Graph Classification using Signal Subgraphs: Applications in Statistical Connectomics

Joshua T. Vogelstein, William R. Gray, R. Jacob Vogelstein, and Carey E. Priebe

Abstract—This manuscript considers the following "graph classification" question: given a collection of graphs and associated classes, how can one predict the class of a newly observed graph? To address this question we propose a statistical model for graph/class pairs. This model naturally leads to a set of estimators to identify the class-conditional signal, or "signal subgraph," defined as the collection of edges that are probabilistically different between the classes. The estimators admit classifiers which are asymptotically optimal and efficient, but differ by their assumption about the "coherency" of the signal subgraph (coherency is the extent to which the signal edges "stick together" around a common subset of vertices). Via simulation, the best estimator is shown to be not just a function of the coherency of the model, but also the number of training samples. These estimators are employed to address a contemporary neuroscience question: can we classify "connectomes" (brain-graphs) according to sex? The answer is yes, and significantly better than a naïve strategy. Synthetic data analysis demonstrates that even when the model is correct, given the relatively small number of training samples, the estimated signal subgraph should be taken with a grain of salt. We conclude by discussing several possible extensions.

Index Terms—statistical inference, graph theory, network theory, structural pattern recognition, connectome, classification.

1 Introduction

Graphs are emerging as a prevalent form of data representation in fields ranging from optical character recognition and chemistry [1] to neuroscience [2]. While statistical inference techniques for vector-valued data are widespread, statistical tools for the analysis of graph-valued data are relatively rare [1]. In this work we consider the task of *labeled graph classification*: given a collection of labeled graphs and their corresponding classes~~labels~~, can we accurately infer the class ~~label~~ for a new graph? Note that we assume throughout that each vertex has a unique label, and that all graphs have the same number of vertices with the same vertex labels.

We propose and analyze a joint graph/class model—sufficiently simple to characterize its asymptotic properties, and sufficiently rich to afford useful empirical applications. This model admits a class-conditional signal encoded in a subset of edges, the *signal subgraph*. Finding the signal subgraph amounts to providing an understanding of the differences between the two graph classes. Moreover, borrowing a term from the compressive sensing literature [3], we are interested in learning to what extent this signal is *coherent*; that is, to what extent are the signal subgraph edges incident to a relatively small set of vertices. In other words, if the signal is sparse in the edges, then the signal subgraph is incoherent, if it is also sparse in the vertices, then the signal subgraph is coherent (we formally define these notions below). ~~If the signal subgraph is strongly coherent, this suggests that the signal is carried by a few important vertices in the graph; otherwise, the signal is more widely distributed across the graph, with no particularly special vertices.~~

This graph-model based approach is qualitatively different from most previous approaches which utilize only unique vertex labels or graph structure. In the former case, simply representing the adjacency matrix with a vector and applying standard machine learning techniques ignores graph structure (for instance, it is not clear how to implement a coherent signal subgraph estimator in this representation). In the latter case, computing a set of graph invariants (such as clustering coefficient), and then classifying using only these invariants ignores vertex labels [1], [4], [5]. ~~Neither of these approaches use both vertex labels and graph structure.~~

While some of the above approaches consider attributed vertices or edges, we are unable to find any that utilize both *unique* vertex labels and graph structure. The field of *connectomics* (the study of brain-graphs), however, is ripe with many examples of brain-graphs with vertex labels. In invertebrate brain-graphs, for example, often each neuron is named, such that one can compare neurons across individuals of the same species [6]. In vertebrate neurobiology, while neurons are rarely named, "neuron types" [7] and neuroanatomical regions [8] are named. Moreover, a widely held view is that many psychiatric issues are fundamentally "connectopathies" [9], [10].

- *J.T. Vogelstein and C.E. Priebe are with the Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD 21218. E-mail: {joshuav,cep}@jhu.edu*
- *W.R. Gray and R.J. Vogelstein are with the Johns Hopkins University Applied Physics Laboratory, Laurel, MD, 20723.*

For prognostic and diagnostic purposes, merely being able to differentiate groups of brain-graphs from one another is sufficient. However, for treatment, it is desirable to know which vertices and/or edges are malfunctioning, such that therapy can be targeted to those locations. This is the motivating application for our work.

We demonstrate via theory, simulation, analysis of a neurobiological data set (magnetic resonance based connectome sex classification), and synthetic data analysis, that utilizing graph structure can significantly enhance classification accuracy. However, the best approach for any particular data set is not just a function of the model, but also the amount of data. Moreover, even when the model is true, given a relatively small sample size, the estimated signal subgraph will often overlap with the truth, but not fully capture it. Nonetheless, the classifiers described below still significantly outperform the benchmarks.

2 METHODS

2.1 Setting

Let $\mathbb{G} : \Omega \to \mathcal{G}$ be a graph-valued random variable with samples G_i. Each graph $G = (\mathcal{V}, E)$ is defined by a set of V vertices, $\mathcal{V} = \{v_i\}_{i\in[V]}$, where $[V] = \{1, \ldots, V\}$, and a set of edges between pairs of vertices $E \subseteq V \times V$. Let $A : \Omega \to \mathcal{A}$ be an adjacency matrix-valued random variable taking values $a \in \mathcal{A} \subseteq \mathbb{R}^{V\times V}$, identifying which vertices share an edge. Let $Y : \Omega \to \mathcal{Y}$ be a discrete-valued random variable with samples y_i. Assume the existence of a collection of n exchangeable samples of graphs and their corresponding classes from some true but unknown joint distribution: $\{(\mathbb{G}_i, Y_i)\}_{i\in[n]} \overset{exch.}{\sim} F_{\mathbb{G},Y}$. Our aim (exploitation task) is to build a graph classifier that can take a new graph, $\mathbb{G}$, and correctly estimate its class, y, assuming that they are jointly sampled from the same distribution, $F_{\mathbb{G},Y}$. Moreover, we are interested solely in graph classifiers that are *interpretable* with respect to the vertices and edges of the graph. In other words, nonlinear manifold learning, feature extraction, and related approaches are unacceptable.

2.2 Model

~~A model defines the set of distributions under consideration. In the graph classification domain, we~~ Consider the model, $\mathcal{F}_{\mathbb{G},Y}$, which includes all joint distributions over graphs and classes under consideration: $\mathcal{F}_{\mathbb{G},Y} = \{F_{\mathbb{G},Y}(\cdot;\boldsymbol{\theta}) : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, where $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ indexes the distributions. ~~Two standard approaches for tackling a classification problem are (i) the *generative* approach and (ii) the *discriminative* approach. In a generative strategy, one decomposes the joint distribution into a product of a likelihood term and a prior term: $F_{\mathbb{G},Y} = F_{\mathbb{G}|Y}F_Y$. In a discriminative strategy, one decomposes the joint distribution into a posterior term and a marginal term: $F_{\mathbb{G},Y} = F_{Y|\mathbb{G}}F_{\mathbb{G}}$.~~ We proceed via a hybrid generative-discriminative approach [11] whereby we describe a generative model and place constraints on the discriminant boundary.

First, assume that each graph has the same set of uniquely labeled vertices, so that all the variability in the graphs is in the adjacency matrix, which implies that $F_{\mathbb{G},Y} = F_{A,Y}$. Second, assume edges are independent; that is, $F_{A,Y} = \prod_{u,v\in\mathcal{E}} F_{A_{uv},Y}$, where $\mathcal{E} \subseteq V \times V$ is the set of all possible edges. Now, consider the generative decomposition $F_{A,Y} = F_{A|Y}F_Y$, and let $F_{uv|y} = F_{A_{uv}|Y=y}$ and $\pi_y = F_{Y=y}$. Third, assume the existence of a class-conditional difference; that is, $F_{uv|0} \neq F_{uv|1}$ for some $(u, v) \in \mathcal{E}$, and denote the edges satisfying this condition the *signal subgraph*, $\mathcal{S} = \{(u, v) \in \mathcal{E} : F_{uv|0} \neq F_{uv|1}\}$. Fourth, although the following theory and algorithms are valid for both directed and undirected graphs, for concreteness, assume that the graphs are *simple* graphs; that is, undirected, with binary edges, and lacking (self-) loops (so $\mathcal{E} = \binom{V}{2}$). Thus, the likelihood of an edge between vertex u and v is given by a Bernoulli random variable with a scalar probability parameter: $F_{uv|y}(A_{uv}) = \text{Bern}(A_{uv}; p_{uv|y})$. Together, these four assumptions imply the following model:

$$\mathcal{F}_{\mathbb{G},Y} = \{F_{A,Y}(a, y; \boldsymbol{\theta}) \quad \forall a \in \mathcal{A}, y \in \mathcal{Y} : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}, \quad (1)$$

where

$$F_{A,Y}(a, y; \theta) = \prod_{uv\in\mathcal{S}} \text{Bern}(a_{uv}; p_{uv|y})\pi_y \times \prod_{uv\in\mathcal{E}\setminus\mathcal{S}} \text{Bern}(a_{uv}; p_{uv}), \quad (2)$$

and $\boldsymbol{\theta} = \{\boldsymbol{p}, \boldsymbol{\pi}, \mathcal{S}\}$. The likelihood parameter is constrained such that each element must be between zero and one: $\boldsymbol{p} \in (0, 1)^{\binom{V}{2}\times|\mathcal{Y}|}$. The prior parameter, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{|\mathcal{Y}|})$, must have elements greater than or equal to zero and sum to one: $\pi_y \geq 0, \sum_y \pi_y = 1$. The signal subgraph parameter is a non-empty subset of the set of possible edges, $\mathcal{S} \subseteq \mathcal{E}$ and $\mathcal{S} \neq \emptyset$.

