Graph Classification using Signal Subgraphs

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

Abstract—The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.

The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.The abstract goes here.

Index Terms—Computer Society, IEEEtran, journal, LaTeX, paper, template.

1 Introduction

Graphs are becoming increasingly popular vehicles for data representation, spanning fields from optical character recognition [], to chemistry [], to neuroscience []. While statistical inference techniques for vector-valued data are widespread, statistical tools for the analysis of graph-valued data are relatively rare []. In this work we propose and analyze a relatively simple yet rich random graph model—sufficiently simple to characterize its asymptotic properties, and sufficiently rich to afford useful empirical applications. For concreteness, we consider the task of *graph classification*: given a collection of graphs and their corresponding classes, can we accurately estimate the class of a new graph lacking a class label? Our approach is statistical in nature. We first define a graph/class model. This model admits that the class-conditional signal is encoded in a subset of edges, the *signal subgraph*. Finding the signal subgraph amounts to providing an understanding of what are the differences between the two classes of graphs. This approach is quite different from quantifying a number of graph invariants (such as clustering coefficient), and building a classifier in a graph invariant space. Specifically, in the proposed model-based approach, we use both (i) the vertex labels and (ii) graph structure to help classify. To our knowledge, this is the first proposed strategy using both of those cues. We demonstrate via proofs, simulation, and a neurobiological data set (MR connectomes []) that ignoring one or the other of these cues significantly degrades one's ability to classify.

2 Methods

2.1 Setting

Let $\mathbb{G} : \Omega \mapsto \mathcal{G}$ be a graph-valued random variable with samples G_i. Each graph 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 $\mathcal{E}$, where $|\mathcal{E}| \leq V^2$. An adjacency matrix, A, is a binary $V \times V$ matrix listing which vertices share an edge. Let $Y : \Omega \mapsto \mathcal{Y}$ 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 (or exploitation task) is then to build a graph classifier that could take a new graph, 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, manifold learning, feature extraction, and related approaches are inadmissible.

2.2 Model

A model defines the set of admissible distributions. 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: $\mathcal{F}_{\mathbb{G},Y} = \{F_{\mathbb{G},Y} : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, where $\boldsymbol{\theta}$ indexes each distribution, and must live in $\boldsymbol{\Theta}$. Two "standard" approaches for tackling a classification problem are (i) the *generative* approach and (ii) the *discriminative* approach. In a generative strategy, one deconstructs 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 deconstructs the joint distribution into a posterior term and a marginal term: $F_{\mathbb{G},Y} = F_{Y|\mathbb{G}} F_{\mathbb{G}}$. We proceed using a hybrid generative-discriminative approach whereby we describe a generative model and place constraints on the discriminant boundary.

- *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@jhu.edu
- *W.R. Gray and R.J. Vogelstein are with the Johns Hopkins University Applied Physics Laboratory, Laurel, MD, 20723.*

First, assume that each graph has the same set of 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: $F_{A,Y} = \prod_{u,v\in\mathcal{E}} F_{A_{uv},Y}$. Now, consider the generative deconstruction, and let $F[A_{uv}|Y = y] = f_{uv|y}$ and $F_Y = \pi_Y$, noting that $F_{\mathbb{G},Y} = f_Y \pi_Y$. 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 that condition comprise the *signal subgraph*, $\mathcal{S} = \{(u,v) \in \mathcal{E} : f_{uv|0} \neq f_{uv|1}\}$. Fourth, for concreteness, assume that the graphs are *simple* graphs, that is, undirected, with binary edges, and lacking (self-) loops. Thus, the likelihood of an edge between vertex u and v is given by a Bernoulli random variable with a scalar probability parameter: $f_{uv|y} = \text{Bern}(A_{uv}; p_{uv|y})$. Together, these four assumptions imply the following model:

$$\mathcal{F}_{\mathbb{G},Y} = \{F_{\mathbb{G},Y} = \prod_{uv\in\mathcal{S}} \text{Bern}(A_{uv|Y}; p_{uv|Y})\pi_Y : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}, \tag{1}$$

where θ is composed of three parameters: $\mathcal{S}$, $\boldsymbol{p}$, and $\boldsymbol{\pi}$, each of which lives in a parameter space. First, the signal subgraph parameter, $\mathcal{S}$, must be a non-empty subset of all possible edges: $\mathcal{S} \subseteq \mathcal{E} \cap \mathcal{S} \neq \emptyset$. Second, the likelihood term parameter, $\boldsymbol{p} = \{p_{uv|y}\}_{uv\in\mathcal{S}, y\in\mathcal{Y}}$, is constrained in that each term must be between zero and one: $p_{uv|y} \in (0,1)$. Third, the prior terms, $\boldsymbol{\pi} = \{\pi_y\}$, must be greater than zero and sum to one: $\pi_y \geq 0, \sum_y \pi_y = 1$. Thus, given a specification of the signal subgraph, the class-conditional likelihood of an edge in each the signal subgraph, and class-priors, one completely defines a possible joint distribution over graphs and classes.

