Direct Estimation of Differential Functional Graphical Models

Boxin Zhao
Department of Statistics
The Unveristy of Chicago
Chicago, IL 60637
boxinz@uchicago.edu

Y. Samuel Wang
Booth School of Business
The Unveristy of Chicago
Chicago, IL 60637
swang24@uchicago.edu

Mladen Kolar
Booth School of Business
The Unveristy of Chicago
Chicago, IL 60637
mkolar@chicagobooth.edu

Abstract

We consider the problem of estimating the difference between two functional undirected graphical models with shared structures. In many applications, data are naturally regarded as high-dimensional random function vectors rather than multivariate scalars. For example, electroencephalography (EEG) data are more appropriately treated as functions of time. In these problems, not only can the number of functions measured per sample be large, but each function is itself an infinite dimensional object, making estimation of model parameters challenging. We develop a method that directly estimates the difference of graphs, avoiding separate estimation of each graph, and show it is consistent in certain high-dimensional settings. We illustrate finite sample properties of our method through simulation studies. Finally, we apply our method to EEG data to uncover differences in functional brain connectivity between alcoholics and control subjects.

1 Introduction

Undirected graphical models are widely used to compactly represent pairwise conditional independence in complex systems. Let $G = \{V, E\}$ denote an undirected graph where V is the set of vertices with $|V| = p$ and $E \subset V^2$ is the set of edges. For a random vector $X = (X_1, \ldots, X_p)^T$, we say that X satisfies the pairwise Markov property with respect to G if $X_v \not\perp\!\!\!\perp X_w | \{X_u\}_{u \in V \setminus \{v,w\}}$ implies $\{v, w\} \in E$. When X follows a multivariate Gaussian distribution with covariance $\Sigma = \Theta^{-1}$, then $\Theta_{vw} \neq 0$ implies $\{v, w\} \in E$. Thus, recovering the structure of the undirected graph is equivalent to estimating the support of the precision matrix, Θ [10, 13, 4, 24, 25].

We consider a setting where we observe two samples X and Y from (possibly) different distributions, and the primary object of interest is the difference between the conditional dependencies of each population rather than the conditional dependencies in each population. For example, in Section 4.3 we analyze neuroscience data sampled from a control group and a group of alcoholics, and seek to understand how the brain functional connectivity patterns in the alcoholics differ from the control group. Thus, in this paper, the object of interest is the *differential graph*, $G_\Delta = \{V, E_\Delta\}$, which is defined as the difference between the precision matrix of X, Θ^X and the precision matrix of Y, Θ^Y, $\Delta = \Theta^X - \Theta^Y$. When $\Delta_{vw} \neq 0$, it implies $\{v, w\} \in E_\Delta$. This type of differential model has been adopted in [30, 22, 3].

In this paper, we are interested in estimating the differential graph in a more complicated setting. Instead of observing vector valued data, we assume the data are actually random vector valued functions (see [5] for a detailed exposition of random functions). Indeed, we aim to estimate the difference between two functional graphical models and the method we propose combines ideas from graphical models for functional data and direct estimation of differential graphs.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Multivariate observations measured across time can be modeled as arising from distinct, but similar distributions [9]. However, in some cases, it may be more natural to assume the data are measurements of an underlying continuous process [31, 18, 11, 28]. [31, 18] treat data as curves distributed according to a *multivariate Gaussian process* (MGP). [31] shows that Markov properties hold for Gaussian processes, while [18] shows how to consistently estimate underlying conditional independencies.

We adopt the functional data point of view and assume the data are curves distributed according to a MGP. However, we consider two samples from distinct populations with the primary goal of characterizing the difference between the conditional cross-covariance functions of each population. Naively, one could apply the procedure of [18] to each sample, and then directly compare the resulting estimated conditional independence structures. However, this approach would require sparsity in both of the underlying conditional independence graphs and would preclude many practical cases; e.g., neither graph could contain hub-nodes with large degree. We develop a novel procedure that directly learns the difference between the conditional independence structures underlying two MGPs. Under an assumption that the difference is sparse, we can consistently learn the structure of the differential graph, even in the setting where individual graphs are dense and separate estimation would suffer.

Our paper builds on recent literature on graphical models for vector valued data, which suggests that direct estimation of the differences between parameters of underlying distributions may yield better results. [12] considers data arising from pairwise interaction exponential families and propose the *Kullback-Leibler Importance Estimation Procedure* (KLIEP) to explicitly estimate the ratio of densities. [21] uses KLIEP as a first step to directly estimate the difference between two directed graphs. Alternatively, [30, 26] consider two multivariate Gaussian samples, and directly estimate the difference between the two precision matrices. When the difference is sparse, it can be consistently estimated even in the high-dimensional setting with dense underlying precision matrices. [22] extends this approach to Gaussian copula models.

The rest of the paper is organized as follows. In Section 2 we introduce our method for **Fu**nctional **D**ifferential **G**raph **E**stimation (FuDGE). In Section 3 we provide conditions under which FuDGE consistently recovers the true differential graph. Simulations and real data analysis are provided in Section 4[1]. Discussion is provided in Section 5. Appendix contains all the technical proofs and additional simulation results.

We briefly introduce some notation used throughout the rest of the paper. Let $|\cdot|_p$ denote vector p-norm and $\|\cdot\|_p$ denote the matrix/operator p-norm. For example, for a $p \times p$ matrix A with entries a_{jk}, $|A|_1 = \sum_{j,k} |a_{jk}|$, $\|A\|_1 = \max_k \sum_j |a_{jk}|$, $|A|_\infty = \max_{j,k} |a_{jk}|$, and $\|A\|_\infty = \max_j \sum_k |a_{jk}|$. Let $a_n \asymp b_n$ denote that $z_1 \leq \inf_n |a_n/b_n| \leq \sup_n |a_n/b_n| \leq z_2$ for some positive constants z_1 and z_2. Let $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote the minimum and maximum eigenvalues, respectively. For a bivariate function $g(s,t)$, we define the Hilbert-Schmidt norm of $g(s,t)$ (or equivalently, the norm of the integral operator it corresponds to) as $\|g\|_{\text{HS}}^2 = \int\int \{g(s,t)\}^2 dsdt$.

