

CLUSTERED GAUSSIAN GRAPHICAL MODEL VIA SYMMETRIC CONVEX CLUSTERING

Tianyi Yao^a and Genevera I. Allen^{b,c}

^aDept. of Statistics, Rice University

^bDept. of Statistics, Computer Science, and Electrical and Computer Engineering, Rice University

^cNeurological Research Institute, Baylor College of Medicine
Houston, TX

ABSTRACT

Knowledge of functional groupings of neurons can shed light on structures of neural circuits and is valuable in many types of neuroimaging studies. However, accurately determining which neurons carry out similar neurological tasks via controlled experiments is both labor-intensive and prohibitively expensive on a large scale. Thus, it is of great interest to cluster neurons that have similar connectivity profiles into functionally coherent groups in a data-driven manner. In this work, we propose the clustered Gaussian graphical model (GGM) and a novel symmetric convex clustering penalty in an unified convex optimization framework for inferring functional clusters among neurons from neural activity data. A parallelizable multi-block Alternating Direction Method of Multipliers (ADMM) algorithm is used to solve the corresponding convex optimization problem. In addition, we establish convergence guarantees for the proposed ADMM algorithm. Experimental results on both synthetic data and real-world neuroscientific data demonstrate the effectiveness of our approach.

Index Terms— Gaussian graphical model, Convex clustering, ADMM, Computational neuroscience

1. INTRODUCTION

In neuroscience, an important goal is to identify which neurons are involved in similar computations and how they are organized into functionally coherent units to carry out specific computational tasks in the brain. Such knowledge of functional organizations of neurons could lead to a better understanding of structures of interconnected neural circuits and thus the operating mechanisms of the brain. Advancement of optical imaging technologies such as calcium imaging has enabled indirect recordings of spiking activity from thousands of neurons simultaneously [1, 2]. Learning the functional organizations of large neuronal populations from such high-dimensional neural activity recording data is a major challenge in computational neuroscience.

Functional connectivity, which is defined as statistical dependence among measurements of neuronal activity in [3], has been widely used to describe functional interactions among measured neuronal populations. Because functional connectivity is not directly observable, numerous techniques such as correlations and partial correlations have been proposed to estimate such functional connectivity from neural recording data (see [3] for a comprehensive review). In this work, we define functional connectivity between each pair of recorded neurons to be their pairwise partial correlation or edges in an undirected GGM in high dimensions. Because the pairwise partial correlation between two neurons takes activities of all the other recorded neurons into account, it captures only direct associations between neurons and discard all indirect associations [3, 4], which makes pairwise partial correlation coefficient a better indicator of functional connectivity than Pearson correlation. Furthermore, because pairwise partial correlation is the same as the corresponding off-diagonal entries of the standardized precision matrix, the functional connectivity graph of all recorded neurons can be represented by the standardized precision matrix or the corresponding undirected GGM [5].

While there is no standardized definition for functional cluster, many neuroscientific studies have found that each neuronal type has its own distinct input-output connectivity patterns [6] and neurons with similar connectivity patterns typically have similar neurological roles and functions [7]. Therefore, in this work, we seek to define functional clusters to be groups of neurons that share functional connectivity patterns. Hence, inferring functionally coherent groups of neurons is equivalent to clustering neurons with similar functional connectivity patterns.

While many techniques have been proposed for uncovering clusters from multivariate data (see [8] for a comprehensive review) as well as for finding community structures in network data (see [9] for a comprehensive review), they are somewhat limited in this application for various reasons. First of all, distance-based clustering techniques such as k-means and hierarchical clustering on pairwise Euclidean distances cluster variables based on the first-moment of the distribution,

GA and TY acknowledge support from NSF DMS-1554821 and NSF NeuroNex-1707400.

whereas functional clusters are defined by functional connectivity patterns, which are characterized by the second-moment of the distribution. Some studies in fMRI have applied hierarchical clustering on empirical partial correlation based dissimilarity matrix to cluster brain regions [10]. However, such approaches are not applicable to high-dimensional neural activity data because the MLE of partial correlation matrix does not exist due to singularity of the empirical covariance matrix. Others have taken a two-step approach where a functional connectivity graph is first estimated and then community detection algorithms are used subsequently to infer clusters [11, 12]. Yet such two-step approaches are highly sensitive to noise as a single erroneously estimated functional connection in the first step could adversely impact the clustering results of the community detection algorithms. Last but not least, some studies have proposed nonparametric Bayesian approaches for estimating the block structures of GGM and clustering variables using a MCMC sampling method [13]. However, such MCMC-based approaches can easily become computationally infeasible on moderate-sized graphs.

In this paper, we make several methodological contributions: (1) We propose the clustered GGM that involves a novel symmetric convex clustering penalty, which allows us to exploit the symmetric structures of a functional connectivity graph for better estimation of functional clusters. (2) We provide a tractable ADMM algorithm with convergence guarantees to fit our clustered GGM method in big-data settings. Because of these contributions, our clustered GGM method enjoys many advantages over existing approaches to inferring functional clusters from neural activity data: (i) With our novel symmetric convex clustering penalty, our method explicitly leverages functional connectivity patterns to cluster neurons into functionally coherent groups. (ii) Because the clustered GGM is formulated in an unified convex optimization framework, our single-step method is more stable and conducive to data-driven model selection.

2. THE CLUSTERED GGM

2.1. Model Setup and Background

Suppose the neural activity recordings of p neurons over n time points are arranged into the data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ and the recording of all p neurons at the i th time point, $\mathbf{X}_i = \{X_{i1}, \dots, X_{ip}\}$, is a random p -vector independently drawn from the same time-invariant p -variate Gaussian distribution $\mathcal{N}(\mathbf{0}_p, \Sigma_{p \times p})$ [4]. We can approximately achieve the assumption of independence by prewhitening the raw time series using appropriate time series models. As noted before, the functional connectivity graph can be represented by the standardized precision matrix $\Theta \succ 0$, where $\Theta_{ij} = -\Sigma_{ij}^{-1} / \sqrt{\Sigma_{ii}^{-1} \Sigma_{jj}^{-1}}$. Hence, estimating functional clusters based on functional connectivity patterns is equivalent to recovering the group structures that form checkerboard patterns in Θ .