2.3 Classifier

Formally, we say that a graph classifier, h, is any function satisfying $h : \mathcal{G} \mapsto \mathcal{Y}$. We desire to obtain the best possible classifier, h_*. To determine which is best, we first define a loss function, which rates the performance of each classifier as a function of the distribution: $\ell : \mathcal{F} \times \mathcal{H} \mapsto \mathbb{R}_+$, where $\mathcal{H}$ is the space of admissible classifiers. We choose to measure classification performance by the expected misclassification rate:

$$\ell_F(h) = \mathbb{E}_F[h(G) \neq Y] = \int F[h(g) \neq y] F[g,y] dg dy. \tag{2}$$

The best classifier (under model $\mathcal{F}$ and loss-function ℓ) is the classifier with minimal loss: $h_* = \text{argmin}_{h\in\mathcal{H}} \ell_F(h)$. Such a classifier is called *Bayes optimal*, and the error associated with such a classifier is called *Bayes error* or *Bayes risk*. It can be shown that the classifier that maximizes the class-conditional posterior, $F_{Y|\mathbb{G}}$ is Bayes optimal [1]:

$$\begin{aligned} h_*(g) &= \underset{h\in\mathcal{H}}{\text{argmin}}\, \ell_F(h) = \underset{y\in\mathcal{Y}}{\text{argmax}}\, F[y|g] \\ &= \underset{y\in\mathcal{Y}}{\text{argmax}}\, F[g|y]F[y]. \end{aligned} \tag{3}$$

Given the proposed model, Eq. (3) can be further factorized using the above four assumptions:

$$\underset{y\in\mathcal{Y}}{\text{argmax}}\, F[g|y]F[y] = \underset{y\in\mathcal{Y}}{\text{argmax}} \prod_{u,v\in\mathcal{S}} \text{Bern}(a_{uv}; p_{uv|y})\pi_y. \tag{4}$$

Unfortunately Bayes optimal classifiers are typically unavailable. In such settings, it is therefore desirable to construct a classifier estimate from a set of *training data*. Formally, let $\mathcal{D}_n$ denote the data corpus, assumed to be sampled exchangeably from the true but unknown distribution: $\mathcal{D}_n = \{(\mathbb{G}_i, Y_i)\}_{i\in[n]} \overset{exch.}{\sim} F_{\mathbb{G},Y}$. Given such a training corpus, and a new, as yet unclassified graph, g, an estimated classifier predicts the true (but unknown) class of g by utilizing the training corpus: $\hat{h}_n : \mathcal{G} \times (\mathcal{G} \times \mathcal{Y})^n \mapsto \mathcal{Y}$. When a model, $\mathcal{F}_{\mathbb{G},Y}$ is specified, a beloved approach is to use a *Bayes plug-in classifier*. Due to the above simplifying assumptions, the Bayes plug-in classifier for this model is defined as follows. First, estimate the three model parameters (1) $\mathcal{S}$, (2) $\boldsymbol{p} = \{p_{uv|y}\}_{uv\in\hat{\mathcal{S}}, y\in\mathcal{Y}}$, and (3) $\boldsymbol{\pi} = \{\pi_y\}$. Second, plug those estimates into the above equation. The result is a Bayes plug-in graph classifier:

$$\hat{h}_n(g) \overset{\triangle}{=} \underset{y\in\mathcal{Y}}{\text{argmax}} \prod_{u,v\in\hat{\mathcal{S}}} \hat{p}_{uv|y}^{a_{uv}} (1 - \hat{p}_{uv|y})^{(1-a_{uv})} \hat{\pi}_y, \tag{5}$$

where the Bernoulli probability is explicit. To implement such a classifier estimate, we require estimators for estimating the above three estimands.

2.4 Estimators

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

- **Consistent**: an estimator is consistent if its sequence converges in the limit to the true value: $\lim_{n\to\infty} \hat{\boldsymbol{\theta}}_n = \boldsymbol{\theta}$.
- **Efficient**: an estimator is efficient if its sequence converges to the minimum variance: $\lim_{n\to\infty} \text{Var}(\hat{\boldsymbol{\theta}}_n) = \mathcal{I}_{\boldsymbol{\theta}}^{-1}$. A maximally efficient estimator yields an estimate with *minimum variance*.
- **Robust**: an estimator is robust if the resulting estimate is relatively insensitive to small model misspecifications. Because the space of models is

quite large (uncountably infinite), it is intractable to consider all misspecifications, so we only consider a few of them below.

- **Quadratic complexity**: computational time complexity should be quadratic in the number of or less.
- **Interpretable**: we desire that the parameters are interpretable with respect to a subset of vertices and/or edges.

In addition to the above categorical desiderata, we also desire "nice" finite sample and empirical performance.

2.4.1 Signal Subgraph Estimators

Naïvely, one might consider a search over all signal subgraphs, plugging each one in to the classifier, and select the best performing signal subgraph. This strategy performs poorly with respect to at least two desiderata. First, the number of signal subgraphs scales super-exponentially with the number of vertices (see Figure 1, left panel). Specifically, the number of edges in a simple graph with V vertices is $d_V = \binom{V}{2}$, so the number of unique subgraphs is $2^{\binom{V}{2}}$. Searching over all of them is therefore ridiculously computationally taxing, and does not meet the quadratic complexity criteria. Second, the estimate will be determined partially by the chosen classifier. This makes interpreting the results a bit tricky, as one cannot ascertain whether the signal subgraph chosen is the one that works best for the chosen classifier, or is the true signal subgraph (assuming that they could be different). We therefore consider several alternatives.