We consider up to two additional constraints on $\mathcal{S}$. First, the size of the signal subgraph may be constrained such that $|\mathcal{S}| \leq s$. Second, the minimum number of vertices onto which the collection of edges is incident to is constrained such that $\mathcal{S} = \{(u, v) : u \cup v \in \mathcal{U}\}$, where $\mathcal{U}$ is a set of *signal vertices* with $|\mathcal{U}| \leq m$. Edges in the signal subgraph are called *signal edges*. While it may be natural to treat $\mathcal{S}$ as a prior, we treat it as a parameter of the model; the constraints, s and m, are considered *hyper-parameters*.

Note that given a specification of the class-conditional likelihood of each edge and class-prior, one completely defines a joint distribution over graphs and classes; the signal subgraph is implicit in that parameterization. However, the likelihood parameters for all edges not in the signal subgraph,

$p_{uv|y} = p_{uv} \, \forall \, y \in \mathcal{Y}, (u,v) \notin \mathcal{S}$, are *nuisance* parameters; that is, they contain no class-conditional signal. When computing a relative posterior class estimate, these nuisance parameters cancel in the ratio.

2.3 Classifier

~~Formally, we say that~~ A graph classifier, $h \in \mathcal{H}$, is any function satisfying $h : \mathcal{G} \to \mathcal{Y}$. We desire the "best" possible classifier, h_*. To define best, we first choose a loss function, $\ell_h : \mathcal{G} \times \mathcal{Y} \to \mathbb{R}_+$, specifically the $0-1$ loss function:

$$\ell_h(G, y) \stackrel{\triangle}{=} \mathbb{I}\{h(G) \neq y\}, \tag{3}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function, equaling one whenever its argument is true, and zero otherwise. Further, let risk, $R : \mathcal{F} \times \mathcal{H} \to \mathbb{R}_+$ be the expected loss under the true distribution:

$$R(F, h) \stackrel{\triangle}{=} \mathbb{E}_F[\ell_h(\mathbb{G}, Y)]. \tag{4}$$

The Bayes optimal (best) classifier for a given distribution F minimizes risk. It can be shown that the classifier that maximizes the class-conditional posterior $F_{Y|\mathbb{G}}$ is optimal [12]:

$$\begin{aligned} h_* &= \operatorname*{argmin}_{h \in \mathcal{H}} \mathbb{E}_F[\ell_h(\mathbb{G}, Y)] \\ &= \operatorname*{argmax}_{y \in \mathcal{Y}} F_{\mathbb{G}|Y=y} F_{Y=y}. \end{aligned} \tag{5}$$

Given the proposed model, Eq. (5) can be further factorized using the above four assumptions:

$$h_*(G) = \operatorname*{argmax}_{y \in \mathcal{Y}} \prod_{u,v \in \mathcal{S}} \text{Bern}(A_{uv}; p_{uv|y}) \pi_y. \tag{6}$$

Unfortunately Bayes optimal classifiers are typically unavailable. In such settings, it is therefore desirable to induce a classifier estimate from a set of *training data*. Formally, let $\mathcal{T}_n = \{(\mathbb{G}_i, Y_i)\}_{i \in [n]}$ denote the training corpus, where each graph-class pair is sampled exchangeably from the true but unknown distribution: $(\mathbb{G}_i, Y_i) \stackrel{exch.}{\sim} F_{\mathbb{G},Y}$. Given such a training corpus and an unclassified graph G, an induced classifier predicts the true (but unknown) class of G, $\widehat{h} \colon \mathcal{G} \times (\mathcal{G} \times \mathcal{Y})^n \to \mathcal{Y}$. When a model $\mathcal{F}_{\mathbb{G},Y}$ is specified, a beloved approach is to use a *Bayes plugin classifier*. Due to the above simplifying assumptions, the Bayes plugin classifier for this model is defined as follows. First, estimate the model parameters $\boldsymbol{\theta} = \{\mathcal{S}, \boldsymbol{p}, \boldsymbol{\pi}\}$. Second, plug those estimates into the above equation. The result is a Bayes plugin graph classifier:

$$\widehat{h}(G; \mathcal{T}_n) \stackrel{\triangle}{=} \operatorname*{argmax}_{y \in \mathcal{Y}} \prod_{u,v \in \widehat{\mathcal{S}}} \widehat{p}_{uv|y}^{a_{uv}} (1 - \widehat{p}_{uv|y})^{(1-a_{uv})} \widehat{\pi}_y, \tag{7}$$

where the Bernoulli probability is explicit. To implement such a classifier estimate, we specify estimators for $\mathcal{S}$, $\boldsymbol{\pi}$ and $\boldsymbol{p}$.

2.4 Estimators

2.4.1 Desiderata

~~In this section we describe estimators for the above model. An *estimator* is a function that maps from the multiple-sample space to the parameter space: $\widehat{\boldsymbol{\theta}}_n : \Xi^n \to \boldsymbol{\Theta}$. The output of this function is called the *estimate*. In the graph classification domain, for example, $\Xi = \mathcal{G} \times \mathcal{Y}$. In a slight abuse of notation, we will also refer to the sequence of estimators, $\widehat{\boldsymbol{\theta}}_1, \widehat{\boldsymbol{\theta}}_2, \ldots$, as an estimator.~~ We desire a ~~(~~sequence of~~)~~ estimators, $\widehat{\boldsymbol{\theta}}_1, \widehat{\boldsymbol{\theta}}_2, \ldots$, that satisfy the following desiderata:

- **Consistent**: an estimator is consistent (in some specified sense) if its sequence converges in the limit to the true value: $\lim_{n \to \infty} \widehat{\boldsymbol{\theta}}_n = \boldsymbol{\theta}$.
- **Robust**: an estimator is robust if the resulting estimate is relatively insensitive to small model misspecifications. Because the space of models is massive (uncountably infinite), it is intractable to consider all misspecifications, so we consider only a few of them, as described below.
- **Quadratic complexity**: computational time complexity should be no more than quadratic in the number of vertices.
- **Interpretable**: we desire that the parameters are interpretable with respect to a subset of vertices and/or edges.

In addition to the above theoretical desiderata, we also desire appealing finite sample and empirical performance.

2.4.2 Signal Subgraph Estimators

Naïvely, one might consider a search over all possible signal subgraphs by plugging each one in to the classifier and selecting the best performing option. This strategy is intractable because the number of signal subgraphs scales super-exponentially with the number of vertices (see Figure 1, left panel). Specifically, the number of possible edges in a simple graph with V vertices is $d_V = \binom{V}{2}$, so the number of unique possible signal subgraphs is $2^{\binom{V}{2}}$. Searching over all of them is therefore computationally taxing. We therefore consider several alternatives.

Before proceeding, recall that each edge is independent; thus, one can evaluate each edge separately (although not necessarily advisable, consider the Stein estimator [13]). Formally, consider a hypothesis test for each edge. The null hypothesis is that the class-conditional edge distributions are the same, so $H_0 : F_{uv|0} = F_{uv|1}$. The composite alternative hypothesis is that they differ, $H_A : F_{uv|0} \neq F_{uv|1}$. Given such hypothesis tests, one can construct test statistics $T_{uv}^{(n)} : \mathcal{T}_n \to \mathbb{R}_+$. We reject the null in favor of the alternative whenever the value of the test statistic is greater than some critical-value: $T_{uv}^{(n)}(\mathcal{T}_n) > c$. We

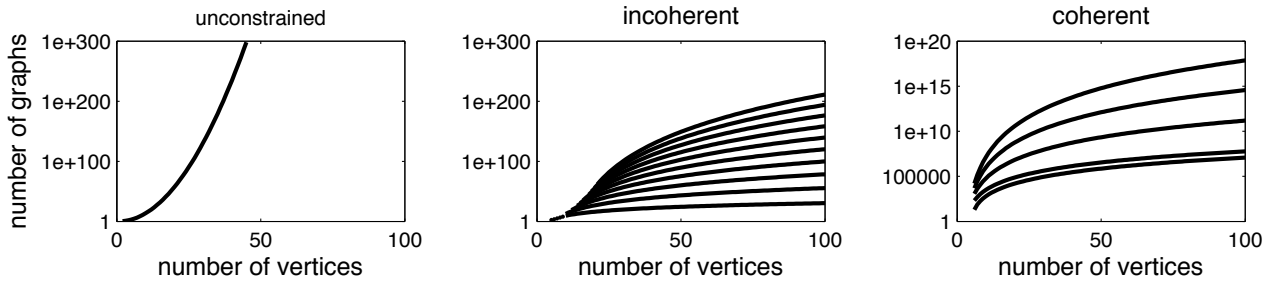

Fig. 1. Exhaustive searches for the signal subgraph, even given severe constraints, are computationally intractable even for small graphs. The three panels illustrate the the number of unique simple subgraphs as a function of the number of vertices V for the three different constraint types considered: unconstrained, edge constrained, and both edge and vertex constrained (coherent). Note the ordinates are all log scale. On the left is the unconstrained scenario, that is, all possible subgraphs for a given number of vertices. In the middle panel, each line shows the number of subgraphs with fixed number of signal edges, s, ranging from 10 to 100, incrementing by 10 with each line. The right panel shows the number of subgraphs for various fixed s and only a single signal vertex; that is, all edges are incident to one vertex.

can therefore construct a *significance matrix* $\boldsymbol{T} \triangleq T_{uv}^{(n)}$, which is the sufficient statistic for the signal subgraph estimators. Example test statistics include Fisher's and chi-squared, which will be discussed further below. Whichever test statistic one uses, the sufficient statistics are captured in a $2 \times |\mathcal{Y}|$ contingency table, indicating the number of times edge u, v was observed in each class. For example, the two-class contingency table for each edge is given by:

	Class 0	Class 1	Total
Edge	$n_{uv\|0}$	$n_{uv\|1}$	n_{uv}
No Edge	$n_0 - n_{uv\|0}$	$n_1 - n_{uv\|1}$	$n - n_{uv}$
Total	n_0	n_1	n.