2 Methodology

2.1 Functional differential graphical model

Let $X_i(t) = (X_{i1}(t), \dots, X_{ip}(t))^T$, $i = 1, \dots, n_X$, and $Y_i(t) = (Y_{i1}(t), \dots, Y_{ip}(t))^T$, $i = 1, \dots, n_Y$, be iid p-dimensional *multivariate Gaussian processes* with mean zero and common domain $\mathcal{T}$ from two different, but connected population distributions, where $\mathcal{T}$ is a closed subset of the real line.[2] Also, assume that for $j = 1, \dots, p$, $X_{ij}(t)$ and $Y_{ij}(t)$ are random elements of a separable Hilbert space $\mathbb{H}$. For brevity, we will generally only explicitly define notation for $X_i(t)$; however, the reader should note that all notations for $Y_i(t)$ are defined analogously.

Following [18], we define the conditional cross-covariance function for $X_i(t)$ as

$$C_{jl}^X(s,t) \ = \ \text{Cov}\left(X_{ij}(s), X_{il}(t) \ | \ \{X_{ik}(\cdot)\}_{k \neq j,l}\right). \tag{2.1}$$

If $C_{jl}^X(s,t) = 0$ for all $s, t \in \mathcal{T}$, then the random functions $X_{ij}(t)$ and $X_{il}(t)$ are conditionally independent given the other random functions. The graph $G_X = \{V, E_X\}$ represents the pairwise

[1] The code for this part is on https://github.com/boxinz17/FuDGE

[2] Both $X_i(t)$ and $Y_i(t)$ are indexed by i, but they are not paired observations and are completely independent. Also, we assume mean zero and a common domain $\mathcal{T}$ to simplify the notation, but the methodology and theory generalize to non-zero means and different time domains $\mathcal{T}_X$ and $\mathcal{T}_Y$ when fixing some bijection $\mathcal{T}_X \mapsto \mathcal{T}_Y$.

Markov properties of $X_i(t)$ if

$$E_X \;=\; \{(j,l) \in V^2 :\; j \neq l \;\text{ and }\; \exists \{s,t\} \in \mathcal{T}^2 \text{ such that } C^X_{jl}(s,t) \;\neq\; 0\}. \tag{2.2}$$

In this paper, the object of interest is $C^\Delta(s,t)$ where $C^\Delta_{jl}(s,t) = \; C^X_{jl}(s,t) - C^Y_{jl}(s,t)$. We define the differential graph to be $G_\Delta = \{V, E_\Delta\}$, where

$$E_\Delta \;=\; \{(j,l) \in V^2 :\; j \neq l \;\text{ and }\; \|C^\Delta_{jl}\|_{HS} \;\neq\; 0\}. \tag{2.3}$$

Again, we include an edge between j and l, if the conditional dependence between $X_{ij}(t)$ and $X_{il}(t)$ given all the other curves differs from that of $Y_{ij}(t)$ and $Y_{il}(t)$ given all the other curves.

2.2 Functional principal component analysis

Since $X_i(t)$ and $Y_i(t)$ are infinite dimensional objects, for practical estimation, we reduce the dimensionality using *functional principal component analysis* (FPCA). Similar to the way principal component analysis provides an L_2 optimal lower dimensional representation of vector valued data, FPCA provides an L_2 optimal finite dimensional representation of functional data. As in [18], for simplicity of exposition, we assume that we fully observe the functions $X_i(t)$ and $Y_i(t)$. However, FPCA can also be applied to both densely and sparsely observed functional data, as well as data containing measurement errors. Such an extension is straightforward, cf. [23] and [20] for a recent overview. Let $K^X_{jj}(t,s) = \text{Cov}(X_{ij}(t), X_{ij}(s))$ denote the covariance function for X_{ij}. Then, there exists orthonormal eigenfunctions and eigenvalues $\{\phi_{jk}(t), \lambda^X_{jk}\}_{k\in\mathbb{N}}$ such that for all $k \in \mathbb{N}$ [5]:

$$\int_{\mathcal{T}} K^X_{jj}(s,t)\phi^X_{jk}(t)dt = \lambda^X_{jk}\phi^X_{jk}(s). \tag{2.4}$$

Without loss of generality, assume $\lambda^X_{j1} \geq \lambda^X_{j2} \geq \cdots \geq 0$. By the Karhunen-Loève expansion [5, Theorem 7.3.5], $X_{ij}(t)$ can be expressed as $X_{ij}(t) = \sum_{k=1}^\infty a^X_{ijk}\phi^X_{jk}(t)$ where the principal component scores satisfy $a^X_{ijk} = \int_{\mathcal{T}} X_{ij}(t)\phi^X_{jk}(t)dt$ and $a^X_{ijk} \sim N(0, \lambda^X_{jk})$ with $E(a^X_{ijk}a^X_{ijl}) = 0$ if $k \neq l$. Because the eigenfunctions are orthonormal, the L_2 projection of X_{ij} onto the span of the first M eigenfunctions is

$$X^M_{ij}(t) = \sum_{k=1}^M a^X_{ijk}\phi^X_{jk}(t). \tag{2.5}$$

Functional PCA constructs estimators $\hat\phi^X_{jk}(t)$ and $\hat a^X_{ijk}$ through the following procedure. First, we form an empirical estimate of the covariance function:

$$\hat K^X_{jj}(s,t) = \frac{1}{n_X}\sum_{i=1}^{n_X}(X_{ij}(s) - \bar X_j(s))(X_{ij}(t) - \bar X_j(t)),$$

where $\bar X_j(t) = n_X^{-1}\sum_{i=1}^{n_X} X_{ij}(t)$. An eigen-decomposition of $\hat K^X_{jj}(s,t)$ then directly provides the estimates $\hat\lambda^X_{jk}$ and $\hat\phi^X_{jk}$ which allow for computation of $\hat a^X_{ijk} = \int_{\mathcal{T}} X_{ij}(t)\hat\phi^X_{jk}(t)dt$. Let $a^{X,M}_{ij} = (a^X_{ij1}, \ldots, a^X_{ijM})^T \in \mathbb{R}^M$ and $a^{X,M}_i = ((a^{X,M}_{i1})^T, \ldots, (a^{X,M}_{ip})^T)^T \in \mathbb{R}^{pM}$ with corresponding estimates $\hat a^{X,M}_{ij}$ and $\hat a^{X,M}_i$. Since $X^M_i(t)$ are p-dimensional MGP, $a^{X,M}_i$ will have a multivariate Gaussian distribution with $pM \times pM$ covariance matrix which we denote as $\Sigma^{X,M} = (\Theta^{X,M})^{-1}$. In practice, M can be selected by cross validation as in [18].