Before proceeding, recall that each edge is independent, which means that whether or not each edge is in the signal subgraph could be evaluated on its own. Formally, consider a hypothesis test for each edge. The null hypothesis is that the class-conditional edge distributions are the same, so $H_0 : f_{uv|0} = f_{uv|1}$ for all $(u, v) \in \mathcal{S}$. The composite alternative hypothesis is that they differ, $H_A : f_{uv|0} \neq f_{uv|1}$ for all $(u, v) \in \mathcal{S}$. Given such hypothesis tests, one can construct test statistics using the data: $T = T_{uv}^{(n)} : \mathcal{D}_n \mapsto \mathbb{R}_+$. We reject the null in favor of the alternative whenever the value of the test-statistic is greater than some critical-value c: $T(\mathcal{D}_n) > c$. We can therefore construct a *significance matrix*: $\boldsymbol{T}_n = T_{uv}^{(n)}$, which encapsulates the significance of the difference for each edge between the classes.

2.4.1.1 *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 $\binom{d_V}{s}$, where d_V is the number of distinct edges in a graph with V vertices; also super-exponential (see Figure 1, middle panel). Thus searching them all is computationally intractable at this time. When s is given and the independent edge assumption is "good", one can choose the critical value *a posteriori* to ensure that only s edges are rejected, $c = \min_{c'} \mathbb{I}\{\sum_{(u,v)\in\mathcal{S}} \mathbb{I}\{T_{uv}^{(n)} > c'\} - s\}$, where $\mathbb{I}\{\cdot\}$ is the identity function, equaling one whenever its argument is true, and zero otherwise. Therefore, an estimate of the signal subgraph is the collection of s edges with minimal test-statistics. Formally, let $T_{(1)} < T_{(2)} < \cdots < T_{(d_V)}$ indicated the *ordered* test statistics (dropping the superscript indicating the number of samplesfor brevity). Then, the *incoherent signal subgraph estimator* is given by: $\hat{\mathcal{S}}_n^{inc}$=$\{a_{(1)}, \ldots, a_{(s)}\}$, where $a_{(u)}$ indicates the u^{th} edge ordered by significance of its test statistic, $T_{(u)}$.

2.4.1.2 *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 *star 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). Instead, we again take a greedy approach.

First, compute the significance of each edge, as above, yielding ordered test-statistics, and 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 $k \in \mathcal{V}$. Second, recursively increase the critical value, c. With each iteration, assign each vertex a "score" equal to the number of edges per vertex with significance smaller than the critical value, $w_{(i);c} = \sum_{u\in[V]} \mathbb{I}\{T_{i,u} < c\}$. If there exists m vertices whose scores sum to greater than or equal the size of the signal subgraph, s, then stop iterating. That is, find $\min_c!$ such that $\sum_i w_{(i);c} \geq s$. Call the collection of s most significant edges from within that subset the *incoherent signal subgraph estimate*, $\hat{\mathcal{S}}_n^{coh}$.

2.4.1.3 *Coherograms*: In the process of estimating the incoherent signal subgraph, one builds a "coherogram". Each row of the coherogram corresponds to a different critical value c, and each column corresponds to a different vertex v. The c, v element of the coherogram is the number of edges incident to vertex v with significance smaller than c. Thus, the coherogram gives a quick depiction of how coherent is the signal subgraph.

2.4.2 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 an independent Bernoulli parameter for each edge in each class. The maximum likelihood estimator (MLE), which simply the average value of each edge per class, seems to be a good choice:

$$\hat{p}_{uv|y}^{MLE} = \frac{1}{n_y} \sum_{i|y_i=y} a_{uv}^{(i)}, \tag{6}$$

where $\sum_{i|y_i=y}$ indicates the sum is over all data samples from class y. Unfortunately, the MLE has relatively poor finite sample properties. In particular,