2.4.2.1 *Incoherent Signal Subgraph Estimators*: Assume the size of the signal subgraph, $|\mathcal{E}| = s$, is known. The number of subgraphs with s edges on V vertices is given by $\binom{d_V}{s}$; also super-exponential (see Figure 1, middle panel). Thus searching them all is currently computationally intractable. When s is given, under the independent edge assumption, one can choose the critical value *a posteriori* to ensure that only s edges are rejected under the null (that is, have significant class-conditional differences):

$$\begin{aligned} &\text{minimize } c \\ &\text{subject to } \sum_{(u,v)\in\mathcal{E}} \mathbb{I}\{T_{uv}^{(n)} < c\} \geq s. \end{aligned} \tag{8}$$

Therefore, an estimate of the signal subgraph is the collection of s edges with minimal test statistics. Let $T_{(1)} < T_{(2)} < \cdots < T_{(d_V)}$ indicate the *ordered* test statistics (dropping the superscript indicating the number of samples for brevity). Then, the *incoherent signal subgraph estimator* is given by $\widehat{\mathcal{S}}_n^{inc}(s) = \{e_{(1)}, \ldots, e_{(s)}\}$, where $e_{(u)}$ indicates the u^{th} edge ordered by significance of its test statistic, $T_{(u)}$. Pseudocode for implementing the incoherent signal-subgraph estimator is provided in Algorithm 1, and MATLAB code is available from http://jovo.me.

Algorithm 1 Pseudocode for estimating incoherent signal-subgraph.

Input: $\mathcal{T}_n$ and s
Output: $\widehat{\mathcal{S}}_n^{inc}(s)$
1: Compute test statistics $T_{uv}^{(n)}$ for all $(u, v) \in \mathcal{E}$
2: Sort each edge according to its test-statistic rank, $T_{(1)} \leq T_{(2)} \leq \cdots \leq T_{(d_V)}$
3: Let $\widehat{\mathcal{S}}_n^{inc}(s) = \{e_{(1)}, \ldots, e_{(s)}\}$

2.4.2.2 *Coherent Signal Subgraph Estimators*: In addition to the size of the signal subgraph, also assume that each of the edges in the signal subgraph are incident to one of m special vertices called *signal vertices*. While this assumption further constrains the candidate sets of edges, the number of feasible sets still scales super exponentially (see Figure 1, right panel). Therefore, we again take a greedy approach.

First, compute the significance of each edge, as above, yielding ordered test statistics. Second, rank edges by significance with respect to each vertex, $e_{k,(1)} \leq e_{k,(2)} \leq \ldots \leq e_{k,(n-1)}$ for all $k \in \mathcal{V}$. Third, initialize the critical value at zero, $c = 0$. Fourth, assign each vertex a score equal to the number of edges incident to that vertex more significant than the critical value, $w_{v;c} = \sum_{u\in[V]} \mathbb{I}\{T_{v,u} > c\}$. Fifth, sort the vertex significance scores, $w_{(1);c} \geq w_{(2);c} \geq \cdots \geq w_{(V);c}$. Sixth, check if there exists m vertices whose scores sum to greater than or equal the size of the signal subgraph, s. That is, check whether the

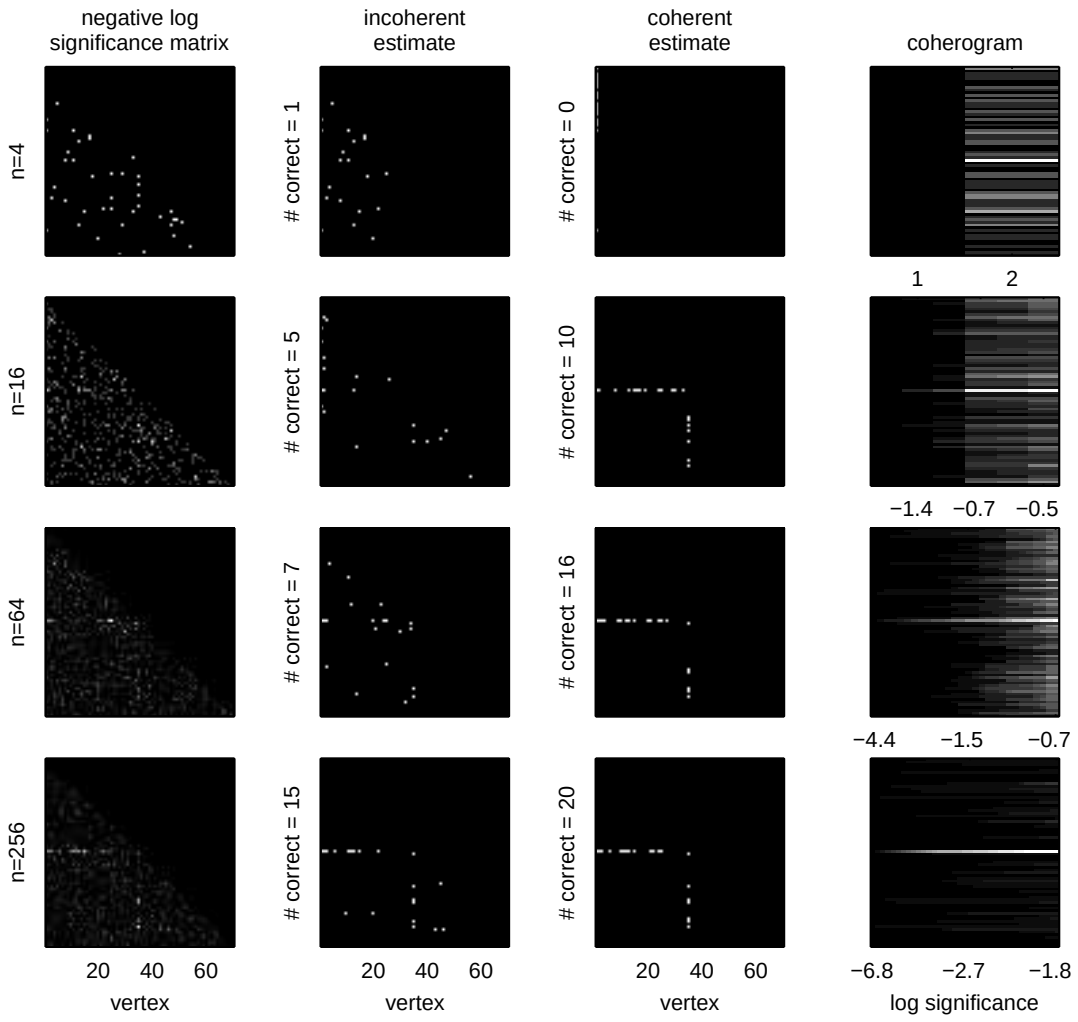

Fig. 2. An example of the coherent signal subgraph estimate's improved accuracy over the incoherent signal subgraph estimate, for a particular homogeneous two-class model specified by: $\mathcal{M}_{70}(1, 20; 0.5, 0.1, 0.3)$. Each row shows the same columns but for increasing the number of graph/class samples. The columns show the: (far left) negative log-significant matrix, computed using Fisher's exact test (lighter means more significant; each panel is scaled independent of the others because only relative significance matters here); (middle left) incoherent estimate of the signal subgraph; (middle right) coherent estimate of the signal subgraph; (far right) coherogram. As the number of training samples increases (lower rows), both the incoherent and coherent estimates converge to the truth (the ordinate labels of the middle panels indicate the number of edges correctly identified). For these examples, the coherent estimator tends to find more true edges. The coherogram visually depicts the coherency of the signal; it is also converging to the truth—the signal subgraph here contains a single signal vertex.

following optimization problem is satisfied:

$$\begin{aligned} &\text{minimize } c \\ &\text{subject to } \sum_{v \in [m]} w_{(v);c} \geq s. \end{aligned} \tag{9}$$

If so, call the collection of s most significant edges from within that subset the *coherent signal subgraph estimate*, $\widehat{\mathcal{S}}_n^{coh}(s, m)$. If not, increase c and go back to step four. Pseudocode for implementing the coherent signal-subgraph estimator is provided in Algorithm 2, and MATLAB code is available from http://jovo.me.

2.4.2.3 *Coherograms*: In the process of estimating the incoherent signal subgraph, one builds a "coherogram". Each column of the coherogram corresponds to a different critical value c, and each row corresponds to a different vertex v. The $(c, v)^{th}$ element of the coherogram $w_{v;c}$ is the number of edges incident to vertex v with test statistic larger than c. Thus, the coherogram gives a visual depiction of the coherence of the signal subgraph (see Figure 2, right column, for some examples).

Algorithm 2 Pseudocode for estimating coherent signal-subgraph.

Input: $\mathcal{T}_n$ and (s, m)
Output: $\widehat{\mathcal{S}}_n^{coh}(s, m)$

1: Compute test statistics $T_{uv}^{(n)}$ for all $(u, v) \in \mathcal{E}$
2: Sort each edge according to its vertex-conditional test-statistic rank, $T_{(1),k} \leq T_{(2),k} \leq \cdots \leq T_{(d_V),k}$ for all $k \in \mathcal{V}$
3: Let $c = 0$
4: Let $w_{v;c} = \sum_{u \in \mathcal{V}} \mathbb{I}\{T_{v,u} > c\}$ for all $v \in \mathcal{V}$
5: Let $w_c = \sum_{v \in [m]} w_{(v);c}$
6: **while** $w_c < s$ **do**
7: Let $c \leftarrow c + 1$
8: Update w_c
9: **end while**
10: Let $\widehat{\mathcal{S}}_n^{coh}(s, m)$ be the collection of s edges from amongst those that satisfy Eq. (**??**) for the final value of c.

