ is an optimal solution of (2.2), but $\hat{\mathbf{C}}\mathbf{M}_*$ is not a column permutation of $\mathbf{C}$, then at least it is a transformation of $\mathbf{C}$, i.e., there exists a non-singular matrix $\mathbf{G}$ (which is not a permutation matrix) such that $\hat{\mathbf{C}}\mathbf{M}_* = \mathbf{CG}$. Let $\tilde{\mathbf{C}} = \hat{\mathbf{C}}\mathbf{M}_*$, then obviously,

$$\tilde{\mathbf{C}} \geq 0, \quad \tilde{\mathbf{C}}^T\mathbf{1} = \mathbf{1}.$$

Since $\mathbf{C}$ satisfies Assumption 1, according to Lemma A.1, and the fact that $\mathbf{G}$ is not a permutation matrix, we have

$$|\det \mathbf{G}| < 1.$$

Now let $\mathbf{M}_0 = \mathbf{M}_*\mathbf{G}^{-1}$, clearly

$$|\det \mathbf{M}_0| = |\det \mathbf{M}_*||\det \mathbf{G}|^{-1} > |\det \mathbf{M}_*|,$$

and $\mathbf{M}_0$ is feasible for (2.2). However, $\mathbf{M}_0$ has a larger objective value than $\mathbf{M}_*$, which is assumed to be optimal for (2.2). This means the initial statement is a contradiction. Therefore, $\hat{\mathbf{C}}\mathbf{M}_*$ must be a column permutation of $\mathbf{C}$. **QED**

B Sparsity and Assumption 1

We show empirically that sparse non-negative tall matrices satisfy Assumption 1 with very high probability. It is shown in [6] that to check this condition exactly is NP-hard, but here we will show that by using a simple majorization technique, it can be solved locally; and with multiple initializations global optimality can be often attained.

The essence of Assumption 1 is $\text{cone}(\mathbf{C}^T)^* \subseteq \mathcal{K}^*$, and this condition can be checked if we can solve the following non-convex quadratic programming optimally.

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \|\mathbf{x}\|_2^2, \\ \text{subject to} \quad & \mathbf{Cx} \geq 0, \quad \mathbf{x}^T\mathbf{1} = 1. \end{aligned} \tag{B.2}$$

Then $\text{cone}(\mathbf{C}^T)^* \subseteq \mathcal{K}^*$ is true if and only if the optimal value of (B.2) is strictly larger than 1. A simple observation is that since $\mathbf{C}$ is non-negative, the standard vectors $\mathbf{e}_i$ are clearly feasible, and that they lead to the cost equal to 1. In fact, if a feasible point makes the cost equal to 1, then it lies on $\mathbf{bd}\mathcal{K}^*$. Therefore, $\mathbf{C}$ satisfies Assumption 1 if and only if the optimal value of (B.2) is 1, and all the optimal solutions are the set of standard vectors.

Since that particular set containment problem is NP-hard to check, clearly (B.2) is also NP-hard. A heuristic is to iteratively linearize the objective function and solve a linear program, i.e.,

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \mathbf{x}_k^T\mathbf{x}, \\ \text{subject to} \quad & \mathbf{Cx} \geq 0, \quad \mathbf{x}^T\mathbf{1} = 1, \end{aligned} \tag{B.3}$$

where $\mathbf{x}_k$ is obtained from the solution of the previous iteration. Since the first-order Taylor expansion of a convex function is always a lower-bound of that function, we can see that by iteratively solving (B.3), we are actually iteratively maximizing a lower-bound of (B.2), which falls into the majorization-optimization method category.

One very important implication from Assumption 1 is the following.

PROPOSITION B.1. *If the* $p \times n$ *matrix* $\mathbf{C}$ *satisfies Assumption 1, then each column of* $\mathbf{C}$ *contains at least* $n - 1$ *zeros.*

Proof. Cf. [6].

Using Proposition B.1 as a rule of thumb for the sparsity requirement for the matrix, which is not very strict in terms of sparsity, we can generate random sparse non-negative matrices and try to check

Table 3: The percentage of the matrices with $p = 300$ rows that result in a solution with norm larger than 1.