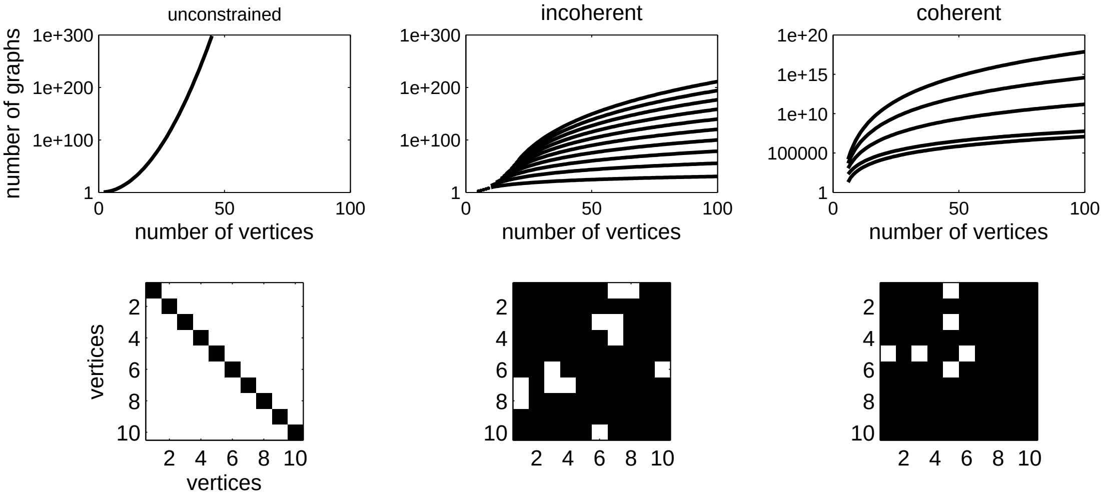

Fig. 1. Exhaustive searches for the signal subgraph, even given severe constraints, are computationally intractable for small graphs (e.g., with $\mathcal{O}(10)$ vertices). Top panels show the number of unique simple subgraphs as a function of the number of vertices, V. Note the ordinates are all log 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 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 star-vertex, that is, all edges are incident to one vertex. Bottom panels show a particular example subgraph via its adjacency matrix; white elements indicate an edge.

if the data contains no examples of an edge in a particular class, then the MLE will be zero. If the new to be classified graphs exhibits that edge, then the probability of it being from that class is zero, which we do not believe. We therefore consider an estimator with better finite sample performance, the maximum a posteriori (MAP) estimator (other choices, such as an ML$_q$ estimator, might be better []). The MAP estimator for a Bernoulli random variable requires specifying a prior. Because the beta distribution is the conjugate prior to the Bernoulli distribution, it is a convenient choice:

$$\begin{aligned} F[p_{uv|y}|\alpha,\beta] &= \text{Beta}(p_{uv|y};\alpha,\beta) \\ &= \frac{1}{B(\alpha,\beta)} p_{uv|y}^{\alpha-1}(1-p_{uv|y})^{\beta-1}. \end{aligned} \quad (7)$$

And because we have relatively little prior knowledge about the probabilities other than a disbelief with regard to zero, we choose a *weakly informative prior* [cite], namely the uniform prior: $\alpha = \beta = 1$. Given such a prior, the posterior distribution is simply: Beta$(\widetilde{\alpha}_{uv|y}, \widetilde{\beta}_{uv|y})$, where $\widetilde{\alpha}_{uv|y} = \alpha + n_{uv|y}$, $\widetilde{\beta}_{uv|y} = \beta + (n_y - n_{uv|y})$, and $n_{uv|y} = \sum_{i|y_i=y} a_{uv}^{(i)}$. The posterior is unimodal because both the parameters are greater than one [cite]. The mode (which is the maximum a posteriori estimate) is given by:

$$\hat{p}_{uv|y}^{MAP} = \text{Beta}(\widetilde{\alpha}_{uv|y}, \widetilde{\beta}_{uv|y}), \quad (8)$$

which we use for our likelihood estimates.

2.4.3 Prior Estimators

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

$$\hat{\pi}_y = \frac{n_y}{n}, \quad (9)$$

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

2.5 Evaluation Criteria

The properties of the MAP and ML estimators for the likelihood and prior terms, respectively, are well studied [] and will therefore not be discussed in much detail. We evaluate the classifier finite sample properties using either hold-out or leave-one-out misclassification performance, depending on whether the data is simulated or experimental. Formally, given C equally sized subsets of the data: $\{\mathcal{D}_1, \ldots, \mathcal{D}_C\}$, the *cross-validated error* is:

$$\hat{L}_{\hat{f}(\cdot;\mathcal{D}_n)} = \frac{1}{C}\sum_{c=1}^{C} \frac{1}{|\mathcal{D}_n \backslash \mathcal{D}_c|} \sum_{g \notin \mathcal{D}_c} \mathbb{I}\{\hat{f}(g;\mathcal{D}_c) \neq y\}, \quad (10)$$

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

$$R_n^x = \frac{1}{\mathcal{S}} \sum_{(u,v)\in\mathcal{S}} \mathbb{I}\{(u,v) \in \hat{\mathcal{S}}_n^x\} \quad (11)$$