2.4.3 Likelihood Estimators

The class-conditional likelihood parameters $p_{uv|y}$ are relatively simple. In particular, because the graphs are assumed to be simple, $p_{uv|y}$ is just a Bernoulli parameter for each edge in each class. The maximum likelihood estimator (MLE), which is simply the average value of each edge per class, is a principled choice:

$$\widehat{p}_{uv|y}^{MLE} = \frac{1}{n_y} \sum_{i|y_i=y} a_{uv}^{(i)}, \tag{10}$$

where $\sum_{i|y_i=y}$ indicates the sum is over all training samples from class y. Unfortunately, the MLE has an undesirable property; in particular, if the data contains no examples of an edge in a particular class, then the MLE will be zero. If the unclassified graph exhibits that edge, then the estimated probability of it being from that class is zero, which is undesirable. We therefore consider a smoothed estimator:

$$\widehat{p}_{uv|y} = \begin{cases} \eta_n & \text{if } \max_i a_{uv}^{(i)} = 0 \\ 1 - \eta_n & \text{if } \min_i a_{uv}^{(i)} = 1 \\ \widehat{p}_{uv|y}^{MLE} & \text{otherwise} \end{cases} \tag{11}$$

where we let $\eta_n = 1/(10n)$.

2.4.4 Prior Estimators

The priors are the simplest. The prior probabilities are Bernoulli, and we are concerned only with the case where $|\mathcal{Y}| \ll n$, so the maximum likelihood estimators suffice:

$$\widehat{\pi}_y = \frac{n_y}{n}, \tag{12}$$

where $n_y = \sum_{i \in [n]} \mathbb{I}\{y_i = y\}$.

2.4.5 Hyper-Parameter Selection

The signal subgraph estimators require specifying the number of signal edges s, as well as the number of signal vertices m for the coherent classifier. In both cases, the number of possible values of finite. In particular, $s \in [d_V]$ and $m \in [V]$. Thus, we implement cross-validation procedures, iterating over all possible settings of the hyper-parameters, to choose the hyper-parameters. For all simulated data, we compare hyper-parameter performance via a training and held-out set. For the real data application, the sample size is unfortunately too low to justify a held-out corpus, we therefore utilize a leave-one-out cross-validation procedure.

Note that the number of distinct test-statistic values is typically much smaller than the number of possible settings of s or m; specifically, the number of unique test statistic values will be $t \leq \min(|\mathcal{E}|, (n_0+1)(n_1+1))$. In practice, t is often be far less than either of the upper bounds, because not every edge has a unique contingency table. In such scenarios, certain settings of the hyper-parameters will lead to "ties", that is, edges that are equally valid under the assumptions. In such settings, we simply randomly choose edges satisfying the criterion.

2.4.6 All together

Putting the above pieces together, Algorithm **??** provides pseudo-code for implementing our signal-subgraph classifiers. MATLAB code is available from the first author's website, http://jovo.me

Algorithm 3 Pseudocode for training signal-subgraph classifiers.

Input: $(\mathbb{G}_i, Y_i)_{i \in [n]}$, and a set of constraints, $s \in \vec{s}$, $m \in \vec{m}$

1: Partition the data in to a training corpus and a held-out corpus (which might be the same)
2: Construct the contingency table for all $(u, v) \in \mathcal{E}$
3: Compute test statistics $T_{uv}^{(n)}$ for all $(u, v) \in \mathcal{E}$
4: Compute $\widehat{\mathcal{S}}_s^{inc}$ using Algorithm 1 for all $s \in \vec{s}$ and/or $\widehat{\mathcal{S}}_{m,s}^{coh}$ using Algorithm 2 for all $(m, s) \in \vec{m} \times \vec{s}$
5: Compute $\hat{p}_{uv|y}$ for all $(u, v) \in \widehat{\mathcal{S}}$'s using Eq. (11)
6: Compute π_y for all y using Eq. (12)
7: Calculate cross-validated error using Eq. (**??**) for all $\widehat{\mathcal{S}}$'s
8: Let $\widehat{\mathcal{S}} = \text{argmin}_{\widehat{\mathcal{S}}_{m,s}} \hat{L}_{\widehat{\mathcal{S}}_{m,s}}$

Output: $\widehat{\mathcal{S}}$, $\hat{p}_{uv|y} \forall (u, v) \in \widehat{\mathcal{S}}, y \in \{0, 1\}$

2.5 Finite Sample Evaluation Criteria

2.5.1 Likelihoods and priors

The likelihood and prior estimators will be evaluated with respect to robustness to model misspecifications, finite samples, efficiency, and complexity.

2.5.2 Classifier

We evaluate the classifier's finite sample properties using either hold-out or leave-one-out misclassification performance, depending on whether the data is simulated or experimental, respectively. Formally, given C equally sized subsets of the data, $\{\mathcal{T}_1, \ldots, \mathcal{T}_C\}$, the *cross-validated error* is given by

$$\widehat{L}_{\widehat{h}(\cdot;\mathcal{T}_n)} = \frac{1}{C}\sum_{c=1}^{C} \frac{1}{|\mathcal{T}_n \setminus \mathcal{T}_c|} \sum_{G \notin \mathcal{T}_c} \mathbb{I}\{\widehat{h}(G;\mathcal{T}_c) \neq y\}. \quad (13)$$

Given this definition, let $L_{\hat{\pi}}$ be the error of the classifier using only the prior estimates, and let L_* be the error for the Bayes optimal classifier.

To determine whether a classifier is significantly better than chance, we randomly permute the classes of each graph n_{MC} times, and then estimate a naïve Bayes classifier using the permuted data, yielding an empirical distribution. The p-value of a permutation test is the minimum fraction of Monte Carlo permutations that did better than the classifier of interest [14].

To determine whether a pair of classifiers are significantly different, we compare the leave-one-out classification results using McNemar's test [15].

2.5.3 Signal Subgraph Estimators

To evaluate absolute performance of the signal subgraph estimators, we define "miss-edge rate" as the fraction of true edges missed by the signal subgraph estimator:

$$R_n^x = \frac{1}{|\mathcal{S}|} \sum_{(u,v)\in\mathcal{S}} \mathbb{I}\{(u,v) \notin \widehat{\mathcal{S}}_n^x\}. \quad (14)$$

Further, we estimate the *relative rate* and *relative efficiency* to evaluate the relative finite sample properties of a pair of consistent estimators. The relative rate is simply $(1 - R_n^{inc})/(1 - R_n^{coh})$. Relative efficiency is the number of samples required for the coherent estimator to obtain the same rate as the incoherent estimator.

3 Estimator Properties

3.1 Likelihood and Prior Estimators

Lemma 3.1. *$\widehat{p}_{uv|y}$ as defined in Eq. (11) is an L-estimator.*

Proof: Huber defines an L-estimator as an estimator that is a linear combination of (possibly nonlinear functions of) the order statistics of the measurements [16]. Indeed, is a thresholded function of the minimum, maximum, and mean. □

Because L-estimators converge to the MLE, our estimators share all the nice asymptotic properties of the MLE. Moreover, L-estimators are known to be robust to certain model misspecifications [16]. The prior estimators are MLE's, and therefore also consistent and efficient. Both prior and likelihood estimates are trivial to compute, as closed-form analytic solutions are available for both.

3.2 Signal Subgraph Estimators

A variety of test statistics are available for computing the edge-specific class-conditional signal, $T_{uv}^{(n)}$. Fisher's exact test computes the probability of obtaining a table equal to or more extreme than the table resulting from the null hypothesis: that the two classes have the same probability of sampling an edge. In other words, Fisher's exact test is the most powerful statistical test assuming independent edges [17]. This leads to the following lemma:

Lemma 3.2. *$\widehat{\mathcal{S}}_n^x \to \mathcal{S}$ as $n \to \infty$ for $x \in \{inc, coh\}$ when computing $T_{uv}^{(n)}$ via Fisher's exact test, even when s (and m) are unknown.*

Proof: Whenever $p_{uv|0} \neq p_{uv|1}$, the p-value of Fisher's exact test converges to zero; whereas whenever $p_{uv|0} = p_{uv|1}$, the distribution of p-values converges to the uniform distribution on $[0, 1]$. Therefore, Fisher's exact test induces a consistent estimator of the signal subgraph as $n \to \infty$, assuming a fixed and finite V. Moreover, as $V \to \infty$, as long as $V/n \to 0$, Fisher's exact test remains consistent [17]. □

While most powerful, computing Fisher's exactly is computationally taxing. Fortunately, the chi-squared test is asymptotically equivalent to Fisher's test, and therefore shares those convergence properties [17]. Even the absolute difference of MLE's, $|\widehat{p}_{uv|1}^{MLE} - \widehat{p}_{uv|0}^{MLE}|$, which is trivially easy to compute, is asymptotically equivalent to Fisher's [17] and therefore consistent. ~~The implications of the above convergence properties are that any incoherent signal subgraph estimated using a consistent test statistic is a consistent signal subgraph estimator. Moreover, the incoherent signal subgraph estimator is robust to a variety of model misspecifications. Specifically, as long as all the marginal probability of all the edges in the signal subgraph are different between the two classes, $p_{uv|1} \neq p_{uv|0}$, any consistent test statistic will yield a consistent signal subgraph. For example, when the signal subgraph is coherent, even if m is unknown, the incoherent signal subgraph estimator will converge to the truth. More generally, even if the independent edge assumption is not satisfied, the incoherent estimator will converge to the truth. Moreover, the coherent signal subgraph estimator uses the same test statistics. Thus, it shares the above consistency and robustness properties. Estimating the coherent signal subgraph is more computationally time consuming.~~ What is lost by computational time, however, is typically gained by finite sample efficiency whenever the model ~~is coherent~~ does not induce too much bias, as will be shown below.

3.3 Bayes plugin classifier

Lemma 3.3. *The Bayes plug-in classifier, using the signal subgraph, likelihood, and prior estimators described above, is consistent under the model defined by Eq.* (2).

