[4] E.J. Candès and M. Wakin, "An Introduction to Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21-30, Mar. 2008.

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

[6] N.S. Ketkar, L.B. Holder, and D.J. Cook, "Empirical Comparison of Graph Classification Algorithms," *Proc. IEEE Symp. Computational Intelligence and Data Mining*, pp. 259-266, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4938658, Mar. 2009.

[7] G. North, *Invertebrate Neurobiology*, R.J.G. Geoffrey North, ed. CSHL Press, 2007.

[8] J.D. Shepherd and R.L. Huganir, "The Cell Biology of Synaptic Plasticity: AMPA Receptor Trafficking," *Ann. Rev. Cell and Developmental Biology*, vol. 23, pp. 613-643, http://www.ncbi.nlm.nih.gov/pubmed/17506699, Jan. 2007.

[9] J. Nolte, *The Human Brain: An Introduction to Its Functional Anatomy*. Mosby, 2002.

[10] J.W. Lichtman, J. Livet, and J.R. Sanes, "A Technicolour Approach to the Connectome," *Nature Rev. Neuroscience*, vol. 9, no. 6, pp. 417-422, http://dx.doi.org/10.1038/nrn2391, June 2008.

[11] 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.

[12] J. Lasserre, C.M. Bishop, and T. Minka, "Principled Hybrids of Generative and Discriminative Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 87-94, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1640745, 2006.

[13] P.J. Bickel and K.A. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics*, vol. 1, second ed. Prentice Hall, 2000.

[14] C.M. Stein, "Inadmissibility of the Usual Estimator of the Mean of a Multivariate Normal Distribution," *Proc. Third Berkeley Symp. Math. Statistics and Probability*, pp. 197-206, 1956.

[15] P.I. Good, *Permutation, Parametric, and Bootstrap Tests of Hypotheses*. Springer, 2010.

[16] Q. McNemar, "Note on the Sampling Error of the Difference between Correlated Proportions or Percentages," *Psychometrika*, vol. 12, no. 2, pp. 153-157, June 1947.

[17] P.J. Huber, *Robust Statistics*. Wiley, http://doi.wiley.com/10.1002/9780470434697, 1981.

[18] J.A. Rice, *Mathematical Statistics and Data Analysis*. Duxbury Press, 1995.

[19] D.J. Hand and K. Yu, "Idiot's Bayes: Not So Stupid After All?" *Int'l J. Statistical Rev.*, vol. 69, no. 3, pp. 385-398, Nov. 2001.

[20] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *J. Royal Statistical Soc., Series B*, vol. 58, pp. 267-288, 1996.

[21] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least Angle Regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407-499, 2004.

[22] A.K. Jain, R.P.W. Duin, J. Mao, and S. Member, "Statistical Pattern Recognition: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4-37, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=824819, Jan. 2000.

[23] O. Sporns, *Networks of the Brain*. The MIT Press, 2010.

[24] 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.

[25] W.R. Gray, J.A. Bogovic, J.T. Vogelstein, B.A. Landman, J.L. Prince, and R.J. Vogelstein, "Magnetic Resonance Connectome Automated Pipeline: An Overview." *IEEE Pulse*, vol. 3, no. 2, pp. 42-48, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6173097, Mar. 2010.

[26] J.T. Vogelstein, R.J. Vogelstein, and C.E. Priebe, "Are Mental Properties Supervenient on Brain Properties?" *Nature Scientific Reports*, p. 11, 2011.

[27] H. Pao, G.A. Coppersmith, C.E. Priebe, H.P. Ao, G.A.C. Oppersmith, and C.E.P. Riebe, "Statistical Inference on Random Graphs: Comparative Power Analyses via Monte Carlo," *J. Computational and Graphical Statistics*, 20, pp. 1-22, http://pubs.amstat.org/doi/abs/10.1198/jcgs.2010.09004, 2010.

[28] 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.

[29] A. Rukhin and C.E. Priebe, "A Comparative Power Analysis of the Maximum Degree and Size Invariants for Random Graph Inference," *J. Statistical Planning and Inference*, vol. 141, no. 2, pp. 1041-1046, 2011.

[30] I.S. Dhillon, "Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning," *Proc. Seventh ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pp. 269-274, http://portal.acm.org/citation.cfm?doid=502512.502550, 2001.