2.2. The Symmetric Convex Clustering Penalty

At first glance, designing a penalty function to encourage checkerboard patterns in the estimate of Θ seems straightforward as one might ask whether we can simply apply the convex biclustering fusion penalty [14] to simultaneously force rows and columns of Θ to coalesce to form block structures. However, simply applying such biclustering penalty does not guarantee the same amount of fusion along the rows and columns of Θ and it can easily result in different estimated functional clusters between rows and columns. In fact, any fusion penalty that directly regularizes elements of Θ would lead to asymmetric estimates, thus leading to difficult interpretations. Also recognized by [13], designing a penalty function to force such checkerboard patterns in a GGM in a computationally feasible way is indeed a challenging task.

Our objective is to develop a convex penalty function that allows us to explicitly model functional clusters among neurons based on mutual pairwise functional connectivity patterns and preserve the symmetry of estimated functional connectivity graph as well as neuron cluster assignments. To this end, we build upon the convex fusion penalty [15, 16] and introduce a novel symmetric convex clustering penalty that encourages symmetric checkerboard patterns in the estimated precision matrix.

Consider a $p \times p$ symmetric matrix Θ , the symmetric convex clustering penalty function takes the form

$$P(\Theta) = \sum_{l \in \mathcal{M}} w_l \|\Psi_{l,1} - \Psi_{l,2}\|_2$$

subject to $(\mathbf{Q}_l \Theta \mathbf{R}_l) - \Psi_l = 0, \forall l \in \mathcal{M}$

Here, we index a neuron pair by $l = (i, j)$ with $1 \leq i < j \leq p$ and define the fusion set over the non-zero fusion weights $\mathcal{M} = \{l = (i, j) : w_l > 0\}$. The set of all nonnegative, pairwise fusion weights $\{w_l\}_{1 \leq i < j \leq p}$ can be specified beforehand to incorporate domain knowledge and take auxiliary information (e.g. interneuron distances) into account. Additionally, $\Psi_{l,1}$ (and $\Psi_{l,2}$) denotes the 1st (and 2nd) column of $\Psi_l \in \mathbb{R}^{(p-2) \times 2}$, which can be interpreted as cluster centroid matrix corresponding to the $l = (i, j)$ th neuron pair. For $l = (i, j)$, the rows of $\mathbf{Q}_l \in \mathbb{R}^{(p-2) \times p}$ consist of canonical basis vectors \mathbf{e}_q for $q \in \{1, 2, \dots, p\} \setminus \{i, j\}$ and the columns of $\mathbf{R}_l \in \mathbb{R}^{p \times 2}$ consist of canonical basis vectors \mathbf{e}_r for $r \in \{i, j\}$.

We now discuss the intuition behind the symmetric convex clustering penalty $P(\Theta)$. For any neuron pair $l = (i, j)$ in the fusion set \mathcal{M} , the canonical basis matrices \mathbf{Q}_l and \mathbf{R}_l extracts a portion of Θ such that the 1st (and 2nd) column of $\mathbf{Q}_l \Theta \mathbf{R}_l \in \mathbb{R}^{(p-2) \times 2}$ represents the functional connectivity patterns of neuron i (and neuron j) with all the other recorded neurons. Ψ_l is taken to be a copy of $\mathbf{Q}_l \Theta \mathbf{R}_l$ and the fusion penalty term $\|\Psi_{l,1} - \Psi_{l,2}\|_2$ induces sparsity in the difference between neuron i and j 's respective functional connectivity patterns with all the other recorded neu-

rons, thus encouraging the estimates of $\Psi_{l,1}$ and $\Psi_{l,2}$ to fuse. Neuron i and j are assigned to the same functional cluster if $\hat{\Psi}_{l,1} = \hat{\Psi}_{l,2}$, which means neuron i and j have the same conditional relationships with all the other recorded neurons. All such fusions can be done separately and in parallel for each neuron pair $l \in \mathcal{M}$, and the set of equality constraints $\mathbf{Q}_l \Theta \mathbf{R}_l - \Psi_l = 0, \forall l \in \mathcal{M}$ aggregates all fusion results back to Θ to form symmetric checkerboard patterns denoting functional clusters among neurons.

2.3. The Clustered GGM via Symmetric Convex Clustering

While one can apply the symmetric convex clustering penalty to any loss functions that take a symmetric matrix as input, we specifically apply $P(\Theta)$ to the negative log-likelihood of the multivariate Gaussian distribution to yield the clustered GGM problem.

$$\begin{aligned} & \underset{\Theta \in \mathbb{S}_{++}^p, \{\Psi_l\}}{\text{minimize}} && -\log \det \Theta + \text{trace}(\hat{\Sigma} \Theta) \\ & && + \lambda \sum_{l \in \mathcal{M}} w_l \|\Psi_{l,1} - \Psi_{l,2}\|_2 \\ & \text{subject to} && (\mathbf{Q}_l \Theta \mathbf{R}_l) - \Psi_l = 0, \forall l \in \mathcal{M} \end{aligned} \quad (1)$$

where \mathbb{S}_{++}^p denotes the set of positive definite matrices of size p and $\hat{\Sigma} = \frac{1}{n} \mathbf{X}^T \mathbf{X}$ denotes the empirical covariance matrix (assuming the columns of \mathbf{X} are properly centered). Unlike the GLasso problem [17], our clustered GGM does not aim to produce a sparse graph estimate. Instead, our clustered GGM leads to a graph estimate $\hat{\Theta}$ with block structures that indicate cluster assignments of nodes. In addition, fusing many elements of Θ to the same values, the symmetric convex clustering penalty significantly reduces the effective number of parameters to be estimated, thus making our clustered GGM an attractive choice for high-dimensional settings. The amount of fusion, and hence the number of clusters, is determined by the penalty parameter λ . The optimal λ can be chosen via data-driven model selection techniques such as consensus clustering [18].