Proof: A Bayes plugin classifier is a consistent classifier whenever the estimates that are plugged in are consistent [12]. Because the likelihood, prior, and signal subgraph estimates are all consistent, the Bayes plugin classifier is also consistent. □

Note that naïve Bayes classifiers often exhibit impressive finite sample performance due to their winning the bias-variance trade-off relative to other classifiers [18]. In other words, even when edges are highly dependent, because marginal probability estimates are more efficient than joint probability estimates, an independent edge based classifier will often outperform a classifier based on dependencies.

4 Simulated Experiments

4.1 Simulation details

To better assess the finite sample properties of the signal subgraph estimators, we conduct a number of simulated experiments. Consider the following *homogeneous* model: each simple graph has $V = 70$ vertices. Class 0 graphs are Erdos-Renyi with probability p for each edge; that is, $f_{uv|0} = p \forall (u, v) \in \mathcal{E}$. Class 1 graphs are a mixture of two Erdos-Renyi models: all edges in the *signal subgraph* have probability q, and all others have probability p, so that $f_{uv|1} = q \forall (u, v) \in \mathcal{S}$, and $f_{uv|1} = p \forall (u, v) \in \mathcal{E} \backslash \mathcal{S}$. The signal subgraph is constrained to have m signal vertices and s signal edges. Let the class-prior probabilities be given by $F_{Y=0} = \pi$ and $F_{Y=1} = 1 - \pi$. Thus, the model is characterized by $F_{\boldsymbol{\theta}} = \mathcal{M}_V(m, s; \pi, p, q)$, where V is a constant, m and s are hyper-parameters, and π, p and q are parameters.

4.2 A simple demonstration

To provide some insight with respect to the finite sample performance of the incoherent and coherent signal subgraph estimators for this model, we run the following simulated experiments, with results depicted in Figure 2. In each row we sample from $\mathcal{M}_{70}(1, 20; 0.5, 0.1, 0.3)$ ~~(note that we are actually conditioning on the class-conditional sample size)~~. Given these n samples, we compute the significance matrix (first column), which contains the sufficient statistics for both estimators. The incoherent estimator simply chooses the s most significant edges as the signal subgraph (second column). The coherent estimator jointly estimates both the m signal vertices and the s signal edges incident to at least one of those vertices (third column). The coherogram shows the "coherency" of the data (fourth column).

From this figure, one might notice a few tendencies. First, both the incoherent and coherent signal subgraph seem to converge to the true signal subgraph. Second, while both estimators perform poorly with $n < 16$, the coherent estimator converges more quickly than the incoherent estimator. Third, the coherogram sharpens with additional samples, showing after only approximately 50 samples that this model is strongly coherent.

4.3 Quantitative Comparisons

To better characterize the relative performance of the two signal subgraph estimators, Figure 3 shows their performance as a function of the number of training samples, n, for the $\mathcal{M}_{70}(1, 20; 0.5, 0.1, 0.3)$ model. The top panel shows the mean and standard error of the missed-edge rate—the fraction of edges incorrectly identified—averaged over 200 trials. For essentially all n, the coherent estimator (black solid line) performs better than the incoherent estimator (gray solid line). We also compare the performance of our incoherent signal subgraph estimator with an ℓ_1-penalized logistic regression classifier, which we call the 'lasso' hereafter [19]. As expected, the missed edge rate for the lasso (gray dashed line) and the incoherent classifier are about the same. The improvement in signal edge detection of the coherent signal subgraph estimator over the incoherent's and lasso's performance ~~This~~ translates directly to improved classification performance (lower panel), where the plugin classifier using the coherent signal subgraph classifier has a better misclassification rate than either the incoherent signal subgraph classifier and the lasso for essentially all n. Note that the incoherent classifier also admits better performance than the lasso. This is expected; although they are very similar, the incoherent classifier was derived specifically for this joint graph/class model. For comparison purposes, the naïve Bayes plugin classifier; that is, the classifier that assumes the whole graph is the signal subgraph, is also shown (black dashed ~~light gray~~ line). Note that the performance of all the classifiers is bounded above by $L_{\hat{\boldsymbol{\pi}}} = 0.5$ and below by $L_* = 0.13$. Moreover, $\widehat{L}_{nb} > \widehat{L}_{inc} > \widehat{L}_{coh}$ for essentially all n.

An important aspect of any algorithm is compute time, both of training and testing. The signal subgraph classifiers that we developed are very fast. Computations essentially amount to computing a test-statistic for all $|\mathcal{E}|$ edges, then sorting them. The parameter estimates of the likelihood and prior terms come directly from the same test-statistics used to obtain the significance of each edge. Thus, obtaining those estimates amounts to essentially computing a mean. On the other hand, the lasso classifier, which yields *worse* signal detection and misclassification rates than both our classifiers, requires an iterative algorithm for each value on the hyper-parameter path [19]. Despite

that efficient computational schemes have been developed for searching the whole regularization path [20], such iterative algorithms should be much slower than our classifiers.

Indeed, the lower panel of Figure 3 demonstrates that our MATLAB implementation of the signal subgraph classifiers are approximately 10 times faster than MATLAB's lasso implementation. All the results shown in Figure 3 include errorbars computed from 100 trials, each with 100 held-out samples, demonstrating that for these simulation parameters, the differences are highly significant. Although the quantitative results may vary for different implementations and different parameter settings, our expectation is that the qualitative results should be consistent. Thus, because our classifiers have lower risk, better signal identification, and run an order of magnitude faster than the standard, we do not consider lasso in further simulations.

The above numerical results suggest that the coherent estimator achieves better signal subgraph identification and classification performance than ~~outperforms~~ the incoherent estimator almost always, despite that the computational time of the coherent classifier is almost identical. However, that result is a function of both the model $\mathcal{M}_V$ (which includes the number of vertices), and the number of training samples n (there is a bias-variance trade-off here, as always). Figure 4 explicitly shows that the relative performance of an estimator for a particular model ($\mathcal{M}_{30}(1, 5; 0.5, 0.1, 0.2)$ here) changes as a function of the number of samples. More specifically, for small n, the incoherent estimator yields better performance, as indicated by the relative rate and relative efficiency being above one. However, with more samples, when the signal subgraph is coherent, the coherent estimator will eventually outperform the incoherent one. At infinite samples, since both estimators are consistent, they will yield identical results: the truth.

Thus, to choose which estimator will likely achieve the best performance, knowledge of the model, $\mathcal{M}_V(m, s; \pi, p, q)$, is insufficient; rather, both the model and the number of samples must be known a priori.

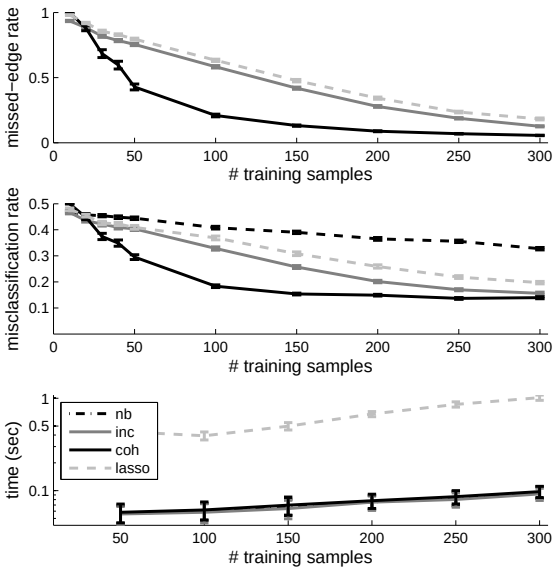

Fig. 3. Performance statistics as a function of sample size demonstrate that the coherent signal subgraph estimator outperforms the incoherent signal subgraph estimator, in terms of both the signal subgraph identification and classification, for nearly all n, using the same model as in Figure 2: $\mathcal{M}_{70}(1, 20; 0.5, 0.1, 0.3)$. Moreover, even the incoherent classifier outperforms the ℓ_1-penalized logistic regression (lasso) on all our metrics. The top panel shows the missed-edge rate for each estimator as a function of the number of training samples, n. The middle ~~bottom~~ panel shows the corresponding misclassification rate for the estimators, as well as the naïve Bayes plugin classifier. Performance of all estimators improves (nearly) monotonically with n for both criteria. The bottom panel shows total training and testing time for each classifier. Clearly, the lasso is about 10 times slower than the others. Error bars show standard error of the mean here and elsewhere unless otherwise noted (averaged over 100 trials; each trial used 100 samples for held-out data). Error bars on the lower panel show the inter-quartile range. Note that for most values of n, we have $L_{\hat{\pi}} > \widehat{L}_{nb} > \widehat{L}_{lasso} > \widehat{L}_{inc} > \widehat{L}_{coh} > L_*$. Legend: "inc": incoherent; "coh": coherent; "nb": naïve Bayes, "lasso": lasso.

4.4 Estimating the hyper-parameters

In the above analyses the hyper-parameters, both the number of signal edges s and signal vertices m, were known. In practice while one might have a preliminary guess of the range of these hyper-parameters, the optimal values will usually be unknown. We can therefore use a cross-validation technique to search over the space of all reasonable combinations of s and m, and choose the best performing combination. Figure 5 shows one such simulation depicting several key features. The top panel shows the misclassification rate on held-out data as a function of the log of the assumed size of the signal subgraph for the incoherent classifier. Although the true size is $s = 20$, the best performing estimate is $\widehat{s}_{inc} = 23$. This is a relatively standard result in model selection: the best performer will include a few extra dimensions because adding a few uninformative features is less costly than missing a few informative features [21]. This intuition is further reified by the U-shape of the misclassification curve on a log scale: including many non-signal edges is less detrimental than excluding a few signal edges.