For $(j,l) \in V^2$, let $\Theta^{X,M}_{jl}$ be the $M \times M$ matrix corresponding to (j,l)th submatrix of $\Theta^{X,M}$. Let $\Delta^M = \Theta^{X,M} - \Theta^{Y,M}$ be the difference between the precision matrices of the first M principal component scores where Δ^M_{jl} denotes the (j,l)th submatrix of Δ^M. In addition, let

$$E_{\Delta^M} \coloneqq \{(j,l) \in V^2 : j \neq l \;\text{ and }\; \|\Delta^M_{jl}\|_F \neq 0\}, \tag{2.6}$$

denote the set of non-zero blocks of the difference matrix Δ^M. In general $E_{\Delta^M} \neq E_\Delta$; however, we will see that for certain M, by constructing a suitable estimator of Δ^M we can still recover E_Δ.

2.3 Functional differential graph estimation

We now describe our method, FuDGE, for functional differential graph estimation. Let $S^{X,M}$ and $S^{Y,M}$ denote the sample covariances of $\hat{a}_i^{X,M}$ and $\hat{a}_i^{Y,M}$. To estimate Δ^M, we solve the following problem with the group lasso penalty, which promotes blockwise sparsity in $\hat{\Delta}^M$ [27]:

$$\hat{\Delta}^M \in \underset{\Delta\in\mathbb{R}^{pM\times pM}}{\arg\min}\; L(\Delta) + \lambda_n \sum_{\{i,j\}\in V^2} \|\Delta_{ij}\|_F, \tag{2.7}$$

where $L(\Delta) = \mathrm{tr}\left[\frac{1}{2}S^{Y,M}\Delta^T S^{X,M}\Delta - \Delta^T\left(S^{Y,M} - S^{X,M}\right)\right]$. Note that although the true Δ^M is symmetric, we do not enforce symmetry in $\hat{\Delta}^M$.

The design of the loss function $L(\Delta)$ in equation (2.7) is based on [15], where in order to construct a consistent M-estimator, we want the true parameter value Δ^M to minimize the population loss $\mathbb{E}\left[L(\Delta)\right]$. For a differentiable and convex loss function, this is equivalent to selecting L such that $\mathbb{E}\left[\nabla L(\Delta^M)\right] = 0$. Since $\Delta^M = \left(\Sigma^{X,M}\right)^{-1} - \left(\Sigma^{Y,M}\right)^{-1}$, it satisfies $\Sigma^{X,M}\Delta^M\Sigma^{Y,M} - (\Sigma^{Y,M} - \Sigma^{X,M}) = 0$. By this observation, a choice for $\nabla L(\Delta)$ is

$$\nabla L(\Delta^M) = S^{X,M}\Delta^M S^{Y,M} - \left(S^{Y,M} - S^{X,M}\right), \tag{2.8}$$

for which $\mathbb{E}\left[\nabla L(\Delta^M)\right] = \Sigma^{X,M}\Delta^M\Sigma^{Y,M} - (\Sigma^{Y,M} - \Sigma^{X,M}) = 0$. Using properties of the differential of the trace function, this choice of $\nabla L(\Delta)$ yields $L(\Delta)$ in (2.7). The chosen loss is quadratic (see (B.10) in supplement) and leads to an efficient algorithm. Such loss has been used in [22, 26, 14] and [30].

Finally, to form $\hat{E}_\Delta$, we threshold $\hat{\Delta}^M$ by $\epsilon_n > 0$ so that:

$$\hat{E}_\Delta = \{(j,l)\in V^2 : j \neq l \;\text{ and }\; \|\hat{\Delta}^M_{jl}\|_F > \epsilon_n \text{ or } \|\hat{\Delta}^M_{lj}\|_F > \epsilon_n\}. \tag{2.9}$$

2.4 Optimization algorithm for FuDGE

Algorithm 1 Functional differential graph estimation

```
Input: $S^{X,M}, S^{Y,M}, \lambda_n, \eta$.
Output: $\hat{\Delta}^M$.
  Initialize $\Delta^{(0)} = 0_{pM}$.
  repeat
    $A = \Delta - \eta\nabla L(\Delta) = \Delta - \eta\left[S_X^{(M)}\Delta S_Y^{(M)} - (S_Y^{(M)} - S_X^{(M)})\right]$
    for $1 \le i, j \le p$ do
      $\Delta_{jl} \leftarrow \left(\frac{\|A_{jl}\|_F - \lambda_n\eta}{\|A_{jl}\|_F}\right)_+ \cdot A_{jl}$
    end for
  until Converge
```

The optimization problem (2.7) can be solved by a proximal gradient method [17], summarized in Algorithm 1. Specifically, in each iteration step, we update the current value of Δ, denoted as Δ^{old}, by solving the following problem:

$$\Delta^{\text{new}} = \underset{\Delta}{\arg\min}\left(\frac{1}{2}\left\|\Delta - \left(\Delta^{\text{old}} - \eta\nabla L\left(\Delta^{\text{old}}\right)\right)\right\|_F^2 + \eta\cdot\lambda_n\sum_{j,l=1}^{p}\|\Delta_{jl}\|_F\right), \tag{2.10}$$

where $\nabla L(\Delta)$ is defined in (2.8) and η is a user specified step size. Note that $\nabla L(\Delta)$ is Lipschitz continuous with the Lipschitz constant $\|S^{Y,M}\otimes S^{X,M}\|_2 = \lambda_{\max}(S^{Y,M})\lambda_{\max}(S^{X,M})$. Thus, for any η such that $0 < \eta \le 1/\lambda^S_{\max}$, the proximal gradient method is guaranteed to converge [1], where $\lambda^S_{\max} = \lambda_{\max}(S^{Y,M})\lambda_{\max}(S^{X,M})$ is the largest eigenvalue of $S^{X,M}\otimes S^{Y,M}$.