	$n = 20$	$n = 30$	$n = 50$
$s = 0.3$	0%	0%	0%
$s = 0.5$	0%	0%	1%
$s = 0.7$	0%	1%	3%

whether they satisfy Assumption 1 by approximately solving (B.2). Although the method we propose to solve (B.2) is not guaranteed to be optimal, we can try different initializations to ensure that most of the local optima are found.

As a simple example, we randomly generate matrices with $p = 300$ rows, and with various number of columns n and/or ratio of non-zeros s. Similar to the synthetic data generated in §4, whether the entries are zeros follow an i.i.d. Bernoulli distribution, and the non-zeros are drawn from an i.i.d. exponential distribution. For each case 100 random matrices are generated and set as input to the optimization problem (B.2), and then approximately solved by successively solving (B.3) with 100 random initial points. The percentage of the matrices that result in a solution with norm larger than 1 is given in Table 3. As we can see, sparse, non-negative tall matrices satisfies Assumption 1 with very high probability.

References

[1] P. J. Antsaklis and A. N. Michel. *Linear Systems*. Springer, 2006.

[2] B. Chai, D. Walther, D. Beck, and F.-F. Li. Exploring functional connectivities of the human brain using multivariate information analysis. In *Advances in Neural Information Processing Systems 22*, pages 270–278. 2009.

[3] D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in Neural Information Processing Systems 16*, pages 1141–1148, 2003.

[4] X. Fu, W.-K. Ma, K. Huang, and N. D. Sidiropoulos. Blind separation of quasi-stationary sources: exploiting convex geometry in covariance domain. *IEEE Trans. on Signal Processing*, 2014, submitted.

[5] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, Mar. 2014.

[6] K. Huang, N. Sidiropoulos, and A. Swami. Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Trans. on Signal Processing*, 62(1):211–224, Jan 2014.

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

[8] Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1235–1256, 2009.

[9] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, 2nd edition, 1999.

[10] W.-K. Ma, J. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A. Plaza, A. Ambikapathi, and C.-Y. Chi. A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. *IEEE Signal Processing Magazine*, 31(1):67–81, Jan 2014.

[11] M. Mørup, K. Madsen, A. M. Dogonowski, H. Siebner, and L. K. Hansen. Infinite relational modeling of functional connectivity in resting state fMRI. In *Advances in Neural Information Processing Systems 23*, pages 1750–1758. 2010.

[12] S. Muthukumaraswamy. High-frequency brain activity and muscle artifacts in MEG/EEG: A review and recommendations. *Frontiers in Human Neuroscience*, 7(138), 2013.

[13] E. E. Papalexakis, A. Fyshe, N. D. Sidiropoulos, P. P. Talukdar, T. M. Mitchell, and C. Faloutsos. Good-enough brain model: Challenges, algorithms and discoveries in multi-subject experiments. *ACM SIGKDD*, 2014.

[14] D. Pfau, E. A. Pnevmatikakis, and L. Paninski. Robust learning of low-dimensional dynamics from large neural ensembles. In *Advances in Neural Information Processing Systems 26*, pages 2391–2399. Curran Associates, Inc., 2013.

[15] S. Roberts and R. Choudrey. Bayesian independent component analysis with prior constraints: An application in biosignal analysis. In *Deterministic and Statistical Methods in Machine Learning*, pages 159–179. Springer Berlin Heidelberg, 2005.

[16] V. Sakkalis. Review of advanced techniques for the estimation of brain connectivity measured with EEG/MEG. *Computers in Biology and Medicine*, 41(12):1110–1117, 2011. Special Issue on Techniques for Measuring Brain Connectivity.

[17] G. Sudre, D. Pomerleau, M. Palatucci, L. Wehbe, A. Fyshe, R. Salmelin, and T. Mitchell. Tracking neural coding of perceptual and semantic features of concrete nouns. *NeuroImage*, 62(1):451–463, 2012.

[18] S. Turaga, L. Buesing, A. M. Packer, H. Dalgleish, N. Pettit, M. Hausser, and J. Macke. Inferring neural population dynamics from multiple partial recordings of the same neural circuit. In *Advances in Neural Information Processing Systems*, pages 539–547, 2013.