The bottom panel shows the coherent performance

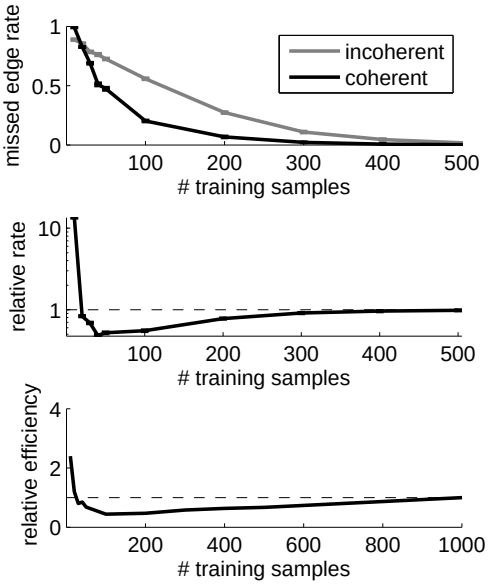

Fig. 4. The relative performance of the coherent and incoherent estimators is a function not just of the model, but also the number of training samples. Specifically, for the same model, $\mathcal{M}_{70}(1, 20; 0.5, 0.1, 0.3)$, we compute the missed-edge rates for both the incoherent estimator (gray line) and the coherent estimator (black line), averaged over 200 trials. The top panel shows that for small training sample size the incoherent estimator achieves a better (lower) missed-edge rate than the coherent estimator. However, the incoherent estimator's convergence rate is slower, and the coherent estimator catches up and outperforms the incoherent estimator until both eventually converge at the truth. The middle and bottom panels show the relative rate and efficiency curves for this model. Note that the curves dip below unity, and then converge to unity, as they must, because both estimators are consistent.

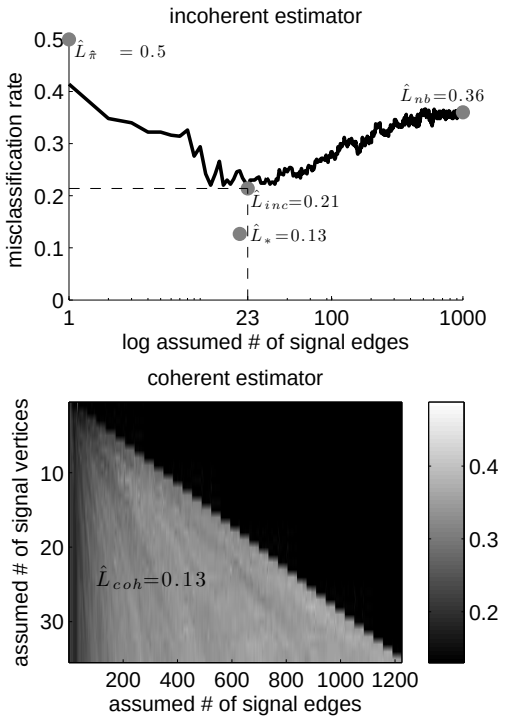

Fig. 5. When constraints on the number of signal edges (s) or signal vertices (m) are unknown, a search over these hyperparameters can yield estimates $\widehat{s}$ and $\widehat{m}$. Both panels depict held-out cross-validation error as a function of varying these parameters for the model $\mathcal{M}_{70}(1, 20; 0.5, 0.1, 0.3)$ (the same as in Figures 2 and 3), using 200 training samples and 500 test samples, with $m = 1$ and $s = 20$. The top panel depicts misclassification rate of the incoherent estimator as a function of the number of estimated signal edges on a log scale, with the best performing classifier achieving $\widehat{L}_{inc} = 0.21$. Note that in this simulation, $s = 20 < \widehat{s}_{inc} = 23$. This "conservatism" is typical and appropriate in many model selection situations. The bottom panel shows $\widehat{L}_{coh}$ as a function of both m' and s'. For this simulation, $\widehat{m}_{coh} = 1$ and $\widehat{s}_{coh} = 24$, further corroborating the conservative stance on model selection. Note that $L_{\boldsymbol{\pi}} > \widehat{L}_{nb} > \widehat{L}_{inc} > \widehat{L}_{coh} \geq L_*$ as one would hope for this coherent simulation. Incidentally, the coherent classifier achieved Bayes error here, $L_* = 0.13$.

by varying both m and s, which exhibits a "banded" structure, indicating that the performance is relatively robust to small changes in m. This banding likely results from the fact that the test statistics are identical for many edges, so therefore minor changes in the number of allowable edges is not expected to change performance much. Moreover, the best performing pair achieved $\widehat{L}_{coh} = 0.13$ (which is equal to the Bayes error) with $\widehat{m}_{coh} = 1$ and $\widehat{s}_{coh} = 24$, suggesting that n was sufficiently large to correctly find the true signal vertex, and further corroborating the "better safe than sorry" attitude to selecting the signal vertices.

5 MR Connectome Sex Classification

A connectome is brain-graph [22]. MR connectomes utilize multi-modal Magnetic Resonance (MR) imaging to determine both the vertex and edge set for each individual [2]. This section investigates the utility of the classifiers developed above on data collected for the Baltimore Longitudinal Study of Aging, as described previously [23]. Briefly, 49 subjects (25 male, 24 female) underwent a diffusion-weighted MRI protocol. The Magnetic Resonance Connectome Automated Pipeline (Mr. Cap) layout was used to convert each subject's raw multi-modal MR data into a connectomes [24] (each connectome is a simple graph with 70 vertices and up to $\binom{70}{2} = 2415$ edges). Lacking strong priors on either the number of signal edges or signal

vertices in the signal subgraph (or even whether a signal subgraph exists), we searched over a large space of hyper-parameters using leave-one-out cross-validated misclassification performance as our metric of success (Figure 6). The naïve Bayes classifier—which assumes the signal subgraph is the whole edge set, $\widehat{\mathcal{S}}_{nb} = \mathcal{E}$—performs marginally better than chance: $\widehat{L}_{nb} = 0.41$ (p-value ≈ 0.05 assessed by a permutation test). With a relatively small number of incoherent edges—$\widehat{s}_{inc} = 10$—the incoherent classifier (top left panel) achieves $\widehat{L}_{inc} = 0.27$, significantly better than chance (p-value < 0.0007), but not significantly better than the naïve Bayes classifier (using McNemar's test). The coherent classifier achieved a minimum of $\widehat{L}_{coh} = 0.16$ (top right and middle panels), not significantly better than the incoherent classifier, but significantly better than both chance and the naïve Bayes classifier (p-values $< 10^{-5}$ and < 0.004, respectively). This improved performance upon using the coherent classifier suggests that the signal subgraph is at least approximately coherent. Using $\widehat{m}_{coh} = 12$ and $\widehat{s}_{coh} = 360$ from the best performing coherent classifier, we can estimate the signal subgraph (bottom left). The coherogram suggests that indeed, the signal is somewhat, but not entirely coherent (bottom right).

We next compare the performance of our classifiers on this MR connectome sex classification data set to several others. First, a standard parametric classifier: lasso. We chose the regularization parameter via a 10-fold cross-validation. Second, a non-parametric (distribution free) classifier: k_n-nearest neighbor (kNN) which operates directly on graphs [25]. This kNN classifier uses the Frobenius norm distance metric. We tried all $k \in [n]$ and simply report the best performance. The universal consistency of this kNN classifier is useful in assessing the algorithm complexity supported by this data. In particular, given enough samples, kNN will achieve optimal performance. Less than optimal performance therefore indicates that the sample size is not sufficiently large for this kNN classifier. Third, a graph invariant based classifier. We computed six graph invariants for each graph: size, max degree, scan statistic, number of triangles, clustering coefficient, and average path length [26], normalized each to have zero mean and unit variance, and then used a kNN with ℓ_2 distance metric on the invariants.

Despite the small sample size, Table 1 demonstrates that the coherent classifier is significantly better than all the others (except the incoherent classifier), as assessed via a one-sided McNemar's test. Note that although lasso and the incoherent classifier achieved the same misclassification rate, they erred on different subjects, suggesting that further improvements in performance might be possible from combining these two classifiers. We do not pursue that option here.

TABLE 1

Bake-off comparing a number of different classifiers on the MR connectome sex classification data. Error indicates misclassification error using the best hyper-parameters found for each classifier. P-value indicates the p-value of a one-sided McNemar's test comparing each classifier to the coherent classifier. The coherent classifier is significantly better than all the others, except the incoherent classifier.

classifier	error	p-val
prior	0.50	< 0.01
naïve Bayes	0.41	< 0.01
lasso	0.27	< 0.02
graph-kNN	0.35	< 0.02
invariant-kNN	0.43	< 0.01
incoherent	0.27	> 0.05
coherent	**0.16**	n\a

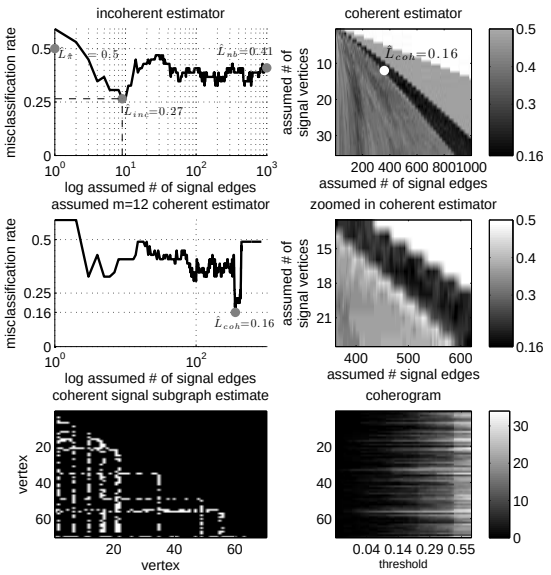

Fig. 6. MR connectome sex signal subgraph estimation and analysis. By cross-validating over hyper-parameters and models, we estimate that the "best" incoherent signal subgraph (for this inference task on these data) has $\widehat{s}_{inc} = 10$ and yields a misclassification rate of $\widehat{L}_{inc} = 0.27$, whereas the best coherent signal subgraph has $\widehat{m}_{coh} = 12$ and $\widehat{s}_{coh} = 360$, achieving $\widehat{L}_{coh} = 0.16$. The top two panels depict the same information as Figure 5. The middle two depict misclassification rate (left) for different choices of $m' = 12$ as a function of s' and (right) a zoomed-in depiction of the top right panel. The bottom left panel shows the estimated signal subgraph, and the bottom right shows the coherogram. Together, these bottom panels suggest that the signal subgraph for these data is at least somewhat coherent.