The update in (2.10) has a closed-form solution:

$$\Delta^{\text{new}}_{jl} = \left[\left(\|A^{\text{old}}_{jl}\|_F - \lambda_n\eta\right)/\|A^{\text{old}}_{jl}\|_F\right]_+ \cdot A^{\text{old}}_{jl}, \qquad 1 \le j, l \le p, \tag{2.11}$$

where $A^{\text{old}} = \Delta^{\text{old}} - \eta \nabla L(\Delta^{\text{old}})$ and $x_+ = \max\{0, x\}, x \in \mathbb{R}$ represents the positive part of x. Detailed derivations are given in the appendix.

After performing FPCA, the proximal gradient descent method converges in $O\left(\lambda^S_{\max}/\text{tol}\right)$ iterations, where tol is error tolerance, each iteration takes $O((pM)^3)$ operations. See [19] for convergence analysis of proximal gradient descent algorithm.

3 Theoretical properties

In this section, we present theoretical properties of the proposed method. Again, we state assumptions explicitly for $X_i(t)$, but also require the same conditions on $Y_i(t)$.

Assumption 3.1. *Recall that λ^X_{jk} and $\phi^X_{jk}(t)$ are the the eigenvalues and eigenfunctions of $K^X_{jj}(t)$, the covariance function for $X_{ij}(t)$, and $\lambda^X_{jk} > \lambda^X_{jk'}$ for all $k' > k$.*

(i) Assume $\max_{j \in V} \sum_{k=1}^{\infty} \lambda^X_{jk} < \infty$ and there exists some constant $\beta_X > 1$ such that for each $k \in \mathbb{N}$, $\lambda^X_{jk} \asymp k^{-\beta_X}$ and $d^X_{jk}\lambda^X_{jk} = O(k)$ uniformly in $j \in V$, where $d^X_{jk} = 2\sqrt{2}\max\left\{\left(\lambda^X_{j(k-1)} - \lambda^X_{jk}\right)^{-1}, \left(\lambda^X_{jk} - \lambda^X_{j(k+1)}\right)^{-1}\right\}$.

(ii) Assume for all $k \in \mathbb{N}$, $\phi^X_{jk}(t)$'s are continuous on the compact set $\mathcal{T}$ and satisfy $\max_{j \in V} \sup_{s \in \mathcal{T}} \sup_{k \geq 1} |\phi^X_{jk}(s)| = O(1)$.

The parameter β_X controls the decay rate of the eigenvalues and $d^X_{jk}\lambda^X_{jk} = O(k)$ controls the decay rate of eigen-gaps (see [2] for more details).

To recover the exact functional differential graph structure, we need further assumptions on the difference operator $C^{\Delta} = \{C^X_{jl}(s,t) - C^Y_{jl}(s,t)\}_{j,l \in V}$. Let $\nu = \nu(M) = \max_{(j,l) \in V^2} \left| \|C^{\Delta}_{jl}\|_{\text{HS}} - \|\Delta^M_{jl}\|_F \right|$, and let $\tau = \min_{(j,l) \in E_{\Delta}} \|C^{\Delta}_{jl}\|_{\text{HS}}$, where $\tau > 0$ by the definition in (2.3). Roughly speaking, $\nu(M)$ measures the bias due to using an M-dimensional approximation, and τ measures the strength of signal in the differential graph. A smaller τ implies that the graph is harder to recover, and in Theorem 3.1, we require the bias to be small compared to the signal.

Assumption 3.2. *Assume that $\lim_{M \to \infty} \nu(M) = 0$.*

We also require Assumption 3.3 which assumes sparsity in E_{Δ}. Again, this does not preclude the case where E_X and E_Y are dense, as long as the difference between the two graphs is sparse. This assumption is common in the scalar setting; e.g., Condition 1 in [30].

Assumption 3.3. *There are s edges in the differential graph; i.e., $|E_{\Delta}| = s$.*

Before we give conditions for recovering the differential graph with high probability, we first introduce some additional notation. Let $n = \min\{n_X, n_Y\}$, $\sigma_{\max} = \max\{|\Sigma^{X,M}|_{\infty}, |\Sigma^{Y,M}|_{\infty}\}$, $\beta = \min\{\beta_X, \beta_Y\}$, and $\lambda^*_{\min} = \lambda_{\min}\left(\Sigma^{X,M}\right) \times \lambda_{\min}\left(\Sigma^{Y,M}\right)$. Given positive constant c_1, denote

$$\delta \;=\; (1/\sqrt{c_1}) M^{1+\beta} \sqrt{2\left(\log p + \log M + \log n\right)/n} \tag{3.1}$$

and

$$\Gamma \;=\; \frac{9\lambda_n^2 s}{\kappa_{\mathcal{L}}^2} + \frac{2\lambda_n}{\kappa_{\mathcal{L}}}(\omega_{\mathcal{L}}^2 + 2p^2\nu), \tag{3.2}$$

where

$$\begin{aligned} \lambda_n \;&=\; 2M\left[\left(\delta^2 + 2\delta\sigma_{max}\right)\left|\Delta^M\right|_1 + 2\delta\right], \\ \kappa_{\mathcal{L}} \;&=\; (1/2)\lambda^*_{\min} - 8M^2 s\left(\delta^2 + 2\delta\sigma_{\max}\right), \text{ and} \\ \omega_{\mathcal{L}} \;&=\; 4Mp^2\nu\sqrt{\delta^2 + 2\delta\sigma_{\max}}. \end{aligned} \tag{3.3}$$

Note that Γ implicitly depends on n through λ_n, $\kappa_{\mathcal{L}}$, $\omega_{\mathcal{L}}$ and δ.