2.4. The Clustered GGM Algorithm

We adopt the generalized ADMM framework described in [19, 20] as well as an approach introduced in [15] for convex clustering problems in order to develop a tractable 3-block ADMM algorithm to solve (1). ADMM is an appealing algorithm for this problem because it permits us to decouple the terms in (1) that are challenging to jointly optimize. Specifically, we reformulate (1) by introducing a set of auxiliary variables $\{\delta_l\}$ and rewrite the penalty term in terms of these auxiliary variables.

$$\begin{aligned} & \underset{\Theta \in \mathbb{S}_{++}^p, \{\Psi_l\}, \{\delta_l\}}{\text{minimize}} && -\log \det \Theta + \text{trace}(\hat{\Sigma} \Theta) \\ & && + \lambda \sum_{l \in \mathcal{M}} w_l \|\delta_l\|_2 \\ & \text{subject to} && \mathbf{Q}_l \Theta \mathbf{R}_l - \Psi_l = 0, \forall l \in \mathcal{M} \\ & && \Psi_{l,1} - \Psi_{l,2} - \delta_l = 0, \forall l \in \mathcal{M} \end{aligned} \quad (2)$$

Following from [15, 19, 20], we give Algorithm 1 to solve the clustered GGM problem:

$\nabla_{\Theta} \mathcal{L}(\Theta^{(k,j-1)})$ denotes the gradient of the corresponding augmented Lagrangian in scaled form evaluated at $\Theta^{(k,j-1)}$ and s_j is the stepsize of gradient descent, which can be selected via the Goldstein-Armijo line search procedure. Here, k is the iteration counter for the outer 3-block ADMM updates and j is the iteration counter for the inner gradient descent updates for the Θ subproblem. $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{R}^2$ are canonical basis vectors and $\mathbf{D} \in \mathbb{R}^{(p-2) \times 2(p-2)} = (\mathbf{e}_1 - \mathbf{e}_2)^T \otimes \mathbf{I}_{(p-2)}$ is the directed difference matrix such that $\mathbf{D} \text{vec}(\Psi_l) = \Psi_{l,1} - \Psi_{l,2}$. Convergence of the algorithm is measured by the norm of the primal and dual residuals and the parameters $\rho_1, \rho_2 > 0$ are fixed throughout the algorithm as recommended by [19].

Algorithm 1: ADMM algorithm for the clustered GGM

Input: $\hat{\Sigma}, \lambda \geq 0, \rho_1, \rho_2 > 0$

Initialize: Primal variables to identity matrices and dual variables to zero matrices;

Precompute: $\mathbf{D}, \{w_l\}, \mathcal{M}$;

while not converged do

(i) Update Θ :

while not converged do

$$\Theta^{(k,j)} \leftarrow \Theta^{(k,j-1)} - s_j \nabla_{\Theta} \mathcal{L}(\Theta^{(k,j-1)});$$

end

(ii) Update Ψ_l ($\forall l \in \mathcal{M}$ in parallel):

$$\Psi_l^{(k)} \leftarrow \frac{\rho_1}{\rho_1 + 2\rho_2} [\mathbf{Q}_l \Theta^{(k)} \mathbf{R}_l + \mathbf{U}_l^{(k-1)} + \frac{\rho_2}{\rho_1} (\delta_l^{(k-1)} - \mathbf{z}_l^{(k-1)}) (\mathbf{e}_1 - \mathbf{e}_2)^T] (\mathbf{I}_2 + \frac{\rho_2}{\rho_1} \mathbf{1}\mathbf{1}^T);$$

(iii) Update δ_l ($\forall l \in \mathcal{M}$ in parallel):

$$\delta_l^{(k)} \leftarrow \text{prox}_{\frac{\lambda w_l}{\rho_2} \|\cdot\|_2} (\mathbf{D} \text{vec}(\Psi_l^{(k)}) + \mathbf{z}_l^{(k-1)});$$

(iv) Update \mathbf{U}_l ($\forall l \in \mathcal{M}$ in parallel):

$$\mathbf{U}_l^{(k)} \leftarrow \mathbf{U}_l^{(k-1)} + (\mathbf{Q}_l \Theta^{(k)} \mathbf{R}_l - \Psi_l^{(k)});$$

(v) Update \mathbf{z}_l ($\forall l \in \mathcal{M}$ in parallel):

$$\mathbf{z}_l^{(k)} \leftarrow \mathbf{z}_l^{(k-1)} + (\mathbf{D} \text{vec}(\Psi_l^{(k)}) - \delta_l^{(k)});$$

end

Proposition 1 *Algorithm 1 converges to a global solution to problem (1).*

Proof sketch: We can first recast our 3-block ADMM problem (2) as the general 3-block ADMM formulation described in [21] by re-writing the set of equality constraints in (2) as a linear combination of the three optimization variables:

$\mathbf{A}_1 \text{vec}(\Theta) + \mathbf{A}_2 \text{vec}(\Psi) + \mathbf{A}_3 \text{vec}(\Delta) = \mathbf{0}$,
 where $\Psi = [\Psi_1, \dots, \Psi_{|\mathcal{M}|}]$, $\Delta = [\delta_1, \dots, \delta_{|\mathcal{M}|}]$,
 $\mathbf{A}_1 = \begin{bmatrix} \mathbf{B}_{2g \times p^2} \\ \mathbf{0}_{g \times p^2} \end{bmatrix}$, $\mathbf{A}_2 = \begin{bmatrix} -\mathbf{I}_{2g} \\ \mathbf{H}_{g \times 2g} \end{bmatrix}$, $\mathbf{A}_3 = \begin{bmatrix} \mathbf{0}_{2g \times g} \\ -\mathbf{I}_g \end{bmatrix}$
 with rows of \mathbf{B} containing appropriate canonical basis vectors and \mathbf{H} containing $|\mathcal{M}|$ directed difference matrices \mathbf{D} on its diagonal. For notational simplicity, we use $g = (p - 2)|\mathcal{M}|$. With $\mathbf{A}_1^T \mathbf{A}_3 = \mathbf{0}$, we can show that (2) satisfies the sufficient conditions (Theorem 2.4 in [21]) for the convergence of such 3-block ADMM algorithm.

3. EXPERIMENTS

3.1. Synthetic Data

In this subsection, we evaluate the comparative performance of our clustered GGM method on simulated data sets.