5.1 Model Evaluation

Although the coherent classifier employed above did estimate a signal subgraph, we investigate to what extent the estimated signal subgraph represents the true signal subgraph. We address this question in two ways: (i) synthetic data analysis and (ii) assumption checking.

5.1.1 *Synthetic data analysis*

For the synthetic data analysis, we generated data as follows. Given the above estimated signal subgraph, for every edge not in $\widehat{\mathcal{S}}_{coh}$, let $p_{uv|0} = p_{uv|1} = \widehat{p}_{uv}$, where $\widehat{p}_{uv}$ is the estimated edge probability averaging over all samples. For all edges in $\widehat{\mathcal{S}}_{coh}$, let $p_{uv|y} = \widehat{p}_{uv|y}$. Set the priors according to the data as well: $\pi = \widehat{\pi}$.

Given this synthetic data model, we first sampled 49 data samples, 25 from class 0 and 24 from class 1, and estimated the incoherent and coherent classifier performance on a single synthetic experiment (Figure 7, top panels). The performance of the classifiers on the synthetic data qualitatively mirrors that of the real data, suggesting some degree of model appropriateness. To assess what fraction of the edges in the estimated signal subgraph were reliable, even assuming a true model, we then sampled up to 100 training samples (and 100 test samples), and computed the missed-edge rate (bottom left) and misclassification rate (bottom right) as a function of the number of samples. Given approximately 50 samples, the incoherent signal subgraph estimator correctly identifies about 40% of the edges, whereas the coherent signal subgraph estimator correctly identifies about 50%. This suggests that even if the model were true (we think it is not) we should believe that only about half the edges in the estimated signal subgraph are in the actual signal subgraph. Despite our stated desideratum of interpretability of the resulting classifier in terms of correctly identifying the signal edges and vertices, for data sampled from this assumed distribution, sample sizes of < 50 seem to be insufficient. That said, ~~Moreover,~~ both missed-edge rate and misclassification rate exhibit a step-like function in performance: after about 50 samples, performance dramatically improves. This suggests that perhaps only a few more data points would be necessary to obtain greatly improved classification accuracy.

5.1.2 *Model checking*

The assumption of independence between edges is (i) very useful for algorithms and analysis, and (ii) almost certainly nonsense for real connectome data. Checking whether edges are independent is relatively easy. Figure 8 shows the correlation coefficient between all pairs of edges in the estimated signal subgraph from the neurobiological data. We used a spectral clustering algorithm [27] to more clearly highlight any significant correlations. Several groups of edges seem to be highly correlated. To assess significance, we compare the distribution of correlation coefficients with the distribution of correlation coefficients obtained from the synthetic data analysis. A two-sample Kolmogorov-Smirnov test shows that the two matrices are significantly different (p-value ≈ 0), rejecting the null hypothesis that the edges in the real data are independent. This analysis further corroborates that making independence assumptions can be fruitful even when the data are dependent [18].

6 Discussion

This work makes the following contributions. First, it introduces a novel graph/class model that admits rigorous statistical investigation. Moreover, it presents two approaches for estimating the signal subgraph: the first using only vertex label information, the second also utilizing graph structure. The resulting estimators have desirable asymptotic and finite sample properties, including consistency and robustness to various model misspecifications. Third, simulated data analysis indicate that neither approach dominates the other; rather, the best approach is a function of both the model and the amount of training data. And while the lasso classifier has similar error properties to our incoherent classifier, lasso's computational time is about an order of magnitude longer. Fourth, these classifiers are applied to an MR connectome sex classification ~~a connectome~~ data set; the coherent classifier performs significantly better than ~~both naïve Bayes and incoherent classifiers~~ a variety of benchmark classifiers. Fifth, synthetic data analysis suggests that while we can use the signal subgraph estimators to improve classification performance, we should not expect that all the edges in the estimated signal subgraph will be the true signal edges, even when the model is correct. Moreover, we might expect a drastic improvement in classification performance with only a few additional data samples. Finally, model checking suggests that the independent edge assumption does not fit the data well.

Collectively, the above analyses suggest a number of possible next steps. First, collect more data. Second, relax various assumptions, including (i) the independent edge assumption by considering conditionally independent edges [28]–[30], (ii) binary edge assumption, and (iii) labeled vertices assumption. Third, transform a number of conjectures that have arisen due to these results into theorems. For instance, perhaps the misclassification rate is a monotonic function of the missed-edge rate. Fourth, (Bayesian) model-averaging to combine estimated signal subgraphs instead of picking one might improve performance (perhaps at the cost of computational resources and interpretability). Fifth, extension to situations for which none of the vertices are labeled [31], [32], only some

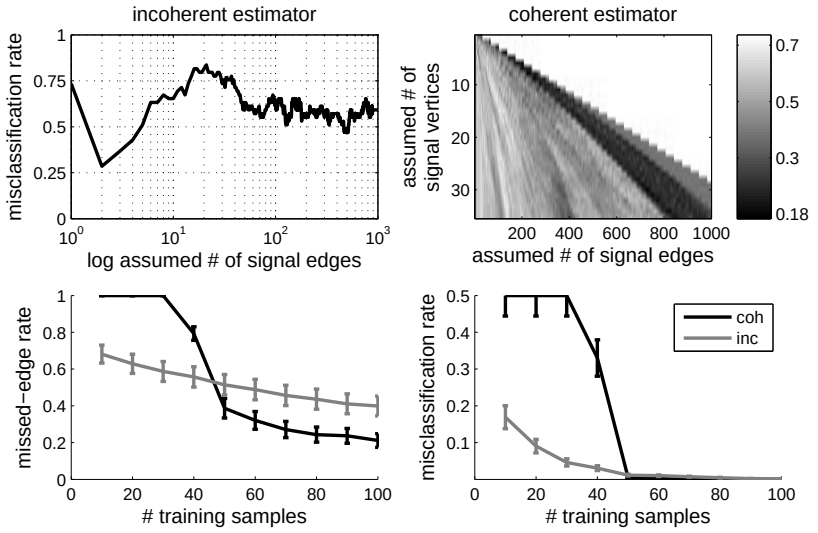

Fig. 7. Synthetic data analysis provides some intuition for model checking and future improvements. The top two panels show the incoherent (left) and coherent (right) misclassification rates as a function of the hyper-parameter choices for $n = 49$. These plots look quite similar to those obtained in the real connectome data (Figure 6), which suggests that the chosen model may be adequate. The bottom panels show the missed-edge rate (left) and misclassification rate (right) as a function of the number of training samples. With about 50 training samples, approximately half of the edges identified by each classifier are true edges. Additionally, slightly more than 50 training samples seems to be sufficient for obtaining nearly perfect classification, suggesting that perhaps only a few more subjects would be sufficient to yield much greater classification performance.

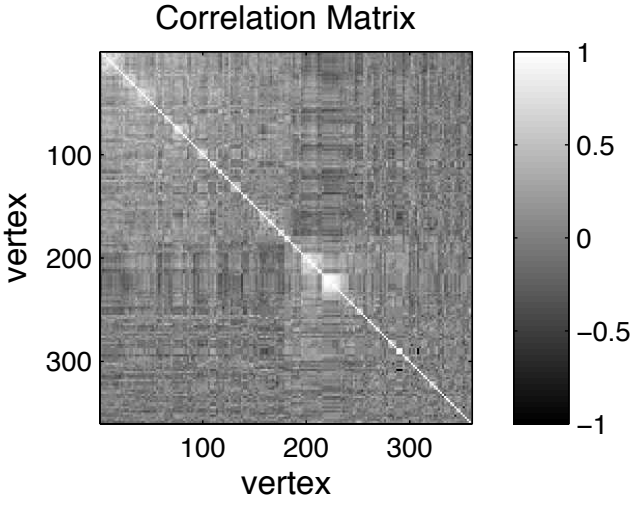

Fig. 8. The correlation matrix between all the edges in the coherent signal subgraph estimate. Edges are organized by co-clustering to highlight any similarities. Although most edges are uncorrelated, several groups of edges cluster, indicative of the fact that the edges are not independent (p-value of ≈ 0 using a two-sample Kolmogorov-Smirnov test comparing the real and synthetic correlation matrices).

subset of vertices are labeled [33], [34], or data are otherwise errorfully observed [35], are all avenues of future investigation.

We hope the proposed approaches will yield many applications. To that end, all the data and code used in this work is available from the author's website, http://jovo.me.

ACKNOWLEDGMENTS

This work was partially supported by the Research Program in Applied Neuroscience.

REFERENCES

[1] H. Bunke and K. Riesen, "Towards the Unification of Structural and Statistical Pattern Recognition," *Pattern Recognition Letters*, vol. 33, pp. 811–825, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865511001309

[2] P. Hagmann, L. Cammoun, X. Gigandet, S. Gerhard, P. Ellen Grant, V. Wedeen, R. Meuli, J.-P. Thiran, C. J. Honey, and O. Sporns, "MR connectomics: Principles and challenges." *Journal of neuroscience methods*, Jan. 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20096730

[3] D. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 6–18, Jan. 2006. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1564423

[4] T. Kudo, "An Application of Boosting to Graph Classification," *Science*.