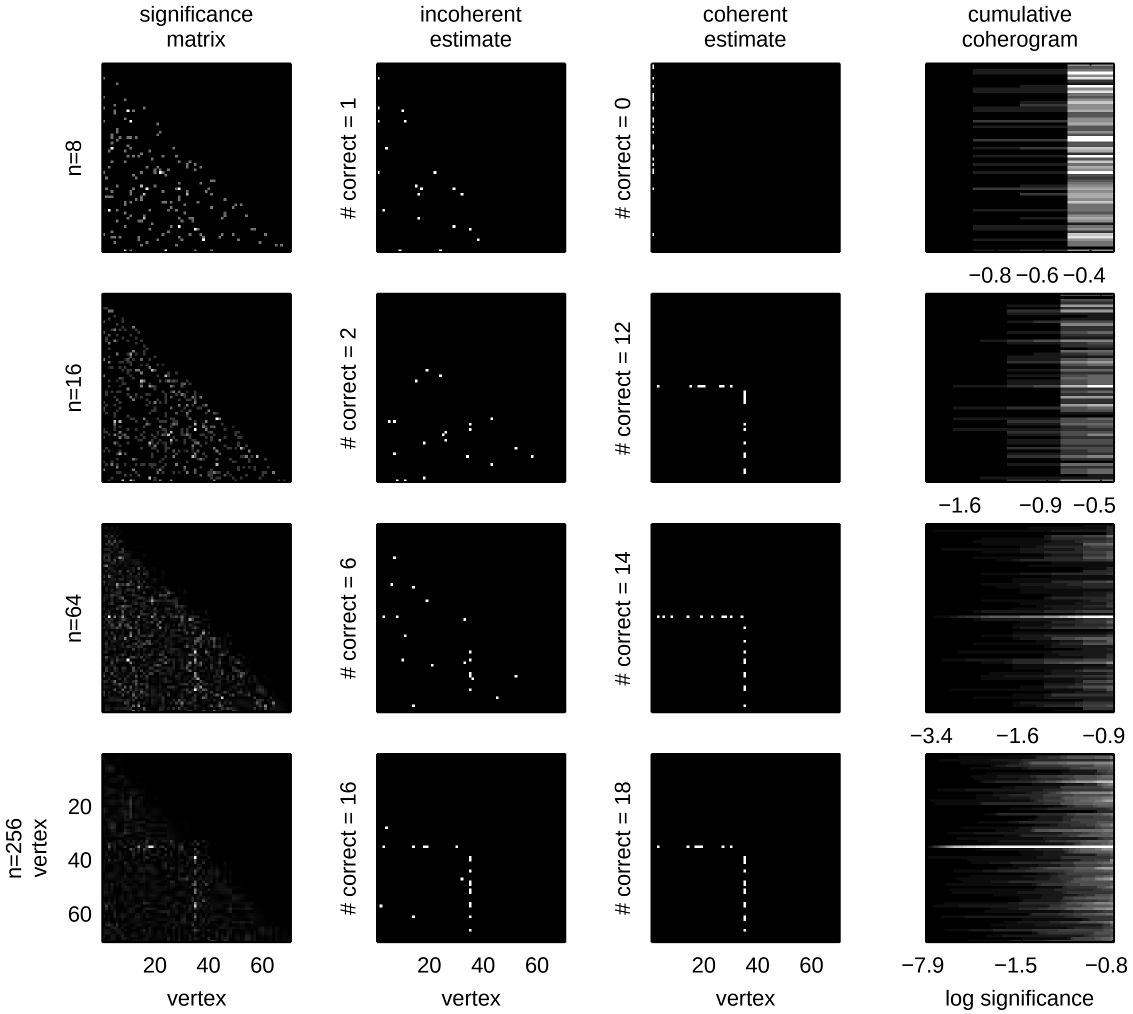

Fig. 2. An example of the coherent signal subgraph estimate's improved accuracy over the incoherent signal subgraph estimate, for a particular homogeneous two-class model specified by: $\mathcal{M}_{70}(1, 20; 0.5, 0.1, 0.3)$. Each row shows the same columns but for increasing the number of graph/class 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 data 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 star-vertex.

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 relative number of samples for the coherent estimator to obtain the same rate as the incoherent estimator.

3 Results

The Results section is subdivided into three subsections, corresponding to asymptotic properties, finite sample properties, and performance on connectome data.

3.1 Asymptotic properties

3.1.1 *Likelihood and Prior term Estimators*

MAP estimators are known to be consistent and efficient, both for finite samples and asymptotically, under certain special cases. Specifically, letting $d_V = \binom{V}{2}$ (the number of edges in a simple graph as a function of the number of vertices, V), and assuming $n \to \infty$ and V is fixed, we know that: $\hat{p}_{uv|y}^{MAP} \to p_{uv|y}$ []. The MAP estimator remains consistent when $V \to \infty$ as long as $d_V/V \to 0$ []. Moreover, both prior and likelihood estimates are trivial to compute, as closed-form analytic solutions are available for both.

3.1.2 Signal Subgraph Estimators

A variety of test-statistics are available for computing the edge-specific class-conditional signal, $T_{uv}^{(n)}$. Which ever test one uses, the sufficient statistics are encapsulated in a $|\mathcal{Y}|$ by two contingency table, indicating the number of edges observed in each class. For example, the two-class contingency table for each edge is:

	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

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 []. Furthermore, 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 between zero and one. Therefore, Fisher's exact test is a consistent estimator 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 []. While most powerful, computing Fisher's exactly is quite computationally time consuming. Fortunately, the chi-squared test is asymptotically equivalent to Fisher's test, and therefore shares those convergence properties []. Even the absolute difference of MAP estimates, $|\hat{p}_{uv|1}^{MAP} - \hat{p}_{uv|0}^{MAP}|$, which is trivially easy to compute, is asymptotically equivalent to Fisher's [] 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.

Moreover, the coherent signal subgraph estimator uses the exact 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 gained by finite sample efficiency whenever the model is coherent, as will be shown below.

3.1.3 Bayes plugin classifier

A Bayes plugin classifier is a consistent classifier whenever the estimates that are plugged in are consistent []. Because the likelihood, prior, and signal subgraph estimates are all consistent, the Bayes plugin classifier is therefore consistent as well. Moreover, it is well-known that a naïve Bayes classifier often exhibits impressive finite sample performance due to it winning the bias-variance trade-off relative to other classifiers. In other words, even when edge 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.