3.1.1. Data Generation

Suppose we have p neurons which form k functional clusters, we simulate a standardized precision matrix Θ with the desired checkerboard patterns reflecting groundtruth functional clusters as follows: first we define the groundtruth cluster membership for the p neurons by creating $\mathbf{Z}_{p \times k} \in \{0, 1\}$ which has exactly one 1 in each row and at least one 1 in each column. We then generate symmetric matrix $\mathbf{B}_{k \times k} \in [-1, 1]$ where B_{ii} denotes the partial correlation between two neurons if both neurons are in the i th cluster and B_{ij} denotes the partial correlation between a neuron from the i th cluster and a neuron from the j th cluster. Specifically, $B_{ii} \stackrel{i.i.d.}{\sim} \text{Unif}([0.6, 0.95])$ and $B_{ij} \stackrel{i.i.d.}{\sim} \text{Unif}([0, 0.55])$. Next, we generate the groundtruth precision matrix $\Theta = \mathbf{Z}\mathbf{B}\mathbf{Z}^T$ and set the diagonal entries of Θ to 1's to ensure positive-definiteness. Finally, we generate the data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ according to $\mathbf{x}_1, \dots, \mathbf{x}_n \stackrel{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \Theta^{-1})$. We consider two simulation scenarios: Scenario I with $n = 110$, and $p = 50$ neurons are randomly divided into $k = 3$ clusters with size 5, 15, and 30, respectively; Scenario II with $n = 200$, and $p = 200$ neurons are randomly divided into $k = 3$ clusters with size 30, 60, and 110, respectively.

3.1.2. Results

We compare our clustered GGM to other popular clustering approaches: 1) k-means; 2) Hierarchical Clustering (HC) with various linkage functions and dissimilarity metrics (Euclidean distance and empirical correlation); 3) Spectral

Table 1. Simulation results averaged over 10 replicates in terms of Rand Index. Best performing methods are boldfaced.

Dataset	Method	Rand Index
Scenario I	Clustered GGM	0.964 (0.068)
	k-means	0.505 (0.003)
	HC Euclidean Ward	0.498 (0.007)
	SC empirical corr	0.511 (0.006)
	HC empirical corr Ward	0.521 (0.027)
	GLasso + Louvain	0.556 (0.022)
Scenario II	Clustered GGM	0.999 (0.003)
	k-means	0.515 (0.01)
	HC Euclidean Ward	0.791 (0.003)
	SC empirical corr	0.526 (0.002)
	HC empirical corr Ward	0.513 (0.023)
	GLasso + Louvain	0.566 (0.003)

Clustering (SC) with various similarity metrics (empirical correlation and various kernel functions), implemented using R packages `anocva` and `kernlab`; 4) GLasso followed by commonly used community detection algorithms such as the Louvain method [22], implemented using R packages `huge` and `igraph`. The best penalty parameter for the GLasso is selected by the `ebic` criterion embedded in `huge`. Moreover, the oracle number of functional clusters $k = 3$ is supplied to all aforementioned clustering techniques. Such practice is reasonable in the neuroscientific context because the number of functional clusters is typically known *a priori* from domain knowledge.

In Table 1, results on functional cluster recovery are presented. In particular, the performance in terms of functional cluster recovery is quantified using Rand Index which measures the agreement between the unsupervised clustering solutions and the true cluster membership. Rand Index takes values between 0 and 1 with 1 indicating perfect cluster recovery. We only include the best performing approaches from each category 2), 3), and 4) in Table 1. Results in Table 1 reveal that our clustered GGM outperforms all competing approaches in terms of functional cluster recovery.

3.2. Case Study: Calcium Imaging Data

We test our method on a publicly available calcium imaging data set from [23, 24]. Neural activity of a subset of excitatory neurons in mouse visual cortex was recorded using multi-plane acquisition and 10 to 12 planes of different depth were recorded at the same time at a sampling rate of about 3Hz (see [23] for detailed data acquisition and processing procedures). During the course of experiments, 32 natural images were shown to an awake mouse sequentially

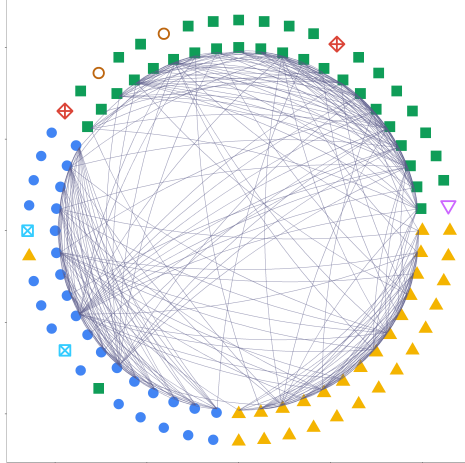


Fig. 1. Comparison of empirically determined neuron tuning labels (inner circle) and functional cluster labels estimated by the clustered GGM (outer circle). Nodes on the inner circle are colored according to neuron tuning labels whereas nodes on the outer circle are colored according to functional cluster labels estimated by the clustered GGM. The Rand Index between neuron tuning labels and functional cluster labels estimated by the clustered GGM is 0.868.

and averaged responses of each recorded neuron to the visual stimuli were determined after adjusting for trial-to-trial variability via model-based approaches. Each neuron is said to be tuned to the natural image to which it had the largest averaged responses and was subsequently assigned a neuron tuning label. Such neuron tuning labels are often used as estimates of functional clusters. However, such empirically inferred neuron tuning labels are likely to be noisy and there could be considerable amount of uncertainty associated with functional groups determined solely by such tuning labels. In this case study, we seek to evaluate how well the noisy neuron tuning labels serve as proxies for identifying functional clusters of neurons in mouse visual cortex.