Theorem 3.1. *There exist positive constants c_1 and c_2, such that for n and M large enough to simultaneously satisfy*

$$\begin{aligned} &0 < \Gamma < (1/2)\tau - \nu(M) \textit{ and} \\ &\delta < \min\left\{(1/4)\sqrt{\left(\lambda^*_{\min} + 16M^2 s(\sigma_{\max})^2\right)/\left(M^2 s\right)} - \sigma_{\max},\; c_1\right\}, \end{aligned} \tag{3.4}$$

setting $\epsilon_n \in (\Gamma + \nu(M), \tau - (\Gamma + \nu(M)))$ ensures that

$$P\left(\hat{E}_\Delta = E_\Delta\right) \geq 1 - 2c_2/n^2.$$

[18] assumed for some finite M, for all $j \in V$, $\lambda^X_{jm'} = 0$ for all $m' > M$. Under this assumption, $X_{ij}(t) = X^M_{ij}(t)$, and E_X will correspond exactly to $(j,l) \in V^2$ such that $\|\Theta^{X,M}_{jl}\|_F \neq 0$ [18, Lemma 1]. If the same eigenvalue condition holds for $Y_i(t)$, then in our setting $E_\Delta = E_{\Delta^M}$. When this holds and we can fix M, we obtain consistency even in the high-dimensional setting since $\nu = 0$ and $\min\{s\log(pn)|\Delta^M|_1^2/n, s\sqrt{\log(pn)/n}\} \to 0$ implies consistent estimation. However, even with an infinite number of positive eigenvalues, high-dimensional consistency is still possible for quickly decaying ν; e.g, if $\nu = o(p^{-2}M^{-1})$ the same rate is achievable as when $v(M) = 0$.

4 Experiments

4.1 Simulation study

In this section, we demonstrate properties of our method through simulations. In each setting, we generate $n_X \times p$ functional variables from graph G_X via $X_{ij}(t) = b(t)^T\delta^X_{ij}$, where $b(t)$ is a five dimensional basis with disjoint support over $[0,1]$ with

$$b_k(t) = \begin{cases} \cos\left(10\pi\left(x - (2k-1)/10\right)\right) + 1 & (k-1)/5 \leq x < k/5; \\ 0 & \text{otherwise,} \end{cases} \qquad k = 1, \dots, 5.$$

$\delta^X_i = ((\delta^X_{i1})^T, \cdots, (\delta^X_{ip})^T)^T \in \mathbb{R}^{5p}$ follows a multivariate Gaussian distribution with precision matrix Ω^X. $Y_{ij}(t)$ was generated in a similar way with precision matrix Ω^Y. We consider three models with different graph structures, and for each model, data are generated with $n_X = n_Y = 100$ and $p = 30, 60, 90, 120$. We repeat this 30 times for each p and model setting.

Model 1: This model is similar to the setting considered in [30], but modified to the functional case. We generate support of Ω^X according to a graph with $p(p-1)/10$ edges and a power-law degree distribution with an expected power parameter of 2. Although the graph is sparse with only 20% of all possible edges present, the power-law structure mimics certain real-world graphs [16] by creating hub nodes with large degree. For each nonzero block, $\Omega^X_{jl} = \delta' I_5$, where δ' is sampled uniformly from $\pm[0.2, 0.5]$. To ensure positive definiteness, we further scale each off-diagonal block by $1/2, 1/3, 1/4, 1/5$ for $p = 30, 60, 90, 120$ respectively. Each diagonal element of Ω^X is set to 1 and the matrix is symmetrized by averaging it with its transpose. To get Ω^Y, we first select the largest hub nodes in G_X (i.e., the nodes with largest degree), and for each hub node we select the top (by magnitude) 20% of edges. For each selected edge, we set $\Omega^Y_{jl} = \Omega^X_{jl} + W$ where $W_{km} = 0$ for $|k - m| \leq 2$, and $W_{km} = c$ otherwise, where c is generated in the same way as δ'. For all other blocks, $\Omega^Y_{jl} = \Omega^X_{jl}$.

Model 2: We first generate a tridiagonal block matrix Ω^*_X with $\Omega^*_{X,jj} = I_5$, $\Omega^*_{X,j,j+1} = \Omega^*_{X,j+1,j} = 0.6I_5$, and $\Omega^*_{X,j,j+2} = \Omega^*_{X,j+2,j} = 0.4I_5$ for $j = 1, \dots, p$. All other blocks are set to 0. We then set $\Omega^*_{Y,j,j+3} = \Omega^*_{Y,j+3,j} = W$ for $j = 1, 2, 3, 4$, and let $\Omega^*_{Y,jl} = \Omega^*_{X,jl}$ for all other blocks. Thus, we form G_Y by adding four edges to G_X. We let $W_{km} = 0$ when $|k - m| \leq 1$, and $W_{km} = c$ otherwise, with $c = 1/10$ for $p = 30$, $c = 1/15$ for $p = 60$, $c = 1/20$ for $p = 90$, and $c = 1/25$ for $p = 120$. Finally, we set $\Omega^X = \Omega^*_X + \delta I$, $\Omega^Y = \Omega^*_Y + \delta I$, where $\delta = \max\{|\min(\lambda_{\min}(\Omega^*_X), 0)|, |\min(\lambda_{\min}(\Omega^*_Y), 0)|\}$+0.05.