[31] L. Devroye, L. Györfi, G. Lugosi, and L. Gyorfi, *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[32] E.J. Candès and B. Recht, "Exact Matrix Completion via Convex Optimization," *Foundations of Computational Math.*, vol. 9, no. 6, pp. 717-772, Apr. 2009.

[33] X. Ding, L. He, and L. Carin, "Bayesian Robust Principal Component Analysis," *Image*, vol. 20, no. 12, pp. 3419-3430, 2011.

[34] V. Chandrasekaran, S. Sanghavi, P.A. Parrilo, and A.S. Willsky, "Rank-Sparsity Incoherence for Matrix Decomposition," *SIAM J. Optimization*, vol. 21, no. 2, p. 572, June 2011.

[35] P.D. Hoff, A.E. Raftery, and M.S. Handcock, "Latent Space Approaches to Social Network Analysis," *J. Am. Statistical Assoc.*, vol. 97, no. 460, pp. 1090-1098, http://pubs.amstat.org/doi/abs/10.1198/016214502388618906, Dec. 2002.

[36] D.L. Sussman, M. Tang, D.E. Fishkind, and C.E. Priebe, "A Consistent Dot Product Embedding for Stochastic Blockmodel Graphs," *J. Am. Statistical Assoc.*, p. 17, 2012.

[37] 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," Rapid Post, http://arxiv.org/abs/1205.0309, p. 20, May 2012.

[38] J.T. Vogelstein, J.C.M. Conroy, L.J. Podrazik, S.G. Kratzer, D.E. Fishkind, R.J. Vogelstein, and C.E. Priebe, "(Brain) Graph Matching via Fast Approximate Quadratic Programming," Rapid Post, 2011.

[39] J.T. Vogelstein and C.E. Priebe, "Shuffled Graph Classification: Theory and Connectome Applications," ~~*IEEE Trans. Pattern Analysis and Machine Intelligence*, 2011~~. (21)

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

[41] D.J. Marchette, C.E. Priebe, and G.A. Coppersmith, "Vertex Nomination via Attributed Random Dot Product Graphs," 2011.

[42] D.S. Lee and C.E. Priebe, "Bayesian Vertex Nomination," Rapid Post, 2012.

[43] D.D. Bock, W.-C.A. Lee, A.M. Kerlin, M.L. Andermann, A.W. Wetzel, S. Yurgenson, E.R. Soucy, H.S. Kim, G. Hood, and R.C. Reid, "Network Anatomy and in Vivo Physiology of Visual Cortical Neurons," *Nature*, vol. 471, no. 7337, pp. 177-182, Mar. 2011.

TABLE 1
Bake-Off Comparing a Number of Different Classifiers on the MR Connectome Sex Classification Data

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
signal-subgraph	0.16	n\a

Error indicates misclassification error using the best hyperparameters found for each classifier. p-val indicates the p-value of a one-sided McNemar's test comparing each classifier to the best signal-subgraph classifier. The signal-subgraph classifier is significantly better than all the others.

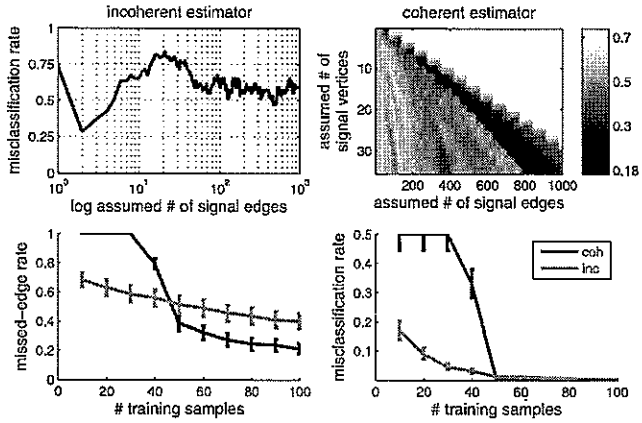

Fig. 7. Synthetic data analysis provides some intuition for model checking and future improvements. The top two panels show the incoherent (left) and coherent (right) misclassification rates as a function of the hyperparameter choices for $n = 49$. These plots look quite similar to those obtained in the real connectome data (Fig. 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.

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, normalized each to have zero mean and unit variance, and then used a kNN with ℓ_2 distance metric on the invariants. These particular invariants were chosen based on their desirable statistical properties [27], [28], [29].