We select 52 excitatory neurons residing in the most superficial imaging plane that were empirically determined to be tuned to three most dissimilar natural images. The calcium imaging data come in the form of deconvolved calcium traces, whose distributions are highly skewed. To accommodate our model assumptions of independence and Gaussianity, we prewhiten individual calcium traces with an autoregressive model of order 1 to remove temporal dependence and subsequently perform the semiparametric copula transformation [25] to make the data approximately follow a multivariate Gaussian distribution. Afterwards, we apply our clustered GGM to the processed traces of these 52 neurons across 855 time points at stimulus onset. Specifically, we fit the clustered GGM to the data on a fine grid of penalty parameter values $\lambda \in [0, 1.92]$ such that all neurons are clustered into one group for $\lambda \geq 1.92$. The best penalty parameter value selected is $\lambda = 1.06$ and the corresponding estimated functional clusters are displayed in the right panel of Fig. 1.

In Fig. 1, each node denotes a neuron and edges represent the functional connectivity graph. Nodes on the inner

circle are colored according to the noisy neuron tuning labels whereas nodes on the outer circle are colored based upon estimated functional cluster labels by our clustered GGM. The Rand Index between neuron tuning labels and functional cluster labels estimated by our clustered GGM is 0.868. Such results show that the functional clusters estimated by our clustered GGM largely agree with the empirically determined neuron tuning labels except for a handful of singletons, suggesting that neuron tuning labels serve as good proxies for identifying functional clusters of neurons in mouse visual cortex.

4. CONCLUSIONS

In this paper, we have introduced the clustered GGM via symmetric convex clustering in an unified convex optimization framework, which can be used to infer functional clusters among neurons from neural activity recordings. Key contributions include developing a novel symmetric convex clustering penalty to explicitly group neurons with similar functional connectivity patterns as well as providing a tractable algorithm to solve the clustered GGM problem with notable convergence guarantees. Experimental results on both synthetic data and real-world neuroscientific data demonstrate the effectiveness of our proposed method.

Even though the focus of this paper has been on the clustered GGM problem, our novel symmetric convex clustering penalty can be applied to many other convex loss functions that take symmetric matrices as inputs. Such flexibility of our novel penalty function suggests that there is potential for broad application of our approach to data in areas such as genomics and proteomics.

5. REFERENCES

- [1] Philipp Berens, Jeremy Freeman, Thomas Deneux, Nikolay Chenkov, Thomas McColgan, Artur Speiser, Jakob H. Macke, Srinivas C. Turaga, Patrick Mineault, Peter Rupprecht, Stephan Gerhard, Rainer W. Friedrich, Johannes Friedrich, Liam Paninski, Marius Pachitariu, Kenneth D. Harris, Ben Bolte, Timothy A. Machado, Dario Ringach, Jasmine Stone, Luke E. Rogerson, Nicolas J. Sofroniew, Jacob Reimer, Emmanouil Froudarakis, Thomas Euler, Miroslav Romn Rosn, Lucas Theis, Andreas S. Tolias, and Matthias Bethge, “Community-based benchmarking improves spike rate inference from two-photon calcium imaging data,” *PLOS Computational Biology*, vol. 14, no. 5, pp. 1–13, 05 2018, DOI: 10.1371/journal.pcbi.1006157.
- [2] R. James Cotton, Emmanouil Froudarakis, Patrick Storer, Peter Saggau, and Andreas Tolias, “Three-dimensional mapping of microcircuit correlation structure,” *Frontiers in Neural Circuits*, vol. 7, pp. 151, 2013, DOI: 10.3389/fncir.2013.00151.
- [3] Ildefons Magrans de Abril, Junichiro Yoshimoto, and Kenji Doya, “Connectivity inference from neural recording data: Challenges, mathematical bases and research directions,” *Neural Networks*, vol. 102, pp. 120–137, 2018, DOI: 10.1016/j.neunet.2018.02.016.
- [4] Antonio Sutera, Arnaud Joly, Vincent François-Lavet, Zixiao Aaron Qiu, Gilles Louppe, Damien Ernst, and Pierre Geurts, “Simple connectome inference from partial correlation statistics in calcium imaging,” in *Proceedings of the 2014th International Conference on Neural Connectomics - Volume 46*. 2014, pp. 23–35, JMLR.org, URL: <http://proceedings.mlr.press/v46/sutera15.pdf>.
- [5] Steffen L Lauritzen, *Graphical models*, vol. 17, Clarendon Press, 1996.
- [6] Xiaolong Jiang, Shan Shen, Cathryn R. Cadwell, Philipp Berens, Fabian Sinz, Alexander S. Ecker, Saumil Patel, and Andreas S. Tolias, “Principles of connectivity among morphologically defined cell types in adult neocortex,” *Science*, vol. 350, no. 6264, 2015, DOI: 10.1126/science.aac9462.
- [7] Ed Bullmore and Olaf Sporns, “Complex brain networks: graph theoretical analysis of structural and functional systems,” *Nature Reviews Neuroscience*, vol. 10, pp. 186–198, 02 2009, DOI: 10.1038/nrn2575.
- [8] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sept. 1999, DOI: 10.1145/331499.331504.
- [9] Steve Harenberg, Gonzalo Bello, L. Gjeltrema, Stephen Ranshous, Jitendra Harlalka, Ramona Seay, Kanchana Padmanabhan, and Nagiza Samatova, “Community detection in large-scale networks: a survey and empirical evaluation,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 6, no. 6, pp. 426–439, 2014, DOI: 10.1002/wics.1319.
- [10] Raymond Salvador, John Suckling, Martin R. Coleman, John D. Pickard, David Menon, and Ed Bullmore, “Neurophysiological architecture of functional magnetic resonance images of human brain,” *Cerebral Cortex*, vol. 15, no. 9, pp. 1332–1342, 2005, DOI: 10.1093/cercor/bhi016.
- [11] Alnur Ali Wednesday, “Segmenting the brain via sparse inverse covariance estimation and graph-based clustering on high-dimensional fmri data,” 2017.
- [12] Jian Guo and Sijian Wang, “Modularized gaussian graphical model,” Dec 2010, URL: http://www-personal.umich.edu/~guojian/publications/manuscript_mggm.pdf.
- [13] Siqi Sun, Yuancheng Zhu, and Jinbo Xu, “Adaptive variable clustering in gaussian graphical models,” in *AIS-TATS*, 2014, URL: <http://proceedings.mlr.press/v33/sun14.pdf>.
- [14] Eric C. Chi, Genevera I. Allen, and Richard G. Baraniuk, “Convex biclustering,” *Biometrics*, vol. 73, no. 1, pp. 10–19, 2017, DOI: 10.1111/biom.12540.
- [15] Eric C. Chi and Kenneth Lange, “Splitting methods for convex clustering,” *Journal of Computational and Graphical Statistics*, vol. 24, no. 4, pp. 994–1013, 2015, DOI: 10.1080/10618600.2014.948181.
- [16] Toby Dylan Hocking, Armand Joulin, Francis Bach, and Jean-Philippe Vert, “Clusterpath: An algorithm for clustering using convex fusion penalties,” in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011, ICML’11, pp. 745–752.
- [17] Ming Yuan and Yi Lin, “Model selection and estimation in the gaussian graphical model,” *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007, DOI: 10.1093/biomet/asm018.
- [18] Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub, “Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data,” *Machine Learning*, vol. 52, no. 1, pp. 91–118, Jul 2003, DOI: 10.1023/A:1023949509487.