Model 3: We generate Ω^*_X according to an Erdös-Rényi graph. We first set $\Omega^*_{X,jj} = I_5$. With probability .8, we set $\Omega^*_{X,jl} = \Omega^*_{X,lj} = 0.1I_5$, and set it to 0 otherwise. Thus, we expect 80% of all possible edges to be present. Then, we form G_Y by randomly adding s new edges to G_X, where $s = 3$ for $p = 30$, $s = 4$ for $p = 60$, $s = 5$ for $p = 90$, and $s = 6$ for $p = 120$. We set each corresponding block $\Omega^*_{Y,jl} = W$, where $W_{km} = 0$ when $|k - m| \leq 1$ and $W_{km} = c$ otherwise. We let $c = 2/5$ for $p = 30$, $c = 4/15$ for $p = 60$, $c = 1/5$ for $p = 90$, and $c = 4/25$ for $p = 120$. Finally, we set $\Omega^X = \Omega^*_X + \delta I$, $\Omega^Y = \Omega^*_Y + \delta I$, where $\delta > \max\{|\min(\lambda_{\min}(\Omega^*_X), 0)|, |\min(\lambda_{\min}(\Omega^*_Y), 0)|\}$+0.05.

Although the theory assumes fully observed functional data, in order to mimic a realistic setting, we use noisy observations at discrete time points, such that the actual data corresponding to X_{ij} are

$$h^X_{ijk} = X_{ij}(t_k) + e_{ijk}, \quad e_{ijk} \sim N(0, 0.5^2),$$

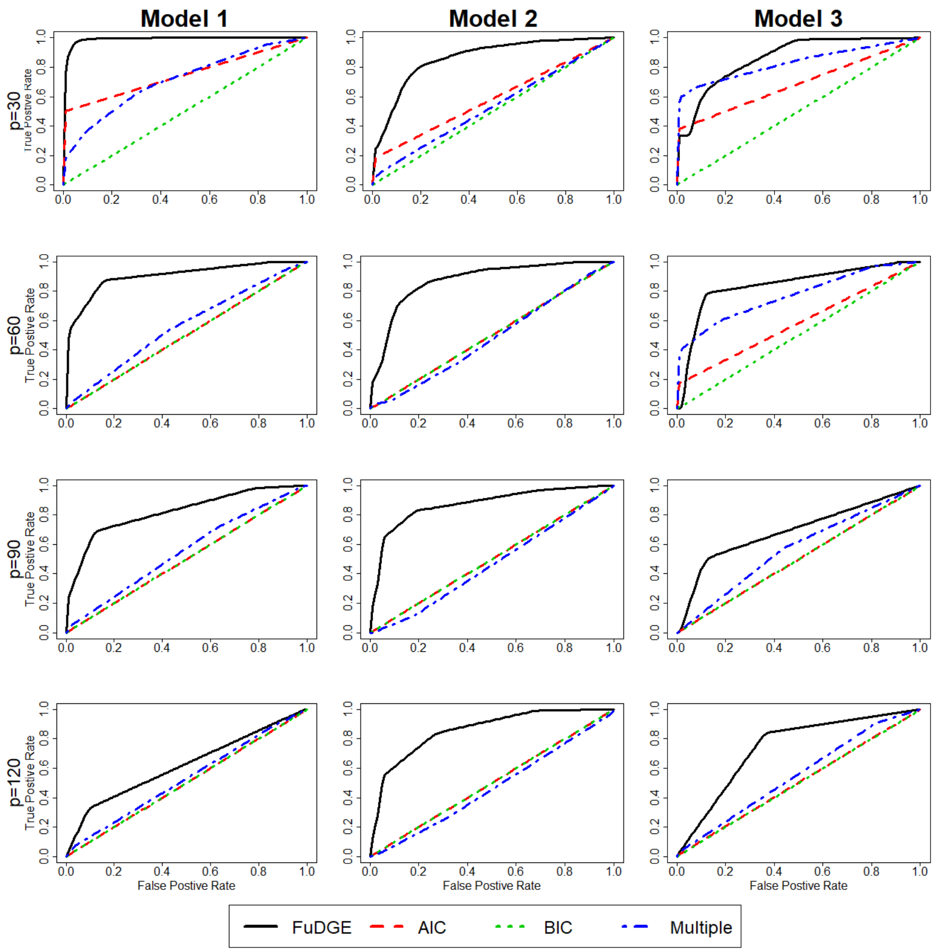

Figure 1: Average ROC curves across 30 simulations. Different columns correspond to different models, different rows correspond to different dimensions.

for 200 evenly spaced time points $0 = t_1 \leq \cdots \leq t_{200} = 1$. h^Y_{ijk} are obtained in a similar way. For each observation, we first estimate a function by fitting an L-dimensional B-spline basis. We then use these estimated functions for FPCA and our direct estimation procedure. Both M and L are chosen by 5-fold cross-validation as discussed in [18]. Since ϵ_n in (2.9) is usually very small in practice, we simply let $\hat{E}_\Delta = \{(j,l) \in V^2 : j \neq l \text{ and } \|\hat{\Delta}^M_{jl}\|_F + \|\hat{\Delta}^M_{lj}\|_F > 0\}$. We can form a receiver operating characteristics (ROC) curve for recovery of E_Δ by using different values of the group lasso penalty λ_n defined in (2.7).

We compare FuDGE to three competing methods. The first two competing methods separately estimate two functional graphical models using fglasso from [18]. Specifically, we use fglasso to estimate $\hat{\Theta}^{X,M}$ and $\hat{\Theta}^{Y,M}$. We then set $\hat{E}_\Delta$ to be all edges $(j,l) \in V^2$ such that $\|\hat{\Theta}^{X,M}_{jl} - \hat{\Theta}^{Y,M}_{jl}\|_F > \zeta$. For each separate fglasso problem, the penalization parameter is selected by maximizing AIC in first competing method and maximizing BIC in second competing method. We define the degrees of freedom for both AIC and BIC to be the number of edges included in the graph times M^2. We form an ROC curve by using different values of ζ.