3.2 in simulo experiments

To better assess the finite sample properties of the signal subgraph estimators, we conduct a number of in simulo experiments. First, 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, with all edges in the *signal subgraph* have probability q, and all others have probability p: $f_{uv|1} = q \forall (u,v) \in \mathcal{S}$, and $f_{uv|1} = p \forall (u,v) \mathcal{E} \backslash \mathcal{S}$. The signal subgraph is constrained to have m star-vertices and s edges. The prior probability of being in class 0 is π, so $1-\pi$ is the probability of coming from class 1. 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. To evaluate the performance of the two above-described signal subgraph estimators for this model, we run some numerical experiments, with results provided in Figure 2. In each row, We sample $n/2$ graphs from each class defined by $\mathcal{M}_{70}(1, 20; 0.5, 0.1, 0.3)$. Given these n samples, we compute the significance matrix (first column), which is the object from which both estimators follow. The incoherent estimator simply chooses the s most significant edges as the signal subgraph (second column). The coherent estimator first guesses which are the m star-vertices, and then chooses the s most significant edges incident to at least one of those vertices (third column). The coherogram shows how "coherent" is the signal from the data (fourth column).

From this figure, one might notice a few tendencies. First, both the incoherent and coherent signal subgraph estimators are converging relatively quickly towards the true signal subgraph. Second, the coherent estimator seems to converge more quickly than the incoherent estimator. Third, the coherogram sharpens with additional samples, clearly showing that this model is strongly coherent after about only 50 samples.

To better characterize the relative performance of the two signal subgraph estimators, Figure 3 shows their performance as a function of the number of samples, n, for this model. The top panel shows the mean and standard error of the missed-edge rate: the fraction of edges incorrectly identified. For essentially all n's, the coherent estimator (black line) performs better. This translates directly to improved classification performance (lower panel), where the plug-in classifier using the coherent signal subgraph classifier (black line) has a better misclassification rate than the incoherent signal subgraph classifier (dark gray line) for essentially all n. For calibration purposes, the naive Bayes plug-in classifier, that is, the classifier that assumes the whole graph is the signal subgraph, is also shown (light gray line). Note that performance is bounded above by $L_{\text{chance}} = 0.5$ and bounded below by $L_* = XXX$, as it should be.

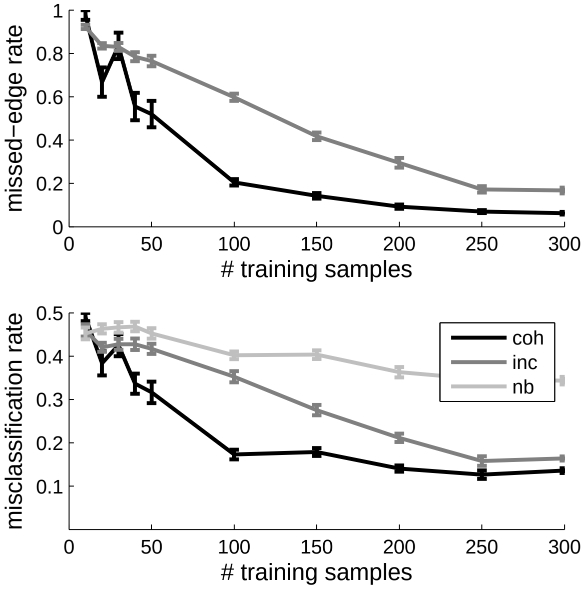

Fig. 3. Performance statistics as a function of sample size demonstrate that the coherent signal subgraph estimator outperforms the incoherent signal subgraph estimator, in terms of both the signal subgraph identification and classification, for the same model as in Figure 2. The top panel shows the missed-edge rate for each estimator as a function of the number of training samples, n. The bottom panel shows the corresponding misclassification rate for the two estimators, as well as the naïve Bayes plugin classifier. Performance of all estimators increases monotonically with n for both criteria. Error bars show standard error of the mean here and elsewhere (averaged over 20 trials; each trial used 100 samples for held-out data). Note that $L_{chance} = 0.5 \geq \hat{L}_{NB} \geq \hat{L}_{INC} \geq \hat{L}_{COH} \geq L_* = XXX$ for essentially all n here.

3.3 Relative Efficiencies

The above numerical results suggest that the coherent estimator outperforms the incoherent estimator. However, that result is a function of both the model, $\mathcal{M}_V$ (which includes the number of vertices), and the number of samples n. Figure 4 explicitly shows that the relative performance of an estimator for a particular model changes as a function of the number of samples. More specifically, for small n, the incoherent estimator yields a better performance, as indicated by by relative rate and relative efficiency being above one. However, with more samples, when the signal subgraph is coherent, the coherent estimator will eventually outperforming 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 best, 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.