- [19] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011, DOI: 10.1561/22000000016.
- [20] Wei Deng, Ming-Jun Lai, Zhimin Peng, and Wotao Yin, “Parallel multi-block admm with $o(1/k)$ convergence,” *Journal of Scientific Computing*, vol. 71, no. 2, pp. 712–736, May 2017, DOI: 10.1007/s10915-016-0318-2.
- [21] Caihua Chen, Bingsheng He, Yinyu Ye, and Xiaoming Yuan, “The direct extension of admm for multi-block convex minimization problems is not necessarily convergent,” *Mathematical Programming*, vol. 155, no. 1, pp. 57–79, Jan 2016, DOI: 10.1007/s10107-014-0826-5.
- [22] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics*, vol. 2008, no. 10, pp. P10008, oct 2008, DOI: 10.1088/1742-5468/2008/10/p10008.
- [23] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Charu Bai Reddy, Matteo Carandini, and Kenneth D. Harris, “Spontaneous behaviors drive multidimensional, brain-wide population activity,” *bioRxiv*, 2018, DOI: 10.1101/306019.
- [24] Carsen Stringer, Marius Pachitariu, Charu Reddy, Matteo Carandini, and Kenneth D. Harris, “Recordings of ten thousand neurons in visual cortex during spontaneous behaviors,” 5 2018, DOI: <https://doi.org/10.25378/janelia.6163622.v6>.
- [25] Han Liu, John Lafferty, and Larry Wasserman, “The nonparanormal: Semiparametric estimation of high dimensional undirected graphs,” *Journal of Machine Learning Research*, vol. 10, pp. 2295–2328, Dec. 2009, URL: <http://dl.acm.org/citation.cfm?id=1577069.1755863>.

6. DETAILED DERIVATIONS

In this section, we provide derivations of the ADMM algorithm in Algorithm 1 for the clustered GGM. Our notation here is the same as that used in the main body of the paper unless otherwise stated.

After introducing the set of auxiliary variables $\{\delta_l\}$ and rewriting the clustered GGM problem as (2), the augmented Lagrangian in scaled form is given by:

$$\begin{aligned} \mathcal{L}_{\rho_1, \rho_2}(\Theta, \{\Psi_l\}, \{\delta_l\}, \{\mathbf{U}_l\}, \{\mathbf{z}_l\}) &= -\log \det \Theta + \text{trace}(\hat{\Sigma} \Theta) + \lambda \sum_{l \in \mathcal{M}} w_l \|\delta_l\|_2 \\ &\quad + \frac{\rho_1}{2} \sum_{l \in \mathcal{M}} \left(\|\mathbf{Q}_l \Theta \mathbf{R}_l - \Psi_l + \mathbf{U}_l\|_F^2 - \|\mathbf{U}_l\|_F^2 \right) \\ &\quad + \frac{\rho_2}{2} \sum_{l \in \mathcal{M}} \left(\|\mathbf{D} \text{vec}(\Psi_l) - \delta_l + \mathbf{z}_l\|_2^2 - \|\mathbf{z}_l\|_2^2 \right) \end{aligned}$$

Following from [19], the scaled form of the ADMM updates are given by:

$$\begin{aligned} \Theta^{(k)} &= \arg \min_{\Theta \in \mathbb{S}_{++}^p} -\log \det \Theta + \text{trace}(\hat{\Sigma} \Theta) + \frac{\rho_1}{2} \sum_{l \in \mathcal{M}} \|\mathbf{Q}_l \Theta \mathbf{R}_l - \Psi_l^{(k-1)} + \mathbf{U}_l^{(k-1)}\|_F^2 \\ \Psi_l^{(k)} &= \arg \min_{\Psi_l \in \mathbb{R}^{(p-2) \times 2}} \frac{\rho_1}{2} \|\mathbf{Q}_l \Theta^{(k)} \mathbf{R}_l - \Psi_l + \mathbf{U}_l^{(k-1)}\|_F^2 + \frac{\rho_2}{2} \|\mathbf{D} \text{vec}(\Psi_l) - \delta_l^{(k-1)} + \mathbf{z}_l^{(k-1)}\|_2^2, \quad \forall l \in \mathcal{M} \\ \delta_l^{(k)} &= \arg \min_{\delta_l \in \mathbb{R}^{p-2}} \lambda w_l \|\delta_l\|_2 + \frac{\rho_2}{2} \|\mathbf{D} \text{vec}(\Psi_l^{(k)}) - \delta_l + \mathbf{z}_l^{(k-1)}\|_2^2, \quad \forall l \in \mathcal{M} \\ \mathbf{U}_l^{(k)} &= \mathbf{U}_l^{(k-1)} + \mathbf{Q}_l \Theta^{(k)} \mathbf{R}_l - \Psi_l^{(k)}, \quad \forall l \in \mathcal{M} \\ \mathbf{z}_l^{(k)} &= \mathbf{z}_l^{(k-1)} + \mathbf{D} \text{vec}(\Psi_l) - \delta_l, \quad \forall l \in \mathcal{M} \end{aligned}$$

where $\{\mathbf{U}_l\}_{l \in \mathcal{M}}$ and $\{\mathbf{z}_l\}_{l \in \mathcal{M}}$ are the corresponding dual variables. First, we consider the Θ -update:

$$\Theta^{(k)} = \arg \min_{\Theta \in \mathbb{S}_{++}^p} -\log \det \Theta + \text{trace}(\hat{\Sigma} \Theta) + \frac{\rho_1}{2} \sum_{l \in \mathcal{M}} \|\mathbf{Q}_l \Theta \mathbf{R}_l - \Psi_l^{(k-1)} + \mathbf{U}_l^{(k-1)}\|_F^2$$