[5] N. S. Ketkar, L. B. Holder, and D. J. Cook, "Empirical comparison of graph classification algorithms," *2009 IEEE Symposium on Computational Intelligence and Data Mining*, pp. 259–266, Mar. 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4938658

[6] G. North, *Invertebrate Neurobiology*. CSHL Press, 2007. [Online]. Available: http://books.google.com/books?id=6-il0L4vATAC\&pgis=1

[7] J. D. Shepherd and R. L. Huganir, "The cell biology of synaptic plasticity: AMPA receptor trafficking." *Annual review of cell and developmental biology*, vol. 23, pp. 613–43, Jan. 2007. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17506699

[8] J. Nolte, *The Human Brain: An Introduction to Its Functional Anatomy*. Mosby, 2002. [Online]. Available: http://www.amazon.com/The-Human-Brain-Introduction-Functional/dp/0323013201

[9] J. W. Lichtman, J. Livet, and J. R. Sanes, "A technicolour approach to the connectome." *Nat Rev Neurosci*, vol. 9, no. 6, pp. 417–422, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1038/nrn2391

[10] D. S. Bassett and E. T. Bullmore, "Human brain networks in health and disease," *Current Opinion in Neurology*, vol. 22, no. 4, pp. 340–347, 2009. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2902726\&tool=pmcentrez\&rendertype=abstract

[11] J. Lasserre, C. Bishop, and T. Minka, "Principled Hybrids of Generative and Discriminative Models," *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*, vol. 1, no. 6, pp. 87–94, 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1640745

[12] P. J. Bickel and K. A. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics, Vol I (2nd Edition)*. Prentice Hall, 2000. [Online]. Available: http://www.amazon.com/Mathematical-Statistics-Basic-Selected-Topics/dp/013850363X

[13] C. M. Stein, "Inadmissibility of the usual estimator of the mean of a multivariate normal distribution," in *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1. University of California Press, 1956, pp. 197–206.

[14] P. I. Good, *Permutation, Parametric, and Bootstrap Tests of Hypotheses (Springer Series in Statistics)*. Springer, 2010. [Online]. Available: http://www.amazon.com/Permutation-Parametric-Bootstrap-Hypotheses-Statistics/dp/1441919074

[15] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, Jun. 1947. [Online]. Available: http://www.springerlink.com/content/843g84t135765212/

[16] P. J. Huber, *Robust Statistics*, ser. Wiley Series in Probability and Statistics. Wiley, 1981, vol. 82, no. 3. [Online]. Available: http://doi.wiley.com/10.1002/9780470434697

[17] J. A. Rice, *Mathematical statistics and data analysis*. Duxbury Press, 1995. [Online]. Available: http://www.citeulike.org/user/tarjeiha/article/1691927

[18] D. J. Hand and K. Yu, "Idiot's Bayes: Not So Stupid after All?" *International Journal Statistical Review*, vol. 69, no. 3, pp. 385–398, Nov. 2001. [Online]. Available: http://www.jstor.org/pss/1403452

[19] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B*, vol. 58, pp. 267–288, 1996.

[20] B. Y. B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "LEAST ANGLE REGRESSION ear regression , will be discussed here , and motivated in terms of a computationally simpler method called Least Angle Regression . Least Angle Regression (LARS) relates to the classic model-selection method known as Forward Sele," *Annals of Statistics*, vol. 32, no. 2, pp. 407–451, 2004.

[21] A. K. Jain, R. P. W. Duin, J. Mao, and S. Member, "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=824819

[22] O. Sporns, *Networks of the Brain*. The MIT Press, 2010. [Online]. Available: http://www.amazon.com/Networks-Brain-Olaf-Sporns/dp/0262014696

[23] J. T. Vogelstein, W. R. Gray, J. L. Prince, L. Ferrucci, S. M. Resnick, C. E. Priebe, and R. J. Vogelstein, "Graph-Theoretical Methods for Statistical Inference on MR Connectome Data," *Organization Human Brain Mapping*, 2010.

[24] W. R. Gray, J. A. Bogovic, J. T. Vogelstein, B. A. Landman, J. L. Prince, and R. J. Vogelstein, "Magnetic Resonance Connectome Automated Pipeline," *submitted*, 2010.

[25] J. T. Vogelstein, R. J. Vogelstein, and C. E. Priebe, "Are mental properties supervenient on brain properties?" *Nature Scientific Reports*, vol. in press, p. 11, 2011. [Online]. Available: http://arxiv.org/abs/0912.1672

[26] C. E. Priebe, G. A. Coppersmith, and A. Rukhin, "You say graph invariant, I say test statistic," *Statistical Computing Statistical Graphics Newsletter*, vol. 21, no. 2, pp. 11–14, 2010.

[27] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining KDD 01*, vol. pages, no. April 2006, pp. 269–274, 2001. [Online]. Available: http://portal.acm.org/citation.cfm?doid=502512.502550

[28] P. D. Hoff, A. E. Raftery, and M. S. Handcock, "Latent Space Approaches to Social Network Analysis," *Journal of the American Statistical Association*, vol. 97, no. 460, pp. 1090–1098, 2002.

[29] D. L. Sussman, M. Tang, D. E. Fishkind, and C. E. Priebe, "A consistent adjacency spectral embedding for stochastic blockmodel graphs," *submitted for publication*, no. 1108 . 2228v3, p. 17, 2011. [Online]. Available: http://arxiv.org/abs/1108.2228

[30] D. E. Fishkind, D. L. Sussman, M. Tang, J. T. Vogelstein, and C. E. Priebe, "Consistent adjacency-spectral partitioning for the stochastic block model when the model parameters are unknown," *submitted for publication*, p. 20, May 2012. [Online]. Available: http://arxiv.org/abs/1205.0309

[31] J. T. Vogelstein, J. C. Conroy, L. J. Podrazik, S. G. Kratzer, D. E. Fishkind, R. J. Vogelstein, and C. E. Priebe, "Fast Inexact Graph Mathing with Applications in Statistical Connectomics," *Submitted to IEEE PAMI*, 2011.

[32] J. T. Vogelstein and C. E. Priebe, "Shuffled Graph Classification: Theory and Connectome Applications," *Submitted to IEEE PAMI*, 2011.

[33] G. A. Coppersmith and C. E. Priebe, "Vertex Nomination via Content and Context," *Technology*, pp. 1–21, 2012.

[34] D. S. Lee and C. E. Priebe, "Bayesian Vertex Nomination," *submitted for publication*, no. i, 2012.

[35] C. E. Priebe, J. T. Vogelstein, and D. D. Bock, "Optimizing the quantity/quality trade-off in connectome inference," *Communications in Statistics Theory and Methods*, p. 7, 2011. [Online]. Available: http://arxiv.org/abs/1108.6271

Joshua T. Vogelstein Joshua T. Vogelstein received the B.S degree in biomedical engineering from Washington University in St. Louis, MO in 2002, the M.S. degree in applied mathematics and statistics from Johns Hopkins University (JHU) in Baltimore, MD in 2009, and the Ph.D. degree in neuroscience from Johns Hopkins School of Medicine in Baltimore, MD in 2009. He is currently a postdoctoral fellow in the Department of Applied Mathematics and Statistics at JHU, with a joint appointment in the Human Language Technology Center of Excellence. His research interests primarily include statistical connectomics, including theory and applications for high-dimensional graph-valued data. His research has been featured in a number of prominent scientific and engineering journals including Annals of Applied Statistics, IEEE Transactions on Neural Systems and Rehabilitation Engineering, and Nature Neuroscience.

William R. Gray William R. Gray graduated from Vanderbilt University in 2003 with a Bachelors degree in electrical engineering, and received his MS in electrical engineering in 2005 from the University of Southern California. Currently, Will is a PhD student in electrical engineering at Johns Hopkins University, where he is conducting research in the areas of connectivity, signal and image processing, and machine learning. He is also a member of the technical staff at the Johns Hopkins University Applied Physics Laboratory, where he manages projects in the Biomedicine and Undersea Warfare business areas. Will is a member of IEEE, Eta Kappa Nu, and Tau Beta Pi

R. Jacob Vogelstein R. Jacob Vogelstein received the Sc.B. degree in neuroengineering from Brown University, Providence, RI, and the Ph.D. degree in biomedical engineering from the Johns Hopkins University School of Medicine, Baltimore, MD. He currently oversees the Applied Neuroscience programs at the Johns Hopkins University (JHU) Applied Physics Laboratory as an Assistant Program Manager, and has an appointment as an Assistant Research Professor at the JHU Whiting School of Engineerings Department of Electrical and Computer Engineering. He has worked on neuroscience technology for over a decade, focusing primarily on neuromorphic systems and closed-loop brainmachine interfaces. His research has been featured in a number of prominent scientific and engineering journals including the IEEE Transactions on Neural Systems and Rehabilitation Engineering, the IEEE Transactions on Biomedical Circuits and Systems, and the IEEE Transactions on Neural Networks.

Carey E. Priebe Carey E. Priebe received the B.S. degree in mathematics from Purdue University in 1984, the M.S. degree in computer science from San Diego State University in 1988, and the Ph.D. degree in information technology (computational statistics) from George Mason University in 1993. From 1985 to 1994 he worked as a mathematician and scientist in the US Navy research and development laboratory system. Since 1994 he has been a professor in the Department of Applied Mathematics and Statistics, Whiting School of Engineering, Johns Hopkins University, Baltimore, Maryland. At Johns Hopkins, he holds joint appointments in the Department of Computer Science, the Department of Electrical and Computer Engineering, the Center for Imaging Science, the Human Language Technology Center of Excellence, and the Whitaker Biomedical Engineering Institute. He is a past President of the Interface Foundation of North America - Computing Science & Statistics, a past Chair of the American Statistical Association Section on Statistical Computing, a past Vice President of the International Association for Statistical Computing, and on the editorial boards of Journal of Computational and Graphical Statistics, Computational Statistics and Data Analysis, and Computational Statistics. His research interests include computational statistics, kernel and mixture estimates, statistical pattern recognition, statistical image analysis, dimensionality reduction, model selection, and statistical inference for high-dimensional and graph data. He is a Senior Member of the IEEE, a Lifetime Member of the Institute of Mathematical Statistics, an Elected Member of the International Statistical Institute, and a Fellow of the American Statistical Association.