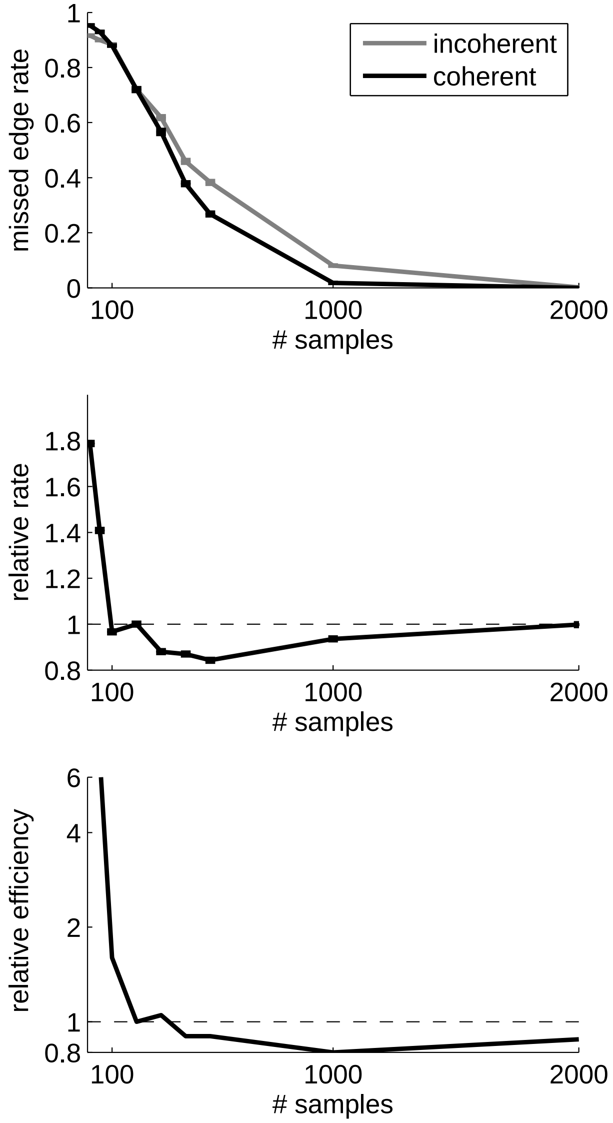

Fig. 4. The relative performance of the coherent and incoherent estimators is a function not just of the model, but also the number of samples. Specifically, for the model $\mathcal{M}_{30}(1, 5; 0.5, 0.1, 0.2)$, we compute the missed-edge rates for both the incoherent estimator (gray line) and the coherent estimator (black line). The top panel shows that for small 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, therefore, 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 they dip below unity, and then converge back up to unity, as they must because both estimators are consistent.

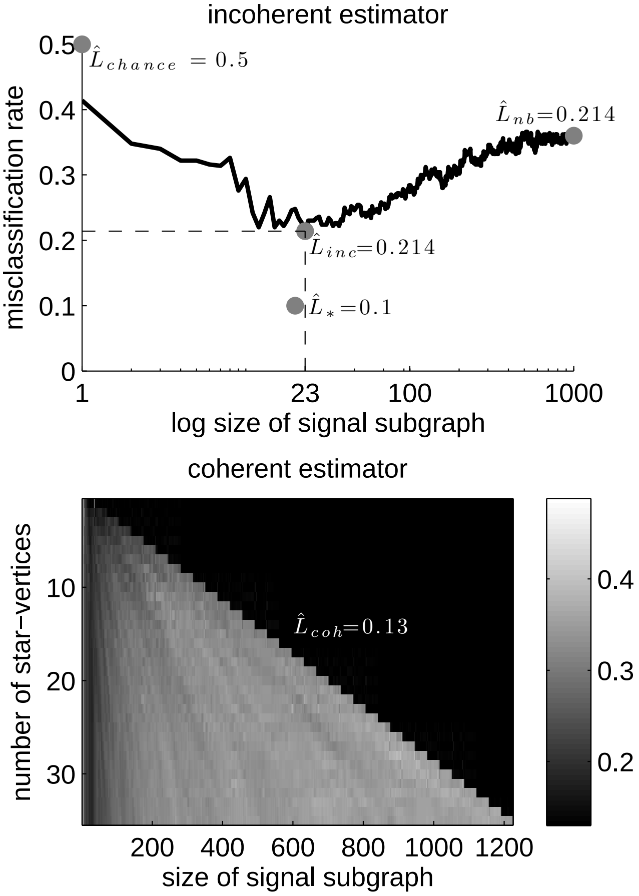

Fig. 5. When constraints on the number of edges (s) or star-vertices (m) are unknown, a search over these hyperparameters can yield estimates $\hat{s}$ and $\hat{m}$. Both panels depict held-out cross-validation error as a function of varying these parameters for the same model as in Figure 4, using 200 training samples and 500 test samples. The left panel depicts performance of the incoherent estimator by varying the number of edges from 1 to 1000. Note that in this simulation, while $s^* = 8$, $\hat{s}_{inc} = 23$. This "conservatism" is typical and appropriate in many model selection situations (see text for details). The right panel shows $\hat{L}_{coh}$ as a function of both m' and s'. For this simulation, $\hat{m} = 1$ and $\hat{s} = 24$.