The third competing method ignores the functional nature of the data. We select 15 equally spaced time points and implement a direct estimation method at each time point. Specifically, for each t, $X_i(t)$ and $Y_i(t)$ are simply p-dimensional random vectors, and we use their sample covariances in (2.7) to obtain a $p \times p$ matrix $\hat{\Delta}$. This produces 15 differential graphs, and we use a majority vote to form a single differential graph. The ROC curve is obtained by changing λ_n, the L_1 penalty used for all time points.

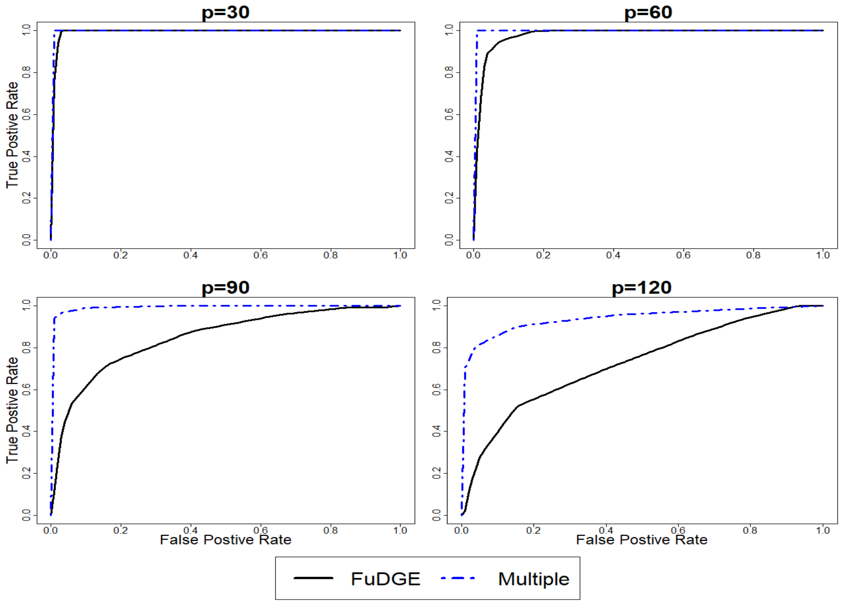

Figure 2: Average ROC curves across 30 simulations of example that multiple network strategy works better

For each setting and method, the ROC curve averaged across the 30 replications is shown in Figure 1. We see that FuDGE clearly has the best overall performance in recovering the support of differential graph. Among the competing methods, ignoring the functional structure and using a majority vote generally performs better than separately estimating two functional graphs. A table with the average area under the ROC curve is given in the appendix.

4.2 Example that combination of multiple networks at discrete time points works better

By construction, the simulations presented in Section 4.1 are estimating E_Δ defined in (2.3), which is not equivalent to

$$\tilde{E}_\Delta(t) = \left\{(j,l) \in V^2 : j \neq l, |\tilde{C}^X_{jl}(t) - \tilde{C}^Y_{jl}(t)| \neq 0\right\}, \tag{4.1}$$

where

$$\tilde{C}^X_{jl}(t) = \text{Cov}\left(X_{ij}(t), X_{il}(t) \mid \{X_{ik}(t)\}_{k\neq j,l}\right), \tag{4.2}$$

and $\tilde{C}^Y_{jl}(t)$ defined similarly. However, when $\tilde{E}_\Delta(t) = E_\Delta, \forall t$, then the differential structure can be recovered by considering individual time points. Since considering time points individually requires estimating fewer parameters than the functional version, the multiple networks strategy performs better than FuDGE.

Here, data are generated with $n_X = n_Y = 100$, and $p = 30, 60, 90, 120$. We repeat the simulation 30 times for each p. The model setting here is similar to model 2 in Section 4.1. However, we make two major changes. First, when we generate the functional variables, we use a 5-dimensional Fourier basis, so that all basis are supported over the entire interval, rather than disjoint support as in Section 4.1. Second, we set matrix W to be diagonal. Specifically, we let $W_{kk} = c$ for $k = 1, 2, \cdots, 5$ and $W_{km} = 0$ for $k \neq m$, where c is drawn uniformly from $[0.6, 1]$, and scaled by $1/2$ for $p = 30$, $1/3$ for $p = 60$, and $1/4$ for $p = 90$. All other settings are the same. The average ROC curves are shown in Figure 2, and the mean area under the curves are shown in Table 2 in section D.2 of supplementary material.

In Section 4.1 we considered extreme settings where the data must be treated as functions, and here we consider an extreme setting where the functional nature is irrelevant. In practice, however, the data may often lie between these two settings, and the method which performs better should depend on the variation of the differential structure across time. However, as it may be hard to measure this variation in practice, treating the data as functional objects should be a more robust choice.

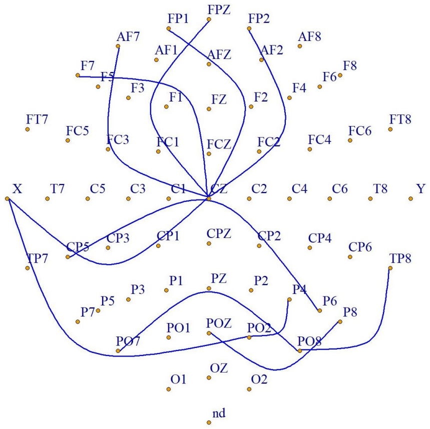

Figure 3: Estimated differential graph for EEG data. The anterior region is the top of the figure and the posterior region is the bottom of the figure.

4.3 Neuroscience application

We apply our method to electroencephalogram (EEG) data obtained from an alcoholism study [29, 6, 18] which included 122 total subjects; 77 in an alcoholic group and 45 in the control group. Specifically, the EEG data was measured by placing $p = 64$ electrodes on various locations on the subject's scalp and measuring voltage values across time. We follow the preprocessing procedure in [8, 31], which filters the EEG signals at α frequency bands between 8 and 12.5 Hz.

[18] separately estimate functional graphs for both groups, but we directly estimate the differential graph using FuDGE. We choose λ_n so that the estimated differential graph has approximately 1% of possible edges. The estimated edges of the differential graph are shown in Figure 3.