This is smooth and so we compute the gradient with respect to Θ :

$$\nabla_{\Theta} \mathcal{L} = -\Theta^{-1} + \hat{\Sigma} + \frac{\rho_1}{2} \sum_{l \in \mathcal{M}} (2\mathbf{Q}_l^T \mathbf{Q}_l \Theta \mathbf{R}_l \mathbf{R}_l^T - 2\mathbf{Q}_l^T \Psi_l \mathbf{R}_l^T + 2\mathbf{Q}_l^T \mathbf{U}_l \mathbf{R}_l^T)$$

using the identity ¹

$$\frac{\partial}{\partial \Theta} \|\mathbf{A} \Theta \mathbf{B} + \mathbf{C}\|_F^2 = 2\mathbf{A}^T (\mathbf{A} \Theta \mathbf{B} + \mathbf{C}) \mathbf{B}^T.$$

Then the solution $\Theta^{(k)}$ to the first subproblem can be obtained by applying gradient descent to convergence.

Now we consider the Ψ_l -update:

$$\begin{aligned} \Psi_l^{(k)} &= \arg \min_{\Psi_l \in \mathbb{R}^{(p-2) \times 2}} \frac{\rho_1}{2} \|\mathbf{Q}_l \Theta^{(k)} \mathbf{R}_l - \Psi_l + \mathbf{U}_l^{(k-1)}\|_F^2 + \frac{\rho_2}{2} \|\mathbf{D} \text{vec}(\Psi_l) - \delta_l^{(k-1)} + \mathbf{z}_l^{(k-1)}\|_2^2 \\ &= \arg \min_{\text{vec}(\Psi_l) \in \mathbb{R}^{2(p-2)}} \frac{\rho_1}{2} \|\text{vec}(\mathbf{Q}_l \Theta^{(k)} \mathbf{R}_l) - \text{vec}(\Psi_l) + \text{vec}(\mathbf{U}_l^{(k-1)})\|_2^2 + \frac{\rho_2}{2} \|\mathbf{D} \text{vec}(\Psi_l) - \delta_l^{(k-1)} + \mathbf{z}_l^{(k-1)}\|_2^2 \end{aligned}$$

Since this is fully smooth, we take the gradient with respect to $\text{vec}(\Psi_l)$ to obtain the stationarity conditions:

$$-\rho_1 \left(\text{vec}(\mathbf{Q}_l \Theta^{(k)} \mathbf{R}_l) - \text{vec}(\Psi_l) + \text{vec}(\mathbf{U}_l^{(k-1)}) \right) + \rho_2 \mathbf{D}^T \left(\mathbf{D} \text{vec}(\Psi_l) - \delta_l^{(k-1)} + \mathbf{z}_l^{(k-1)} \right) = \mathbf{0}.$$

Re-arranging the terms, we obtain

$$(\rho_1 \mathbf{I}_{2(p-2)} + \rho_2 \mathbf{D}^T \mathbf{D}) \text{vec}(\Psi_l) = \rho_1 (\text{vec}(\mathbf{Q}_l \Theta^{(k)} \mathbf{R}_l) + \text{vec}(\mathbf{U}_l^{(k-1)})) + \rho_2 \mathbf{D}^T (\delta_l^{(k-1)} - \mathbf{z}_l^{(k-1)}) \quad (3)$$

¹See Equation (119) in the Matrix Cookbook: <https://www.math.uwaterloo.ca/hwolkowi/matrixcookbook.pdf>

Though an analytical solution can be obtained by:

$$\text{vec}(\Psi_l) = (\rho_1 \mathbf{I}_{2(p-2)} + \rho_2 \mathbf{D}^T \mathbf{D})^{-1} [\rho_1 (\text{vec}(\mathbf{Q}_l \Theta^{(k)} \mathbf{R}_l) + \text{vec}(\mathbf{U}_l^{(k-1)})) + \rho_2 \mathbf{D}^T (\boldsymbol{\delta}_l^{(k-1)} - \mathbf{z}_l^{(k-1)})]$$

This update can quickly become computationally expensive as the dimension p grows due to matrix inversion. To avoid such explicit computation of matrix inverse, we exploit the special structure in $(\rho_1 \mathbf{I}_{2(p-2)} + \rho_2 \mathbf{D}^T \mathbf{D})$ and take an approach that parallel those of the ADMM for the completely connected convex clustering problem [15]. By definition, $\mathbf{D} \in \mathbb{R}^{(p-2) \times 2(p-2)} = (\mathbf{e}_1 - \mathbf{e}_2)^T \otimes \mathbf{I}_{(p-2)}$ is the directed difference matrix and $\mathbf{D}^T \mathbf{D}$ can be simplified as follows:

$$\begin{aligned} \mathbf{D}^T \mathbf{D} &= [(\mathbf{e}_1 - \mathbf{e}_2) \otimes \mathbf{I}_{(p-2)}][(\mathbf{e}_1 - \mathbf{e}_2)^T \otimes \mathbf{I}_{(p-2)}] \\ &= [(\mathbf{e}_1 - \mathbf{e}_2)(\mathbf{e}_1 - \mathbf{e}_2)^T] \otimes \mathbf{I}_{(p-2)} \\ &= (2\mathbf{I}_2 - \mathbf{1}\mathbf{1}^T) \otimes \mathbf{I}_{(p-2)} \end{aligned}$$

using the identity $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC} \otimes \mathbf{BD})$ and $(\mathbf{e}_1 - \mathbf{e}_2)(\mathbf{e}_1 - \mathbf{e}_2)^T = 2\mathbf{I}_2 - \mathbf{1}\mathbf{1}^T$. Expanding $(\rho_1 \mathbf{I}_{2(p-2)} + \rho_2 \mathbf{D}^T \mathbf{D})$, we obtain:

$$\begin{aligned} \rho_1 \mathbf{I}_{2(p-2)} + \rho_2 \mathbf{D}^T \mathbf{D} &= \rho_1 [\mathbf{I}_2 \otimes \mathbf{I}_{p-2}] + \rho_2 [(2\mathbf{I}_2 - \mathbf{1}\mathbf{1}^T) \otimes \mathbf{I}_{(p-2)}] \\ &= [\rho_1 \mathbf{I}_2 + \rho_2 (2\mathbf{I}_2 - \mathbf{1}\mathbf{1}^T)] \otimes \mathbf{I}_{(p-1)} \end{aligned}$$

using the identity $\mathbf{A} \otimes \mathbf{C} + \mathbf{B} \otimes \mathbf{C} = (\mathbf{A} + \mathbf{B}) \otimes \mathbf{C}$. Now the LHS of (3) becomes:

$$\begin{aligned} (\rho_1 \mathbf{I}_{2(p-2)} + \rho_2 \mathbf{D}^T \mathbf{D}) \text{vec}(\Psi_l) &= \left([\rho_1 \mathbf{I}_2 + \rho_2 (2\mathbf{I}_2 - \mathbf{1}\mathbf{1}^T)] \otimes \mathbf{I}_{(p-1)} \right) \text{vec}(\Psi_l) \\ &= \text{vec} \left(\Psi_l [\rho_1 \mathbf{I}_2 + \rho_2 (2\mathbf{I}_2 - \mathbf{1}\mathbf{1}^T)] \right) \end{aligned}$$

using the identity $[\mathbf{A}^T \otimes \mathbf{I}] \text{vec}(\mathbf{B}) = \text{vec}(\mathbf{BA})$. Similarly, the RHS of (3) can be re-written as follows:

$$\text{RHS} = \rho_1 \text{vec}(\mathbf{Q}_l \Theta^{(k)} \mathbf{R}_l + \mathbf{U}_l^{(k-1)}) + \rho_2 \text{vec} \left((\boldsymbol{\delta}_l^{(k-1)} - \mathbf{z}_l^{(k-1)}) (\mathbf{e}_1 - \mathbf{e}_2)^T \right)$$

Hence, equation (3) can be re-written as

$$\text{vec} \left(\Psi_l [\rho_1 \mathbf{I}_2 + \rho_2 (2\mathbf{I}_2 - \mathbf{1}\mathbf{1}^T)] \right) = \rho_1 \text{vec}(\mathbf{Q}_l \Theta^{(k)} \mathbf{R}_l + \mathbf{U}_l^{(k-1)}) + \rho_2 \text{vec} \left((\boldsymbol{\delta}_l^{(k-1)} - \mathbf{z}_l^{(k-1)}) (\mathbf{e}_1 - \mathbf{e}_2)^T \right) \quad (4)$$

Un-vectorizing both sides of (4), we obtain:

$$\Psi_l \left[\left(1 + 2 \frac{\rho_2}{\rho_1} \right) \mathbf{I}_2 - \frac{\rho_2}{\rho_1} \mathbf{1}\mathbf{1}^T \right] = \mathbf{Q}_l \Theta^{(k)} \mathbf{R}_l + \mathbf{U}_l^{(k-1)} + \frac{\rho_2}{\rho_1} (\boldsymbol{\delta}_l^{(k-1)} - \mathbf{z}_l^{(k-1)}) (\mathbf{e}_1 - \mathbf{e}_2)^T \quad (5)$$

Applying the Sherman-Morrison formula, we can write the inverse of $[(1 + 2 \frac{\rho_2}{\rho_1}) \mathbf{I}_2 - \frac{\rho_2}{\rho_1} \mathbf{1}\mathbf{1}^T]$ as

$$\left[\left(1 + 2 \frac{\rho_2}{\rho_1} \right) \mathbf{I}_2 - \frac{\rho_2}{\rho_1} \mathbf{1}\mathbf{1}^T \right]^{-1} = \frac{1}{1 + 2 \frac{\rho_2}{\rho_1}} \left(\mathbf{I}_2 + \frac{\rho_2}{\rho_1} \mathbf{1}\mathbf{1}^T \right).$$

Therefore, solving (5) for Ψ_l , we obtain

$$\Psi_l^{(k)} = \frac{1}{1 + 2 \frac{\rho_2}{\rho_1}} \left[\mathbf{Q}_l \Theta^{(k)} \mathbf{R}_l + \mathbf{U}_l^{(k-1)} + \frac{\rho_2}{\rho_1} (\boldsymbol{\delta}_l^{(k-1)} - \mathbf{z}_l^{(k-1)}) (\mathbf{e}_1 - \mathbf{e}_2)^T \right] \left(\mathbf{I}_2 + \frac{\rho_2}{\rho_1} \mathbf{1}\mathbf{1}^T \right).$$

To solve the third subproblem, we note that it can be written as a proximal operator:

$$\begin{aligned} \boldsymbol{\delta}_l^{(k)} &= \arg \min_{\boldsymbol{\delta}_l \in \mathbb{R}^{p-2}} \lambda w_l \|\boldsymbol{\delta}_l\|_2 + \frac{\rho_2}{2} \|\text{Dvec}(\Psi_l^{(k)}) - \boldsymbol{\delta}_l + \mathbf{z}_l^{(k-1)}\|_2^2 \\ &= \arg \min_{\boldsymbol{\delta}_l \in \mathbb{R}^{p-2}} \frac{\lambda w_l}{\rho_2} \|\boldsymbol{\delta}_l\|_2 + \frac{1}{2} \|\boldsymbol{\delta}_l - (\text{Dvec}(\Psi_l^{(k)}) + \mathbf{z}_l^{(k-1)})\|_2^2 \\ &= \text{prox}_{\frac{\lambda w_l}{\rho_2} \|\cdot\|_2} (\text{Dvec}(\Psi_l^{(k)}) + \mathbf{z}_l^{(k-1)}) \end{aligned}$$