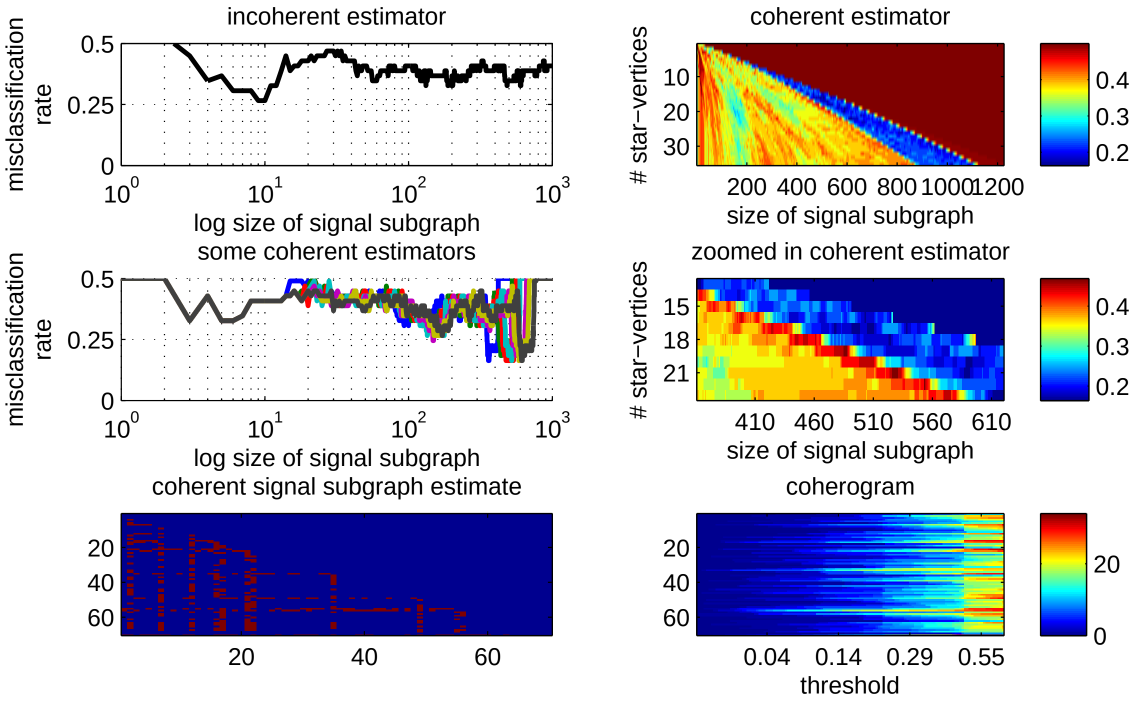

Fig. 6. MR connectome gender signal subgraph estimation and analysis. By cross-validating over hyperparameters and models, we estimate that the "best" signal subgraph (for this inference task on these data) has $\hat{m} = 12$ and $\hat{s} = 400$. As in the simulated data, we expect these estimates (and the particular edges in the signal subgraph) would change with more/different data. The top two panels depict the same as Figure 5. The middle two depict misclassification rate (left) for a few different choices of $\hat{m}$ 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 not particularly coherent.

Fig. 7. Model checking via synthetic data analysis is fundamental to understanding the performance of our estimators, and deciding how to improve them. The top panels demonstrate that the simulated performance is better than the performance on real data, suggesting that some of the assumptions are inaccurate. The middle panels suggest that if the model assumptions and parameter estimates were correct, than we would require approximately XXX samples before misclassification rate apparently converges (note that Bayes optimal performance is unknown in this data).

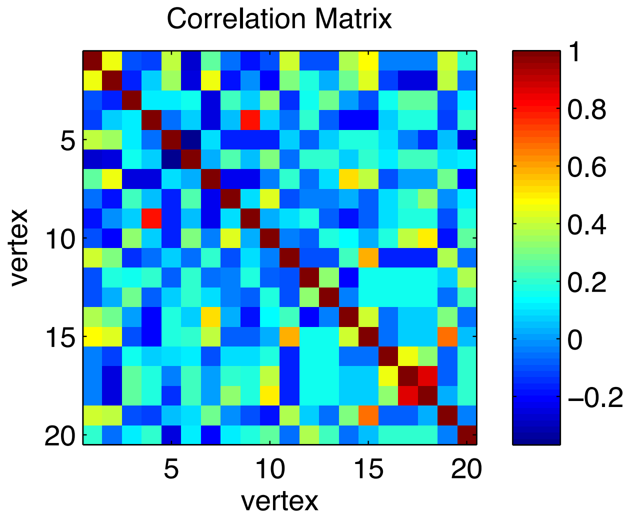

Fig. 8. The correlation matrix between all the edges in the coherent signal subgraph estimate. It should be clear that many of these edges, which were assumed independent, are in fact highly correlated. This perhaps explains why the synthetic data analysis yielded improved estimates over the actual data: edges are not independent as assumed. Moreover, this suggests that improved performance might be achieved by relaxing the independent edge assumption.

4 Discussion

sometimes one wins B/V by doing Ess estimation:

gender data: $L_chance > L_NB > L_inc > L_coh > L_semi > L_* >= 0$

4.1 Contributions

F_GY, algs, advice, data

4.2 Next Steps

semi-coh, priors, proofs, relax independence (a) cond'l ind (b) m-est, model selection (short shrift), model avg (lacks interp)

4.3 Related Work

LASSO/ENET, low-rank+sparse, other graph classification approaches (invariants, embedding, kernels) [1]

Acknowledgments

References

[1] H. Pao, G. A. Coppersmith, and C. E. Priebe, "Statistical Inference on Random Graphs : Comparative Power Analyses via Monte Carlo," 2010.

PLACE PHOTO HERE

Joshua T. Vogelstein Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.