We see that edges are generally between nodes located in the same region–either the anterior region or the posterior region–and there is no edge that crosses between regions. This observation is consistent with the result in [18] where there are no connections between frontal and back regions for both groups. We also note that electrode CZ, lying in the central region has a high degree in the estimated differential graph. While there is no direct connection between anterior and posterior regions, the central region may play a role in helping the two parts communicate.

5 Discussion

In this paper, we propose a method to directly estimate the differential graph for functional graphical models. In certain settings, direct estimation allows for the differential graph to be recovered consistently, even if each underlying graph cannot be consistently recovered. Experiments on simulated data also show that preserving the functional nature of the data rather than treating the data as multivariate scalars can also result in better estimation of the difference graph.

A key step in the procedure is first representing the functions with an M-dimensional basis using FPCA, and Assumption 3.2 ensures that there exists some M large enough so that the signal, τ, is larger than the bias due to using a finite dimensional representation, ν. Intuitively, ν is tied to the eigenvalue decay rate; however, we defer derivation of the explicit connection for future work. Finally, we have provided a method for direct estimation of the differential graph, but development of methods which allow for inference and hypothesis testing in functional differential graphs would be fruitful avenues for future work. For example, [7] has developed inference tools for high-dimensional Markov networks, future works may extend their results to functional graph setting.

References

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imag. Sci.*, 2:183–202, 2009.

[2] D. Bosq. *Linear processes in function spaces*, volume 149 of *Lecture Notes in Statistics*. Springer-Verlag, New York, 2000. Theory and applications.

[3] T. T. Cai. Global testing and large-scale multiple testing for high-dimensional covariance structures. *Annual Review of Statistics and Its Application*, 4(1):423–446, 2017.

[4] M. Drton and M. H. Maathuis. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4:365–393, 2017.

[5] T. Hsing and R. Eubank. *Theoretical foundations of functional data analysis, with an introduction to linear operators*. Wiley Series in Probability and Statistics. John Wiley & Sons, Ltd., Chichester, 2015.

[6] L. Ingber. Statistical mechanics of neocortical interactions: Canonical momenta indicators of electroencephalography. *Physical Review E*, 55(4):4578–4593, 1997.

[7] B. Kim, S. Liu, and M. Kolar. Two-sample inference for high-dimensional markov networks. *arXiv preprint arXiv:1905.00466*, 2019.

[8] G. G. Knyazev. Motivation, emotion, and their inhibitory control mirrored in brain oscillations. *Neuroscience & Biobehavioral Reviews*, 31(3):377–395, 2007.

[9] M. Kolar, L. Song, A. Ahmed, and E. P. Xing. Estimating Time-varying networks. *Ann. Appl. Stat.*, 4(1): 94–123, 2010.

[10] S. L. Lauritzen. *Graphical Models*, volume 17 of *Oxford Statistical Science Series*. The Clarendon Press Oxford University Press, New York, 1996. Oxford Science Publications.

[11] B. Li and E. Solea. A nonparametric graphical model for functional data with application to brain networks based on fMRI. *J. Amer. Statist. Assoc.*, 113(524):1637–1655, 2018.

[12] S. Liu, J. A. Quinn, M. U. Gutmann, T. Suzuki, and M. Sugiyama. Direct learning of sparse changes in Markov networks by density ratio estimation. *Neural Comput.*, 26(6):1169–1197, 2014.

[13] N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the lasso. *Ann. Stat.*, 34(3):1436–1462, 2006.

[14] S. Na, M. Kolar, and O. Koyejo. Estimating differential latent variable graphical models with applications to brain connectivity. *arXiv preprint arXiv:1909.05892*, 2019.

[15] S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Stat. Sci.*, 27(4):538–557, 2012.

[16] M. E. J. Newman. The structure and function of complex networks. *SIAM Rev.*, 45(2):167–256, 2003.

[17] N. Parikh and S. P. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

[18] X. Qiao, S. Guo, and G. M. James. Functional Graphical Models. *J. Amer. Statist. Assoc.*, 114(525): 211–222, 2019.

[19] R. Tibshirani. Proximal gradient descent and acceleration. *Lecture Notes*, 2010.

[20] J.-L. Wang, J.-M. Chiou, and H.-G. Müller. Functional data analysis. *Annual Review of Statistics and Its Application*, 3(1):257–295, 2016.

[21] Y. Wang, C. Squires, A. Belyaeva, and C. Uhler. Direct estimation of differences in causal graphs. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 3774–3785, 2018.

[22] P. Xu and Q. Gu. Semiparametric differential graph models. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1064–1072. Curran Associates, Inc., 2016.

[23] F. Yao and T. C. M. Lee. Penalized spline models for functional principal component analysis. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 68(1):3–25, 2006.

[24] M. Yu, V. Gupta, and M. Kolar. Statistical inference for pairwise graphical models using score matching. In *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 2016.

[25] M. Yu, V. Gupta, and M. Kolar. Simultaneous inference for pairwise graphical models with generalized score matching. *arXiv preprint arXiv:1905.06261*, 2019.

[26] H. Yuan, R. Xi, C. Chen, and M. Deng. Differential network analysis via lasso penalized D-trace loss. *Biometrika*, 104(4):755–770, 2017.

[27] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. B*, 68:49–67, 2006.

[28] C. Zhang, H. Yan, S. Lee, and J. Shi. Dynamic multivariate functional data modeling via sparse subspace learning. *CoRR*, abs/1804.03797, 2018, `arXiv:1804.03797`.

[29] X. L. Zhang, H. Begleiter, B. Porjesz, W. Wang, and A. Litke. Event related potentials during object recognition tasks. *Brain Research Bulletin*, 38(6):531–538, 1995.

[30] S. D. Zhao, T. T. Cai, and H. Li. Direct estimation of differential networks. *Biometrika*, 101(2):253–268, 2014.

[31] H. Zhu, N. Strawn, and D. B. Dunson. Bayesian graphical models for multivariate functional data. *J. Mach. Learn. Res.*, 17:Paper No. 204, 27, 2016.