Despite the small sample size, Table 1 demonstrates that the signal-subgraph classifier is significantly better than all the others, as assessed via a one-sided McNemar's test.

5.1 Model Evaluation

We investigate to what extent the above estimated signal-subgraph represents the true signal-subgraph. We address this question in two ways: 1) synthetic data analysis and 2) 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 $\widehat{\mathcal{S}}_n$, 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 $\widehat{\mathcal{S}}_n$, 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 generated 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 (Fig. 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 percent of the edges, whereas the coherent signal-subgraph estimator correctly identifies about 50 percent. This suggests that even if the model were true (which we doubt) we are justified in believing 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, 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 1) very useful for algorithms and analysis, and 2) almost certainly nonsense for real connectome data. Checking whether edges are independent is relatively easy. Fig. 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 [30] 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 [19].

6 Discussion

6.1 Summary

This work makes the following contributions: First, it introduces a novel graph/class model that admits rigorous

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 naive Bayes plugin classifier, that is, the classifier that assumes the whole graph is the signal-subgraph, is also shown (black dashed line). Note that the performance of all the classifiers is bounded above by $L_{\hat{\pi}} = 0.5$ and below by $L_* = 0.13$. Moreover, $\widehat{L}_{nb} > \widehat{L}_{lasso} > \widehat{L}_{inc} > \widehat{L}_{coh}$ 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 hyperparameter path [20]. Despite the fact that efficient computational schemes have been developed for searching the whole regularization path [21], such iterative algorithms should be much slower than our classifiers.

Indeed, the lower panel of Fig. 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 Fig. 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. 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 the incoherent estimator almost always, despite the fact 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 tradeoff here, as always). Fig. 4 explicitly shows that the relative performance of an estimator for a particular model—$\mathcal{M}_{30}(1, 5; 0.5, 0.1, 0.2)$—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.

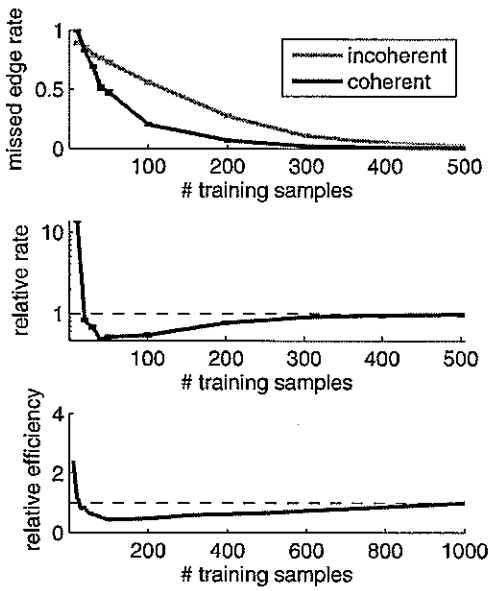

Fig. 4. The relative performance of the coherent and incoherent estimators is a function not just of the model, but also of the number of training samples. Specifically, for the same model, $\mathcal{M}_{70}(1, 20; 0.5, 0.1, 0.3)$, we compute the missed-edge rates for both the incoherent estimator (gray 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.

4.4 Estimating the Hyperparameters

In the above analyses, the hyperparameters, 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 hyperparameters, 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. Fig. 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 [22]. This intuition is further reified by the U-shape of the misclassification curve on a log scale: Including many nonsignal-edges is less detrimental than excluding a few signal-edges.

The bottom panel shows the coherent performance 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 are not expected to change performance much. The best performing

Algorithm 3. Pseudocode for training signal-subgraph classifiers.
Input: $\mathcal{T}_n$ and a set of constraints $(\vec{s}, \vec{m})$
Output: $\widehat{\mathcal{S}}_n$, $\{\widehat{p}_{uv|y}\}_{(u,v)\in\widehat{\mathcal{S}}_n}$, $\{\widehat{\pi}_y\}_{y\in\{0,1\}}$

1: Partition the data for the appropriate cross-validation procedure
2: Estimate $p_{uv|y}$ for all (u, v) using (11)
3: Estimate π_y for all y using (12)
4: for all $(s, m) \in (\vec{s}, \vec{m})$ do
5: Compute $\widehat{\mathcal{S}}_n(s, m)$ using Algorithm 1 or 2, as appropriate
6: Compute cross-validated error $\widehat{L}_{s,m}$ using (13)
7: end for
8: Let $\widehat{\mathcal{S}}_n = \text{argmin}_{(s,m)} \widehat{L}_{s,m}$

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 held-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

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

Given this definition, let $L_{\widehat{\pi}}$ 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 naive 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 [15].

To determine whether a pair of classifiers are significantly different, we compare the leave-one-out classification results using McNemar's test [16].

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:

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

Note that when $|\mathcal{S}|$ is fixed, miss-edge rate is a sufficient statistic for all combinations of false/negative positive/negative results. 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 (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 [17]. Indeed, $\widehat{p}_{uv|y}$ 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 [17]. The prior estimators are MLEs, 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 contingency 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 [18]. This leads to the following lemma:

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

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 [18]. □

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 [18]. Even the absolute difference of MLEs, $|\widehat{p}_{uv|1}^{MLE} - \widehat{p}_{uv|0}^{MLE}|$, which is trivially easy to compute, is asymptotically equivalent to Fisher's [18] and therefore consistent. Moreover, the signal-subgraph estimators are robust to a variety of model misspecifications. Specifically, as long as all the marginal probabilities of all the edges in the signal-subgraph are different between the two classes, $p_{uv|1} \neq p_{uv|0}$, and the constraints are upper-bounds on the true values, $s' \geq s$ and $m' \geq m$, then any consistent test statistic will yield a consistent signal-subgraph estimator. Estimating the coherent signal-subgraph is more computationally time consuming than estimating the incoherent signal-subgraph. What is lost by computational time, however, is typically gained by finite

For simplicity, we will assume that $|\mathcal{Y}| = 2$ for the remainder, though the general case is relatively straightforward.

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 superexponential (see Fig. 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 $\widehat{\mathcal{S}}_n(s) = \{e_{(1)}, \ldots, e_{(s)}\}$, where $e_{(u)}$ indicates the uth edge ordered by significance of its test statistic, $T_{(u)}$.

Note that the number of distinct test-statistic values is typically much smaller than the number of possible settings of s; 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 hyperparameters 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.

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: $\widehat{\mathcal{S}}_n(s)$
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 $\widehat{\mathcal{S}}_n(s) = \{e_{(1)}, \ldots, e_{(s)}\}$, arbitrarily breaking ties as necessary.

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 Fig. 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, $e_{k,(1)} \leq e_{k,(2)} \leq \cdots \leq e_{k,(n-1)}$ 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 exist m vertices whose scores sum to greater than or equal the size of the signal-subgraph, s. That is, check whether the 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*, $\widehat{\mathcal{S}}_n(s,m)$. If not, increase c and go back to step four. As above, we break ties arbitrarily. Pseudocode for implementing the coherent signal-subgraph estimator is provided in Algorithm 2, and MATLAB code is available from http://jovo.me.

Algorithm 2. Pseudocode for estimating coherent signal-subgraph.
Input: $\mathcal{T}_n$ and (s,m)
Output: $\widehat{\mathcal{S}}_n(s,m)$
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 $\widehat{\mathcal{S}}_n(s,m)$ be the collection of s edges from amongst those that satisfy (9) for the final value of c, arbitrarily breaking ties as necessary.

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 Fig. 2, right column, for some examples).

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

We demonstrate via theory, simulation, analysis of a neurobiological dataset (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 dataset 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: $\{(\mathbb{G}_i, Y_i)\}_{i\in[n]} \sim^{exch.} F_{\mathbb{G},Y}$. 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 some 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.

We adopt the common practice of identifying graphs with their adjacency matrices. We note, however, that operations available on the latter (addition, multiplication) are not intrinsic to the former.

2.2 Model

Consider the model, $\mathcal{F}_{\mathbb{G},Y}$, which includes all joint distributions over graphs and classes under consideration: $\mathcal{F}_{\mathbb{G},Y} = \{F_{\mathbb{G},Y}(\cdot;\boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}$, where $\boldsymbol{\theta} \in \Theta$ indexes the distributions. We proceed via a hybrid generative-discriminative approach [12] 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 as 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:

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

where

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

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 nonempty 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*. Note that given a collection of signal-edges, the signal-vertex set may not be unique. 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 *hyperparameters*.

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

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:

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

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:

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

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 [13]: